The confproc package*

Vincent Verfaille†

Printed on August 1, 2011

Abstract

The confproc package provided a LaTeX 2ε document-class together with various tools (Perl and Unix/bash scripts) for building conference proceedings, or concatenating any set of PDFs with a table of contents, index and bookmarks. The LaTeX2e class derives from LaTeX 2ε scripts written for the DAFx-06 conference proceedings. It is mainly based on the 'pdfpages' package for PDF papers including, and the 'hyperref' package for creating proper links, bookmarks and general bibliography back references. It also uses many other packages for fine tuning of table of contents, bibliography and index of authors. Current version 0.8 is the previous 0.7 major update with key-value option management, that has now been tested with TeXLive 2011. The added value of this class is in the time it saves for you to quickly design conference proceedings. See readme.txt for a short overview and additional (legal) information, and exampleN.tex and corresponding files and scripts for an example of use.

Contents

*This file version number is v0.8: last revision on 2011/08/01; doc is dated 2010/08/01.

†confproc.verfaille@gmail.com

New [v0.7]

I do not want to read all this!!!

Yes, that's a fairly long table of contents... Let me give you some shortcuts:

- **very busy people can directly jump to section 3 for a first documented example, provided they have a full TEXLive installation (at least 2008)**;

New [v0.7]

- smart people can jump to the other and condensed documentation: `confproc-short.tex`, should they come back to this one for more details;
- curious people should rather read the not so short introduction in section 4—it provides some more details, such as the options and commands description—and the full example in section 5—which illustrates other functionalities such as the general bibliography).
- compulsive readers are welcome to read everything, starting from the introduction :-). Their comments about the contents, typos, etc. are much appreciated.
- TEX Programmers are not encouraged to read the implementation in section 8, as my LATEX programming skills are, let's say, still improving! Their comments are much appreciated too.

New [v0.7]

Color code and pictograms

This documentation uses the following color code:

- red: package options; new in this version (margin notes);
- blue: reference, URLs, internal links to sections, chapters, etc;
- grey: portions of example code that do not differ from similar portions of the last example previously described in the documentation;

- important things to pay attention at are noticed with this 'danger' margin sign;

New [v0.7]

- new elements in current version are acknowledged with this margin note.

N.B.: this code documentation also uses the `colordoc` package, by Federico Garcia (CTAN: macros/latex/contrib/colordoc).

1 Introduction

The provided `confproc` class is based on several great packages, among which `pdfpages` [11] by Andreas Matthias (IMHO, the most useful package to build proceedings) together with `hyperref` [10] by Sebastian Rahtz and Heiko Oberdiek (to manage all PDF and hyperlinks issues). So, you may consider `confproc` as a time saving package to faster design conference proceedings or a compilation of PDFs (such as an article collection).

1.1 Short history

When editing the DAFx-06 proceedings, I developed a set of LaTeX 2ε commands to produce the best quality proceedings we could achieve thanks to LaTeX 2ε. This was documented on the DAFx-06 website [14] and in a technical report [15]. Later on, I created a shorter example version, that has been used as a basis by other proceedings editors for their own needs (LAC 2007, ICAD 2007, DAFx-07, JMUI). For better sharing of this example with other LaTeX 2ε users, I converted the set of LaTeX 2ε commands into a document class—thanks to the information provided by the LaTeX3 team [2]—and then into a package producing all necessary files (*i.e.* the class, the documentation, the example, the scripts, etc—using Docstrip [3] together with the documentation by Scott Pakin [1]. Then, after using this class with new scripts for the ICMC'09 conference proceedings, I wrapped it up into a package (`.dtx` file) that generates all needed files.

1.2 Other packages or softwares

I tried several alternative solutions in the fall of 2005. Indeed, there are so many talented people out there developing great LaTeX packages that I would have preferred to use anybody else's solution! Unfortunately, I have not been able to make any of them work in the way I needed. So I before finally decided to create my own package.

N.B.: since the following information dates back to the fall of 2005, some of the following packages may have evolved in the meantime. Please take a look at them in order to get the latest information (which I obviously did not do)!

1.2.1 Adobe Acrobat

Eventhough it is nothing related to a LaTeX package, nor a free application, the Acrobat Professional software [8] is a solution to create proceedings with proper internal links for a set of PDF papers with internal links. Some useful explanations will help to understand al that has to be done [5]. Indeed, you have to do all the links for the table of contents, the index of authors and the general bibliography by hand. This sounds like hours of work! Would you really plan to do that, and potentially having to re-do it all when discovering any small error, as it happens during both the editing and the printing processes? Any LaTeX solution would provide automatization of proceedings building.

1.2.2 The `combine` package

The one I would have loved to be able to use in 2006 is the `combine` package by Peter Wilson [9], as it was especially designed for the purpose of combining articles into proceedings. It would have been perfect if it did not have incompatibilities with our `dafx06.sty` proceedings template (or conference style), since many commands are added in the header file. I encountered problems with the `hyperref` package as well as some minor problems with `fancyhdr.sty`: eventually, no paper was inserted in the proceedings, and the LaTeX run would always fail (stopped without any notice during the first paper inclusion). Very frustrating, as it was too late for changing our conference proceedings style to make them compatible with `combine`. I contacted Peter Wilson, to which I am indebt for all the precious advices he gave me, among which was the use of a concurrent solution, *i.e.* the `pdfpages` package!

1.2.3 The `pdfpages` package

As no magic solution do exist (yet?), the `pdfpages` package by Andreas Matthias [11] is a very easy way to combine several PDF documents into a single document. Unfortunately, where `combine` seemed to be able to preserve internal references of each paper, `pdfpages` does not provide such feature, as papers are included as a set of single PDF pages. As I am not a specialist of the PDF format and so on, I can imagine that it is extremely complex to achieve such a feature. Anyway, it means that if your original PDF documents had internal links, hyper-references, links to URL, etc, they will simply be all broken.

With this in mind, we used this package as a basis (so it then is not a concurrent), especially for the following feature: clicking on a page in the proceedings will open the corresponding paper (with its proper internal links). Simple!

1.2.4 The `mini` style

The `mini.sty` package [6] does a very good job for concatenating abstracts in a single proceedings document. However, it is not suited (to my knowledge) for conference proceedings, where each paper has to be compiled with the conference style and has its very own title, authors, etc. (that cannot be inserted as (sub)sections).

1.2.5 The AMS `editor` package

The `editor` package from the AMS [7] provides information and documents to produce both the front end and the back end of proceedings, which is of great help to understand all that has to be done (particularly the table of contents and the re-numbering of all papers). However, as they explicitely say it, there is no mechanism to assemble the files together.

1.3 A solution: the `confproc` package

1.3.1 Short description

Using all the knowledge I could find around (and in the previously cited documentations about how to do a good PDF document for the proceedings), together with many tricks I found, this LaTeX class provides the following features:

1. automatically generates the whole proceedings, after changing any of its paper information (thanks to LaTeX!);
2. concatenates papers by inserting several individual documents into one document (with the `pdfpages` package);
3. provides 'clickable' links (hyper-references) from the table of contents, the index of authors and the full bibliography to access to the corresponding page(s) (with the `hyperref` [10] package);
4. provides access to individual papers: a click on any paper's page opens the corresponding PDF paper (that still has its internal links); this feature comes with the `pdfpages` package.
5. left-aligned page numbers in the table of contents (using the `titlesec`) package;
6. displays the index of authors with two or three columns (hack derived from `twocolindex`, and using the `multicolumn` package);
7. organizes the bookmarks by proceedings' sections: the preamble, the table of contents, the days/sessions, the full bibliography, and the index of authors. Also, authors' names appear under their relative paper title.

8. organizes the table of contents: only the index of authors appearing in the table of contents (using the `tocbibind` package);

9. provides full bibliography, or at least help and informations for you to build one, with right-flushed back-reference page numbers.

10. enables fast LaTeX run, using the draft option of `pdfpages`. Useful when repetitively correcting errors, changing the layout (index, bookmarks, table of contents), merging bibliographies, etc. However, note that with this option, `pdfpages` does not generate the bookmark data. So, do not use it for final LaTeX runs!

11. orders the packages. As `hyperref` [10] redefines most of LaTeX internal commands, a lot of care has to be taken when ordering the insertion of packages, otherwise some of the features can disappear.

12. gives information about the merging process involved to generate a general bibliography, as well as about production issues.

13. offers various `bash/Unix` scripts to help automatize the making of conference proceedings.

1.3.2 Pros

There are numerous advantages with the `confproc` class:

- help: it simplifies operations such as generating a conference program as the table of contents, generating the index of authors, generating the bookmarks, having the same layout for the proceedings as for the paper templates;
- convenient: it provides an all-in-one package (with various useful scripts);
- time saving: directly and elegantly re-use all the tricks previously collected or developed;
- customization: it provides several commands and options to customize your document;
- package ordering: it correctly inserts the `hyperref` package as the last one (all internal macros are redefined), except for packages requiring to be inserted after (like `hypcap`);
- reliability: the `confproc` package is getting older and mature, and has been used for 8 issues. Its documentaton and option set make it now easy to use (however, editing conference proceedings is always a big job).

1.3.3 Cons

There are also disadvantages, among which:

- package ordering: the order of package insertion is fixed, and may not be changed: `hyperref` has to be inserted last because it redefines many internal. After you add packages in your document, this will not be the case anymore!!! This is the main limitation I can think of, and would appreciate any feedback, comments, tricks, that would help to resolve this issue.
- PDFs: `pdfpages` inserts PDFs as vectorial images (my understanding), so internal links are broken and the text cannot anymore be copied/paste. Hopefully, clicking on a paper page from the proceedings opens the original file!
- customization is a bit limited to the class designer's defined commands (which are hopefully expanding in each version);
- does not deal with parallel session programs.

1.3.4 Hall of fame

Under one of its various forms, this package has been used for (at least) the following conferences:

- version 0.5 (class/package):
  - August 2009: Proceedings of the International Computer Music Conference (ICMC 2009) — Montreal, Qc, Canada; used by Gary Scavone and myself; www.icmc2009.org/
- version 0.4e (class/package):
  - September 2010: Proceedings of the 5th Robotour Workshop — Bratislava, Slovakia; used by Richard Balogh; http://www.robotika.sk/robotour/zbornik/Robotour2010.pdf
  - July 2010: Proceedings of the International Society for Photogrammetry and Remote Sensing (ISPRS) Technical Commission VII Symposium: "100 Years ISPRS — Advancing Remote Sensing Science", Volume XXXVIII, Part 7A and 7B — Vienna, Austria; used by Alexandra von Beringe, Peter Dorninger, Sebastian Flöry, Josef Jansa, Clemens Nothegger, Norbert Pfeifer, Andreas Roncat;
    - Part 7A: www.isprs.org/proceedings/XXXVIII/part7/a/proceedings_partAweb.pdf
    - Part 7B: www.isprs.org/proceedings/XXXVIII/part7/b/proceedings_partBweb.pdf
  - June 2010: Proceedings of the 11th UC Systemwide 2010 Bioengineering Symposium University of California — Davis; California, USA; used by Angelique Louie; http://www.be.ucsd.edu/bicsymposium/program
  - March 2010: Proceedings of the Workshop on Inverse Problems for Waves — Palaiseau, France; used by Armin Lechleiter; www.cmap.polytechnique.fr/∼defi/mmsn2010/MMSN-2010.pdf
  - September 2009: Pre-Proceedings of the UC09 Hypercomputation Workshop — Ponta Delgada, The Azores, Portugal; used by Mike Stannett; hypercomputation.net/uc09/preproc.pdf
  - August 2009: Book of abstracts of the 16th European Young Statisticians Meeting (EYSM 2009) — Bucharest, Romania; used by Luiza Bădin and Roxana Ciumara; www.eysm2009.ase.ro/
  - June 2009: Nanophotonics Down Under 2009: Devices and Applications (SMONP: Sir Mark Oliphant Conference on NanoPhotonics) — Melbourne, Australia; used by Michael James Ventura; www.smonp2009.com/
- version 0.4f (scripts):
  - 2008 and 2009: Numediart's Quartely Progress Scientific Report (QPSR); used by Christian Frisson;
    - Vol. 2(4), Dec. 2009: www.numediart.org/docs/numediart_2009_s08_qpsr.pdf
    - Vol. 2(3), Sept. 2009: www.numediart.org/docs/numediart_2009_s07_qpsr.pdf
    - Vol. 2(2), June 2009: www.numediart.org/docs/numediart_2009_s06_qpsr.pdf
    - Vol. 2(1), March 2009: www.numediart.org/docs/numediart_2009_s05_qpsr.pdf
    - Vol. 1(4), Dec. 2008: www.numediart.org/docs/numediart_2008_s04_qpsr.pdf
    - Vol. 1(3), Sept. 2008: www.numediart.org/docs/numediart_2008_s03_qpsr.pdf
  - July 16 - August 10, 2007: Proceedings of the eNTERFACE'07 Workshop on Multimodal Interfaces — Istanbul, Turkey; used by Christian Frisson and Rémy Lehembre; http://www.cmpe.boun.edu.tr/enterface07/results.php
- version 0.2e (scripts):

  - 2008: Numediart's Quartely Progress Scientific Report (QPSR); used by Christian Frisson;
    * Vol. 1(2), June 2008: www.numediart.org/docs/numediart_2008_s02_qpsr.pdf
    * Vol. 1(1), March 2008: www.numediart.org/docs/numediart_2008_s01_qpsr.pdf
  - September 2007: 10th International Conference on Digital Audio Effects (DAFx-07) in Bordeaux, France; used by Sylvain Marchand; dafx.labri.fr/
  - 2007: Journal on Multimodal User Interfaces (JMUI) Vol. 1(1) and 1(2); used by Christian Frisson; www.jmui.org/index.php/JMUI/issue/view/1/showToc
  - June 2007: 13th International Conference on Auditory Display (ICAD-07) — Montreal, Qc, Canada; used by Gary Scavone; www.music.mcgill.ca/icad2007/proceedings.php
- version 0.1 (scripts):
  - March 2007: 5th International Linux Audio Conference (LAC2007) — Berlin, Germany; used by Marije Baalman; www.kgw.tu-berlin.de/~lac2007/proceedings.shtml
  - September 2006: 9th International Conference on Digital Audio Effects (DAFx-06) — Montreal, Qc, Canada; used by myself; www.dafx.ca/dafx06_proceedings.html

1.4 Version history

Here is a list of versions (red versions are public releases):

New [v0.8] **v0.8** last version is a minor revision, that:

New [v0.8]
- version 0.8 has been tested for TeXLive 2011 (as of August 1st, 2011);
- corrects the author's email address;
- adds other conference proceedings to the hall of fames.

New [v0.7] **v0.7** major revision. It took me a while to add functionalities and modify the option interface (as kindly suggested by Andreas Matthias more than a year ago):

- class design:
  - class options: interface re-designed, now uses key-values style (with `kvoptions` package);
  - tested with TeXLive 2008, 2009, and 2010 (as of July 2010);
  - code clarified, using key-values (with the `keyval` package) but also `\ifthenelse` (from the `xifthen` package);
- PDF insertion:
  - command `\procpaper` with 1 argument (file name) and 8 optional arguments replaces `\insertprocpaper` with its 9 arguments. This re-design also makes use of key-values style options (using `keyval`);
  - removed the limitation to a minimum of 2 pages and a maximum of 8 pages (now 1 to anything);
- class options:
  - `hyperref`/`geometry`: can now directly pass options to the `hyperref` and `geometry` packages with the new hyperref={option list} and geometry={option list} options;
  - added papers=empty option and mode. This offers a much faster fake paper insertion, compared to papers=draft (draft mode of `pdfpages`, but not checking if pages actually exist);
  - added new options for layout fine tuning and debug:

`\procpaper` `\insertprocpaper` `hyperref` `geometry` `papers=empty` `papers=draft`

`binding` * binding=Xmm to indicate the binding of the paperback version;

`checktitle` `checkauthor` * checktitle and checkauthor to overlay the title and author list onto the 1st page of each paper, for checking the consistency of the table of contents with individual PDF papers;

`showmarginlines` * showmarginlines to draw the margin lines (so that one can match each page fits the template);

`showpapernumber` * showpapernumber to show the paper number below the page number;

`colorheaders` * colorheaders=red to color the header/footer;

`pdftk` * pdftk to output commands for later setting PDF metadata of individual PDFs;

`verbose` * verbose and debug are now different options, and their output texts have been clarified and now makes use of `\PackageInfo` and `\PackageWarning`.

  - options' default values: reset to simplify the most possible the first tests;

- documentation:
  - added a 2-minutes documentation (`confproc-short.tex`) that summarizes commands and options;
  - added a second (and simpler) example;
  - improved & re-organized for incremental learning, and clarified with margin notes and color code (using gray/black color code to show differences between successive versions of the code, and using the 'colordoc' package for color code);
- scripts: added some `bach/Unix` scripts for more functionalities, such as:
  - `buildprocelpb.sh`: for optimized example (generates both paperback and electronic versions of proceedings).
  - `prepareexample.sh`: prepare example files and scripts.
  - `exportIndividualPDFs.sh`: extracts individual paper with new page numbers and proper metadata;
  - `countnbpages.sh`: counts number of pages in each individual PDF paper.

v0.6 undistributed: integrated and enhanced changes made for ICMC 2009;

v0.5 undistributed: hacked version with `kvoptions` plus many fixes for ICMC 2009;

v0.4e
- enhance package by redefining book commands;
- fixed several issues;
- define page layout with the `geometry` package (thanks to Will Robertson);

v0.4d changes history: re-organized using `macro` environment (shorter and clearer);

v0.4c bug correction: author is back in the bookmark (disappeared in v0.4a);

v0.4b
- debug: `\hypersetup` evaluated only at the document beginning (then taking into account the user changes in the PDF metadata);
- remove formatting from footer and name-like commands: author, title, etc. (suggested by Will Robertson);
- use `mathptmx` package instead of `times` package (thanks to Will Robertson);
- use `nth` package instead of `\textsuperscript` command (thanks to Will Robertson);
- redefine `\thebibliography` to avoid inserting a phantom item to set the introductory paragraph (thanks to Will Robertson);

v0.4a
- allows to insert 1-page long papers (did not work in v0.3 and previous);

- instead of replacing each paper's last page by the list of its bibliography items, print them on top of the header of the last page;
- incorporate font style changes to the class: redefining the `\mainmatter`, `\backmatter`, `\thebibliography`, `\thecontents` commands (thanks to Will Robertson);

v0.3 first released version of the package.

v0.2e first distributed version of the scripts.

v0.1 first version of the scripts (DAFx-06).

1.5 To do / bugs

At this time this package offers many more features than the original scripts did. It looks 'complete' to me, and fully functional as is. I however would like to debug/add the following functionalities:

- backward incompatibility: citation items lost if using old command `\procinsertpaper` (up to version 0.5)
- bibliography: fix the right-flush issue that sometimes happen for a small number of back-references in the bibliography, where 1 or 2 or a longer list of back-referrences are placed onto a next line whereas there is enough space on the previous line.
- index/bibliography: correct the pdf link. Clicking on that link should go to the top of the page of the index/bibliography, and not to a position just below the `\indexname` in the text.
- bookmarks/TOC: find a mechanism to customize the table of contents bookmark entry by setting the argument of `\pdfbookmark[0]{Program}{contents}` (does not work yet because `\pdfbookmark` does not accept commands as arguments).
- packages: provide a mechanism for inserting packages *before/after* the `hyperref` package, directly from the example file without having to hack the class (including the `hyperref` package with `\AtBeginDocument` did not work).
- TOC: handle programs with parallel sessions (table of contents);
- use a makefile instead of buildproc.sh, buildprocelpb.sh (that would looks a bit more professionnal);
- generate `confproc-short.tex` from `confproc.dtx`;
- write a shorter version of `confproc.dtx` that does not loose any information, but rather looses useless redundancies and gains in ease of readinf or finding specific information;
- miscellaneous: fix bugs, misspellings, etc. (never ending task).

N.B.: none of those functionalities/improvements is written on my agenda...

1.6 Thanks

Big thanks go to various people: Gary Scavone for offering me to edit the ICMC 2009 proceedings; Andreas Matthias for suggesting in 2008 to simplify the options using `keyval` (yes, I'm quite slow); Philippe Depalle for offering me to edit the DAFx-06 proceedings; Julien Boissinot for asking "Why don't you make it a class?" in 2006; Eoin Brazil for being so enthusiastic about the package; Will Robertson for suggesting many improvements (v0.4a–e) while writing a PracTEX description of the package [16]; and the courageous guinea-pigs of the first versions (Michael James

Ventura, Gary Scavone, Sylvain Marchand, Marije Baalman, Christian Klünder and Christian Frisson). I also thank Andre Dierker for his example `fbithesis.dtx`[1] that I used as a start to build this `confproc.dtx` package/documentation file; his example helped me so much to learn all this dtx stuff.

[1] `fbithesis.dtx`: CTANmacros/latex/contrib/fbithesis/

2 Installation

2.1 Steps summary

1. check that you have all required packages (see sec. 2.2). Using any TEXLive distribution ensures nothing else has to be installed!

2. generate the documentation: '`latex confproc.dtx`';

3. generate the `confproc.cls` file: '`latex confproc.ins`';

4. finish the documentation: '`latex confproc.dtx`' (two times).
   N.B.: the `buildcls.sh` script implements step 2–4 (+ generates index and bibliography).

5. option: move `confproc.cls`, `confproc.pdf` and `exampleN.tex` and all the other example-related generated files into another folder (*i.e.* `examples/`).
   N.B.: the `prepareexample.sh` bash/Unix script prepares example files for you (see sec. 7.3).

Installation steps are explained with more details in sec. 2.3.

2.2 Packages and compiler

Some packages are required for `confproc` to work. If using a full TEX distribution such as TEXLive 2008–2010, all necessary packages (including `confproc`, which installation process is described in the next section) are already installed on your system. Otherwise, you'll find the most recent versions at CTAN[2], at the address provided below for each package with its version number that corresponds to the TEXLive 2008 distribution. Note that `confproc` has also been successfully tested with TEXLive 2009 as of Sept 14, 2009, and preTEXLive 2010 in July 2010.

2.2.1 Essential packages required by `confproc`

1. LATEX 2ε (at least 1994/12/01) and pdfTeXk Version 3.1415926-1.40.9 (Web2C 7.5.7);
   CTAN: macros/latex/base

2. `book` (at least 2005/09/16 v1.4f): standard document class on which `confproc` is based;
   CTAN: macros/latex/unpacked/

3. `kvoptions` and `kvoptions-patch` (at least 2009/04/10 v3.1): key-value class options management; CTAN: macros/latex/contrib/oberdiek/kvoptions.dtx

4. `xifthen` (at least 2009/04/17 v1.3): clearer code for if/then tests;
   CTAN: macros/latex/contrib/xifthen/

5. `pdfpages` (at least 2009/02/07 v0.4g): to include the proceedings articles as PDF documents;
   CTAN: macros/latex/contrib/pdfpages/pdfpages.dtx

6. `hyperref` [10] (at least 2009/05/01 v6.78r): to add hypertext links in the PDF file;
   CTAN: macros/latex/contrib/hyperref/hyperref.dtx

7. `geometry` (at least 2008/12/21 v4.2): to simplify the page layout settings;
   CTAN: macros/latex/contrib/geometry/geometry.dtx

8. `color` (at least 2005/11/14 v1.0j): to provide color links with `hyperref`;
   CTAN: macros/latex/required/graphics/color.dtx

[2] Comprehensive TEX Archive Network: www.ctan.org/

9. `fancyhdr` (at least 2005/03/22 v3.2): to set the proceedings header/footer so as to match the paper template style, if any; CTAN: macros/latex/contrib/fancyhdr/fancyhdr.sty
10. `index` (at least 2004/01/20 v4.2beta): to produce the index of authors;
    CTAN: macros/latex/contrib/index/index.dtx
11. `tocbibind` (at least 2003/03/13 v1.5g): to change the `\indexname` command and disable automatic insertion of index in the table of contents;
    CTAN: macros/latex/contrib/tocbibind/tocbibind.dtx
12. `titletoc` (at least 2007/08/12 v1.6): to change the table of contents layout
    CTAN: macros/latex/contrib/titlesec/titletoc.sty
13. `multitoc` (at least 1999/06/08 v2.01): to provide a two column table of contents
    CTAN: macros/latex/contrib/ms/multitoc.dtx
14. `multicol` (at least 2006/05/18 v1.6g): to provide multi-column index of authors
    CTAN: macros/latex/required/tools/multicol.dtx
15. `sectsty` (at least 2002/02/25 v2.0.2): `\chapterfont` used to give the same headers/footers to the table of contents; CTAN: macros/latex/contrib/sectsty/sectsty.dtx

2.2.2 Other packages successfully used with `confproc` in the examples

1. `hypcap` (at least 2006/02/20 v1.5): for proper hyperref anchors to table and figure captions;
   CTAN: macros/latex/contrib/oberdiek/hypcap.dtx
2. `graphicx` (at least 1996/08/05 v1.0a): to include logos with `\includegraphics`;
   CTAN: macros/latex/required/graphics/graphicx.dtx
3. `newapa` (at least 1991/06/13 v2.0): for the general bibliography
   (N.B.: it is slightly modified after insertion);
   CTAN: biblio/bibtex/contrib/newapa/
4. `newapave` (at least 2006/07/31 v2.1), included in the `confproc` package: DAFx-06 style for the general bibliography (year at the end, before the right-flushed back-references);
   CTAN: macros/latex/contrib/conferences/confproc/
5. `setspace` (at least 2000/12/01 v6.7): to change the line spacing of welcome letters;
   CTAN: macros/latex/contrib/setspace/setspace.sty
6. `inputenc` (at least 2006/05/05 v1.1b): to change the input encoding, for instance to run LaTeX on a document with accents (for the authors' names and the paper titles);
   CTAN: macros/latex/base/inputenc.dtx
7. `fontenc` (at least 2005/09/27 v1.99g): to change the font encoding;
   CTAN: macros/latex/unpacked/fontenc.sty
8. `mathptmx` (at least 2005/04/12 PSNFSS-v9.2a): to change the default LaTeX font to 'Times' for a better PDF display; CTAN: macros/latex/required/psnfss/
9. `nth` (at least 2002/02/27): to use superscript ordinals in the proceedings name (9^{th});
   CTAN: macros/generic/misc/nth.sty
10. `layout` (at least 2000/09/25 v1.2c): to finely tune you document header and footer so that they match those of the paper templates; CTAN: macros/latex/required/tools/layout.dtx
11. `layouts` (at least 2004/10/25 v2.6c): to check the fine tuning of the table of contents layout (N.B.: if inserted too early, the table of contents layout will not properly display);
    CTAN: macros/latex/contrib/layouts/layouts.dtx

2.3 Installation steps

Download this package at CTAN: macros/latex/contrib/conferences/confproc/. It should already contain the following files:

- package-related files:
  - `confproc.dtx`: main package file (the first LaTeX run `pdflatex confproc.dtx` generates various other files)
  - `readme.txt`: package introduction file;
  - `buildcls.sh`: bash/Unix script to generate the class files and documentation (LaTeX run: `pdflatex confproc.dtx`);
  - `confproc.ins`: package file generated from the first LaTeX run, used to generate example files, index styles, etc (LaTeX run: `pdflatex confproc.dtx`);
  - `confproc.cls`: package class generated from the first LaTeX run, used by all example files;
  - `newapave.bst`: bibliography style for confproc;
  - `newapave.sty`: bibliography style for confproc; generated under the `newapave2.sty` name and renamed by the `buildcls.sh` script.
  - `confproc1.ist` and `confproc2.ist`: examples of index style files;
- package documentation:
  - `confproc.pdf`: full documentation (this file); use the `makedoc.sh` bash script to regenerate it with proper index, version history, bibliography;
  - `confproc-short.pdf`: summary documentation;
  - `confproc-short.tex`: summary documentation (original LaTeX file);
  - `confproc_diag.pdf`: figure used in the documentations, which illustrates proceedings building and compilation steps;
  - `makedoc.sh`: bash/Unix script to generate the package full documentation;
- package example:
  - `papers/`: folder with example files (papers);
  - `pictures/`: folder with an example of proceedings cover.

The provided `confproc.dtx` file is an 'one-file-contains-it-all': it contains the `.cls` class file, its `.pdf` documentation, a customizable driver for the documentation, the `.ins` batch file, a complete example, and a 'read me' file.

To install the package:

1. run `confproc.dtx` through LaTeX. This will generate the batch file `confproc.ins` (only if it does not already exists: if willing to re-generate it, first delete the one that pre-exists) and a `readme.txt`. Additionally the documentation (`confproc.pdf`) is generated (to get the cross-references right, you have to rerun this twice, however).

2. run the newly generated `confproc.ins` through LaTeX to do the actual installation. This will (re)generate the `confproc.cls` class file, the example files[3] and example-related files[4],

[3] Example files: `example1empty.tex`, `example2custom.tex`, `example3optim.tex`, `example4optim.tex`

[4] Example-related files: `exclasspre.tex`, `exclasslastel.tex`, `exclasslastpb.tex`, `expages.tex`, `exsessions.tex`, `expapersswitch.tex`, `exbiblio.bib` and `exprogram.csv`

build scripts[5], the documentation driver (`confproc.drv`) and a sample configuration file (`confproc.cfg`).

3. to finish the installation it is recommended to move the documentation (`confproc.pdf` and `confproc-short.pdf`) and the example-related files to where you collect the documentations. With a TDS compliant LaTeX installation this would for example be: `$(TEXMF)/doc/tex/latex/confproc`

4. to use the examples:

   - move all `ex*.*` files together with the example scripts[6] and the `papers/` and `picture/` folders to the `example/` folder (thus hiding all the mess in the same place ;-));
   - place in `example/` a copy of the following class files: `confproc.cls`, `confproc1.ist`, `confproc2.ist`, `newapave.bst`, `newapave.sty`;
   - create a `pdftk_info/` folder there, and move the `expages.tex` file into `papers/`;
   - change the permissions to 'execute' (`chmod +x ...`) for example scripts[7].

   N.B.: the `prepareexample.sh` bash/Unix script (see sec. 7.3) does it all for you.

5. for a demonstration of the possibilities of `confproc` see the `example*.tex` file and run them through LaTeX. For a more complete demonstration, use the `buildproc.sh` (see sec. 7.5) and `buildprocelpb.sh` (see sec. 7.6) and bash/Unix scripts, that will make for you all the necessary steps to provide the final version of the example proceedings.

The '`latex confproc.dtx`'-run above will by default generate the full documentation (with complete listing of the documented source code, command index and change history). If you need the 'user' documentation, you may edit `confproc.drv` to meet your needs (never edit `confproc.dtx` itself!). For more information on the enhanced documentation see `confproc.drv` or `readme.txt`.

[5] Class and proceedings build scripts in Perl: `generateswitch.pl`; Unix: `buildcls.sh`, `cleancls.sh`, `buildproc.sh`, `buildprocelpb.sh`, `buildpapers.sh`, `buildcppdfpapers.sh`, `countnbpages.sh`, `exportIndividualPDFs.sh`, `papersinfo.sh`, `paperssplitpreamble.sh`, `prepareexample.sh`, `removeLaTeXcmds.sh`

[6] Perl: `generateswitch.pl`; Unix: `buildproc.sh`, `buildprocelpb.sh`, `buildpapers.sh`, `buildcppdfpapers.sh`, `countnbpages.sh`, `exportIndividualPDFs.sh`, `papersinfo.sh`, `paperssplitpreamble.sh`, `prepareexample.sh`, `removeLaTeXcmds.sh`

[7] Example scripts: `buildproc.sh`, `buildprocelpb.sh`, `generateswitch.pl`, `exportIndividualPDFs.sh`, `papersinfo.sh`, `paperssplitpreamble.sh`

3 Example 1 (example1empty.tex) A short introduction to confproc (2 minutes)

For people that are very excited about this package, here is a 2-minute guide, that gives a quick feeling of *what can be done with confproc*. Next two sections provide a detailed description of the package (sec. 4) as well as a full working example with index, general bibliography, etc. (sec. 5). A good advice that could be given is "Do as you would do for a book": table of contents, headers and footers, index, etc.

3.1 Preamble

Consider we are writing a book with several parts (days) and chapter (sessions). Then, we want to set its title, author, a table of contents (or list of days/parts, sessions/chapters and papers/sections), as well as an index. Then, using the same usual commands, the proceedings will use the code provided below.

3.1.1 Document class and options

We first insert the document class with reasonable options (that we will not look at right now):

```
⟨*example1empty⟩
\documentclass[letterpaper,10pt,twoside,%
  electronic,% [printed] | electronic
  papers=countpages,% empty | draft | [final] | countpages
  paperselec=all, %[all] | p_001 | p_fake
  hyperref={bookmarksdepth=1,bookmarksopen,bookmarksopenlevel=0,%
    linkcolor=blue,urlcolor=blue},%
  geometry={text={175truemm,226truemm},% A4 & letter
    inner=0.805in,top=29.15mm,bottom=24.5mm,footskip=9.68mm,voffset=-5mm},%letter
]{confproc}
```

3.1.2 Packages

We then insert packages for input and output font encoding, Times font selection and miscellaneous:

```
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{mathptmx}
\usepackage[super]{nth}

```

3.1.3 Customization

We redefine proceedings-specific commands to customize the document to our needs:

```
\renewcommand{\procpdfauthor}{{\color{red}[Proceedings editor], [University]}}
\renewcommand{\procpdftitle}{{\color{red}[Acronym] Proceedings}}
\renewcommand{\procpdfsubject}{{\color{red}Proc. of the Xth International Conference %
    on [Nice Topic] ([Acronym]), [City], [Country], [Dates]}}

\renewcommand{\procchead}{} %
\renewcommand{\proclhead}{{\em \small \procpdfsubject}}

\author{\procpdfauthor}
\title{\procpdftitle}
\date{\today}
```

We also define the paper path:

```
\renewcommand{\PAPERPATH}{papers/}
```

3.2 Front matter: cover page, index and table of contents

Ready to start? We generate the index and start the document front matter:

```
\makeindex

%%%==========  PROCEEDINGS  ==========
\begin{document}
\frontmatter
\setcounter{page}{1}
\pdfbookmark[0]{Preamble}{preamble}
\pdfbookmark[1]{Cover}{cover}
\maketitle
\newpage

```

It is now time to insert the conference program:

```
\otherpagestyle
\tableofcontents

```

3.3 Main matter: the papers

We then switch to the list of papers, organized by day and session:

```
%%%==== BEGINNING OF PAPERS ====
\mainmatter

\procday{Day 1}
  \session{Oral Session 1}
    \procpaper[switch=45,%
      title={Templates for One Author},%
      author={Alfred Alabama},%
      index={\index{Alabama, Alfred}},%
    ]{p_001}
    \procpaper[switch=21,%
      title={Templates for One Author with Two Affiliations},%
      author={Bob Boogie-Woogie},%
      index={\index{Boogie-Woogie, Bob}},%
    ]{p_003}
  \session{Poster Session 1}
    \procpaper[switch=33,%
      title = {Templates for Two Authors},%
      author={Alfred Alabama, Chris Christmas},%
      index={\index{Alabama, Alfred}\index{Christmas, Chris}},%
    ]{p_005}

\procday{Day 2}
  \session{Oral Session 2}
    \procpaper[switch=75,%
      title={Templates for Three Authors},%
      author={Bob Boogie-Woogie, Chris Christmas, Don Didon},%
      index={\index{Boogie-Woogie, Bob}\index{Christmas, Chris}%
        \index{Didon, Don}},%
```

```
  ]{p_007}
  \procpaper[switch=27,%
    title={Templates f\'or F\‘o\"ur ÃĂuthors},%
    author={J\o{}hn J\"oe, K\’e\~{n}t K\^{\i}ng, L\‘ou L\’ou,%
      M\‘anfr\’ed J. M\^ost\u{e}k\i},
    index={\index{J\"oe, J\o{}hn}\index{K\^{\i}ng, K\’e\~{n}t}%
      \index{L\’ou, L\‘ou}\index{M\^ost\u{e}k\i, M\‘anfr\’ed J.}},
  ]{p_009}

%%%==== END OF PAPERS ====
```

3.4 Back matter: index of authors

We are almost done: we finish by inserting the authors' index before closing the document!

```
\backmatter
\insertindex
\end{document}
⟨/example1empty⟩
```

3.5 LaTeX runs

To build this example, run the following LaTeX steps:

1. generates the first `.aux` and `.idx` files (use option papers=countpages as the paper's number of pages are not specified):
   `pdflatex example1empty.tex`
2. generates the author index:
   `makeindex -s confproc2.ist example1empty.idx`
3. inserts table of contents and index, and update their page numbers for next run (use option papers=countpages again):
   `pdflatex example1empty.tex`
4. final LaTeX run inserting table of contents and index with proper page numbers; useful only if the table of contents is longer than a single page (papers=coutnpages):
   `pdflatex example1empty.tex`

4 Example 2 (example2custom.tex) A not so short introduction to confproc (2 hours or so)

Here is provided exhaustive information about *what can be done* with confproc, but also *how to do so*, as well as an example file (example2custom.tex) that makes use of this information. It requires an hour or so to go through. This file is based on example1empty.tex, and adds all the necessary customization so that the proceedings now look great. We present its code: grey parts are identical to example1empty.tex, whereas black parts are the modified lines.

4.1 Document class loading

The class is loaded with:

```
\documentclass{confproc}
```

To modify the default behavior of confproc, use options:

```
\documentclass[<options>]{confproc}
```

All available options are described below in subsection 4.2.

4.2 Options

There are two types of options:

- some are specific to the confproc class (sometimes also passed to other packages),
- others are simply passed to the book class, the hyperref or pdfpages packages.

A summary of all options is given in Tab. 1 and 2. Note that *for alternative options values (indicated as a list of items separated by vertical bars: |), the default value appears between squared brackets.*

4.2.1 Log and options

New [v0.7]

With the verbose=true option, the confproc package prints the following text, that shows the current option settings is always added to the log window:

```
----------------
| | Document formatting:
| | ____ letterpaper
| | ____ twoside=true
| | ____ twosidepapers=true
| | Proceedings-specific formatting:
| | ____ electronic=true (file version)
| | ____ binding=(for printed version)
| | ____ papers=final(paper insertion)
| | ____ headers=allpages(header add to pages)
| | List of papers:
| | ____ paperselec=all
| | Lists (toc, bib, index):
| | ____ twocoltoc=false (=onecoltoc)
| | ____ tocnum=left
| | ____ twocolbib=true
| | ____ bib=none
| | ____ twocolindex=true
| | Help for layout design:
| | ____ checkauthor=false (do not add author list to 1st page)
```

```
| | ____ checktitle=false (do not add title to 1st page)
| | ____ showpapernumber=false (do not add paper number)
| | ____ movepagenumber=false (do not move paper number)
| | ____ showmarginlines=false (do not add template format)
| | ____ colorheaders=black(color for header/footer)
| | Verbose:
| | ____ debug=true (for hyperref)
| | ____ verbose=false (for confproc+hyperref)
| | ____ pdftk=false (for use with pdftk to add PDF metadata)
| | passed to hyperref: bookmarksdepth=1,bookmarksopen,bookmarksopenlevel=0,lin
kcolor=colorforlink,urlcolor=colorforurl
| | passed to geometry: text={175truemm,226truemm},inner=0.805in,top=29.15mm,bo
ttom=24.5mm,footskip=9.68mm,voffset=-5mm
----------------
```

4.2.2 Document formatting (book-specific)

The following options define some of the general document formatting via the `book` class:

`a4paper` `letterpaper`

- [a4paper] | letterpaper selects the paper format: A4 (European) or letter (North American). The option is passed to the `book` and `hyperref` packages.

`10pt,11pt,12pt`

- [10pt] | 11pt | 12pt selects the font size.

`twoside` `oneside`

- [twoside] | oneside selects two-side or one-side documents. By default, two-side documents have each new chapter and paper starting on odd & right pages. This means that papers with odd number of pages will have an extra blank page at the end: they all start on a right page (easier to find/navigate) but this does not save paper. To change this behavior for papers only, use the onesidepapers option.

- [twosidepapers] | onesidepapers forces individual papers to be in two-side or one-side mode.

`twosidepapers`

  – With twosidepapers, a blank page is added to each individual paper with an odd number of pages, so that all the papers start on a right page (with odd number) even though the document is oneside: this helps a lot to quickly browse through the paperback proceedings, while increases the number of total pages (environmental cost).

`onesidepapers`

  – Conversely, with onesidepapers, it allows to produce a two-side document (twoside) except for the papers, that are not separated by blank pages when having an odd number of page.

  N.B.1: When this option is not used, its default value will match the one of the whole document (ie. twosidepapers is the default if twoside is selected; and onesidepapers is the default when oneside is selected).
  N.B.2: before version 0.5, twosidepapers was called cleardoublepage, and onesidepapers was called clearsinglepage.

4.2.3 Proceedings-specific formatting

Depending on wether the proceedings are generated in their printed (paperback) or electronic (PDF) document, color links may be disabled[8], a binding, headers and footers on some specific pages. Also, we may decide to insert the papers in different ways depending on what we work on. All the hyperlink features work properly by default thanks to `hyperref`, so the only options to set are:

`electronic` `printed`

- [electronic] | printed: the electronic version has user-defined colors for links (same as default

[8] Color pages are much more expensive to be printed, and the color text readibility is reduced when printed in a grey scale.

colorlinks=true option of the pdfpages package), whereas the printed version have black links (same as using colorlinks=false for the pdfpages package) so that they do not appear;

binding

- binding=XXmm (default: 0mm, positive value otherwise) may only used for a paperback version[9]. The binding corresponds to the amount of horizontal (x) shift towards the external side of the proceedings (left and right margins then differ for odd and even pages in two-side mode). The binding value may depend on the thickness of the final document, *e.g.* 2mm for a 1cm thickness, 3mm for a 1.5cm thickness, 5mm for a 3cm thickness, etc.

papers

- papers=[final] | draft | empty | countpages changes the way papers are inserted: *this option is very important to set, as it changes the speed of the LaTeX run*.

papers=final

  – papers=final inserts each PDF page (using pdfpages) from 1 to the number of pages indicated by the user, resulting in a slow LaTeX run. Use this option for instance when working on the layout and on the bibliography merging process (sec. 5.7).

papers=draft

  – papers=draft fakes paper insertion and checks page existence (by pdfpages), resulting in a (slightly) faster LaTeX run. It replaces each PDF page by an almost blank page after checking that this particular page exists. This is slightly faster that papers=final or papers=countpages, but not as fast as papers=empty. It basically is useful when editing the preamble (cover page, edition information, welcome letters), working on generating the table of contents or the index of authors, or generating proper page numbering *and willing to ensure each inserted page exist*.

papers=empty

  – papers=empty fakes paper insertion without checking page existence, resulting in a much faster LaTeX run (but not as safe as the previous two). Use it for the same purposed as papers=draft. It also is useful for debug purposes, as it indicates on each page its number in the paper, the author list, the paper title, the paper number and the list of bibliography item.

papers=countpages

  – papers=countpages inserts the whole paper with pdfpages, *i.e.* each page of the paper regardless of the argument npages=XX of \procpaper.
  N.B.: The papers=countpages breaks the bibliography management, so the bib=* option may look unefficient. Unfortunately, there is no other solution that using papers=final if doing a general bibliography.

Important remarks:

New [v0.7]

1. Since version 0.6, bookmark data are properly generated for all the four options.

New [v0.7]

2. For all values except countpages, if no number of pages is indicated at the paper insertion (using npages=5, see sec. 3.3 and 5.2), *only the first page is inserted* by default. Conversely, the number of pages indicated is ignored by the countpages option.
3. Switching from papers=empty to any of the other three results in broken hyper-links after a single LaTeX run. This means that for a working electronic proceedings, 2 compilations are needed with the 'final' option (papers=final or papers=countpages).

New [v0.7]

4. The option pair draft | final is no more passed to pdfpages, but rather used as papers=final|draft. The option pair papers=final | draft is **not** exclusive, so papers=final (instead of the last inserted option) 'win' if using the two. For instance, using:

   ```
   \documentclass[papers=final,papers=draft]{confproc}
   ```

   we expect the last option (papers=draft) to be used. In fact, it is equivalent to:

   ```
   \documentclass[papers=final]{pdfpages}
   ```

[9] Indeed, when browsing throught the PDF, the x-shifts of the binding distracts the reader

So, make sure to only use papers=draft if that's what you need!

headers

- headers=[allpages] | none | onlypdf | exceptpdf defines the pages to which a header/footer is added. Both header and footer are treated in the same way. On user request, I may later add a second option to manage separately headers and footers.

headers=allpages

  - [headers=allpages] adds header/footer to all pages. Use it for instance if the paper templates has no header nor footer. Advice to conference template designer: this is the simplest solution, as there is no need to renumber all papers. You may then design you templates with headers/footers to see the final result, and discard header/footer before distributing the templates, to save a lot of time at the end!

headers=no

  - headers=no does not ad any header/footer to any pages. This is used when willing to add the header/footer with another software (*e.g.* Acrobat) and other fonts and layout.

headers=pdfonly

  - headers=pdfonly: headers only added to PDF-included files; for use if one wants to insert header/footer on the inserted papers only. *But why would one do so?*

headers=exceptpdf

  - headers=exceptpdf: headers added to all pages except PDF-inserted papers.. This is used when the paper template defines a header/footer, that you then have to cope with. Note that this template should definitely have no page number, otherwise you need to edit pages numbers on each single paper.

bib

- bib=[none] | merge | backref | last: defines how the general bibliography is generated.

bib=none

  - bib=none when no general bibliography is generated. The three following values for the bib option are used for generating a general bibliography.

bib=merge

  - bib=merge helps when merging bibliographic items. It only inserts the first and last page of each paper. All citations from the current paper then appear either in the header (together with papers=final | draft) or onto the page (with papers=empty). It also creates back-references from the bibliography to the last page of the papers that use this citation (as for the bib=backref option). It runs LaTeX faster (only 2 pages), but page numbers are incorrect since some pages are missing.

bib=backref

  - bib=backref generates proper bibliography with back-references. It inserts all the pages from 1 to N as indicated by the user. As for bib=merge, all citations from the current paper then appear either in the header (together with papers=final | draft) or onto the page (with papers=empty). Then, page numbers are ok for the table of contents and the authors' index. This requires several LaTeX runs, as you can see in the corresponding Unix script in sec. 7.5. A *final* compilation with the bib=last option is required.
  N.B.: to check the page numbering relatively to the paper, use papers=final option to force inserting the PDF instead of a blank page, together with movepagenumber in the case papers have page numbers.

bib=last

  - bib=last inserts the bibliography that is no more added to the paper last page, which means back-references will disappear at the next LaTeX run. This is why this should be the final run, as far as the bibliography is concerned. This means that the program (paper ordering) is definitively set, the general bibliography is generated (and common items merged), all papers have been re-compiled if necessary (in order to re-number them all, and have them using the new bibliography), and the document has been run through LaTeX 2_ε enough times with the bib=backref option to have proper page numbering and back-references in the table of contents, the authors' index and the general bibliography (see sec. 6.3 and 7.5).

N.B.: A side effect of the papers=countpages is that it breaks the bibliography management: beware that the other current options may look temporarily unefficient.

4.2.4 List of inserted papers

paperselec=[all] | paperselec=p_001 is used when working on each paper insertion to insert a single paper (paper 'p_001' in the example) of all papers. Selecting a single paper allows to check data in particular cases. For instance when the paper length, author list of title has changed, you want to make sure that no change was forgotten when updated them.

4.2.5 Lists layout (table of contents, general bibliography, index)

The next options deal with the layout customization for the table of contents, the index of authors and the general bibliography:

`onecoltoc` `twocoltoc` New [v0.7]
- [onecoltoc] | twocoltoc: prints the table of contents with one/two column(s).
  N.B.: this option pair is boolean, so twocoltoc, twocoltoc=true and onecoltoc=false are equivalent. Also, onecoltoc, onecoltoc=true and twocoltoc=false are equivalent.

`tocnum` New [v0.7]
- tocnum=[left] | right places page numbers on the left / right side of table of contents. IMHO the left side provides a faster click access to individual papers.
  N.B.: this option was named tocnumleft | tocnumright prior to version 0.7.

`twocolbib` `onecolbib` New [v0.7]
- [twocolbib] | onecolbib prints the general bibliography in a two/one column(s) format.
  N.B.: this option pair is a boolean, so twocolbib, twocolbib=true and onecolbib=false are equivalent. Also, onecolbib, onecolbib=true and twocolbib=false are equivalent.

`threecolindex` `twocolindex`
- [threecolindex] | twocolindex prints the index of authors with three or two columns.
  N.B.: this option pair is boolean, so threecolindex, threecolindex=true and twocolindex=false are equivalent. Also, twocolindex, twocolindex=true and threecolindex=false are equivalent.

4.2.6 Help for checking data and layout

Two new options are helpful to check the validity of individual paper data. They are especially useful when either the title or the author list may have changed at some point during the proceedings edition process. Such changes may happen for the preview proceedings (when generating the proceedings from the first submissions, before the reviewing process), and for the final proceedings (that makes use of the final paper submissions). Indeed, authors often make last minute changes to their paper, and forget to let you know that the title and/or has changed!

`checktitle`
- checktitle is a boolean defaulted to false. When set to true, it overlays each paper's title (according to the table of contents data provided by the `\procpaper[title={...}]{}` command) onto the first page of each paper (except for the paper=empty since it is useless).

`checkauthor`
- checkauthor is a boolean defaulted to false. When set to true, it overlays each paper's author list title (according to the TOC data provided by the `\procpaper[author={...}]{}` command) onto the first page of each paper (except for the paper=empty since it is useless), below the title in the case both checktitle and checkauthor are set to true.

A few more options (and related behaviors) are helpful to design the document layout:

`showpapernumber`
- showpapernumber is a boolean defaulted to false that, when set to showpapernumber=true, adds the paper number below the page number. This is useful when editing the proceedings with papers=final | countpages, if one notices that a paper has a formatting problem and want to give some feedback to the author (or correct the problem by him/herself).

`movepagenumber`
- movepagenumber is another boolean defaulted to false that moves the page number a few millimeters below its normal position (the result is only seen when combined with headers=allpages | pdfonly). Two footers then appear: the one from the paper, and below it the one from the proceedings. This allows to check the accuracy of page numbers in the case papers already have a page number in their template (centered in the footer).

showmarginlines

- showmarginlines is another boolean defaulted to false that overlays the user-defined template format (limited to the upper horizontal line and all vertical lines). Use it when setting the horizontal (xshift) and vertical (yshift) alignment of each paper. To adjust the margin lines to a given proceedings template, we need to redefine the \procmarginlines command, which by default is set to:

\procmarginlines

```
\renewcommand{\procmarginlines}{
  \noindent
  \vspace*{7mm} % adjusting vertical initial space
  \begin{table}[h!] % table for vertical lines
    \centering
    \color{blue}
    \begin{tabular}{|@{}p{3.3in}@{}|@{}p{0.3in}@{}|@{}p{3.3in}@{}|} % spacing between colu
      \hline % upper horizontal line
      ~~~~~~~~~~ & ~ &~~~~~~~~~~~~~\\
      \vspace*{7.5in} % less than a page height
      ~~~~~~~~~~ & ~ &~~~~~~~~~~~~~\\
    \end{tabular}
  \end{table}
}
```

colorheaders

- colorheaders=[black] allows to change the header/footer color. this is a nice feature for setting the geometry package options in the case where PDF papers already have their headers (headers=exceptpdf) and one wants to check if papers are properly aligned: then, temporarily use the headers=all option together.

4.2.7 Verbose and pdftk options

The following two options are adding some output text in the log file, for short-term help to the user:

debug

- debug is a boolean option defaulted to false that is passed to the hyperref package.

New [v0.7]
verbose

- verbose is a boolean option defaulted to false that turns debug on and adds confproc-specific messages to the console when inserting the papers:

```
confproc: partial paper insertion (last page=bib items)
______ debug: insert paper ______
confproc/file: p_001.pdf (2 pages)
confproc/title: Templates for One Author
confproc/authors: Alfred Alabama
confproc/index: \p@index{Alabama, Alfred}
confproc/shift: (0.0pt, 0.0pt)
confproc/citations:
confproc/bookmarks:
confproc/switch ID: 45
-----------------------
```

New [v0.7]
pdftk

When generating PDF proceeding, you may want to (re)generate individual PDF papers with proper (and homogeneous) PDF metadata. For this purpose, the pdftk[10] Unix utility is of big help (see sec. 7.11). To generate pdftk data, the pdftk option has been added. It is a boolean option defaulted to false that generates a confproc.pdftk file. When generated, this file contains data for use with pdftk via the papers_info.sh bash script, and they look like:

```
----------------
```

[10] Get pdftk at: http://www.accesspdf.com/pdftk/

```
pdftk A=${PDFFILE} cat A1-3 output ${SPPATH}/p_001.pdf
gs -dBATCH -dNOPAUSE -q -sDEVICE=pdfwrite -dFirstPage=1 -dLastPage=3 -sOUTPUTFILE=${SPPATH}/p_001.
InfoName: p_001.info
InfoKey: Title
InfoValue: Templates for One Author
InfoKey: Author
InfoValue: Alfred Alabama
InfoKey: Subject
InfoValue: DAFx-06 Conference
InfoKey: Producer
InfoValue: pdftk 1.12 + Ghostscript 8.71
InfoKey: Creator
InfoValue: LaTeX2e + confproc 0.7
InfoEnd
```

When pdftk is combined with verbose, this text also appears in the log window.

pdftkfolder — pdftkfolder ={path/to/pdftk/files/} sets the folder where pdftk files are written string (default: '.', ie. the current folder).

pdftksubject — pdftksubject={DAFx-06 Conference} sets the subject string (default: Conference).

pdftkproducer — pdftkproducer={producer} sets the producer string (default: 'pdftk 1.12+ Ghostscript 8.71').

pdftkcreator — pdftkcreator={creator} sets the creator string (default: LaTeX2e + 'confproc 0.7').

4.2.8 Options for the hyperref package

The confproc package being based on the hyperref package for all PDF and links, all the hyperref option values can be customized. This is a good thing for fine tuning your document, but at your own risks if you do not read the corresponding documentation [10]. To help the reader, next paragraphs indicate which options should be customized (sandbox), and the following paragraph the options that should not be modified (expert side).

New [v0.7]

Mechanism to set options Up to version 0.5, unknown options used with the confproc package were passed to the hyperref package. While this is still possible, we strongly recommend to pass options to hyperref using the hyperref={option list}, since only this mechanism will be allowed in future versions. For instance, default values are:

hyperref

```
hyperref={colorlinks=true,linkcolor=red,citecolor=blue,urlcolor=blue,%
  bookmarksopen=true,bookmarksopenlevel=1}%
```

The sandox Some of hyperref options should be customized to each one's needs, for instance:

colorlinks

- colorlinks=true or colorlinks provides color links in the table of contents, index of authors and general bibliography to the corresponding pages in the proceedings. This option has the same effect as the electronic option from the confproc package.
- colorlinks=false provides links without color, which is particularly helpful for printed proceedings (where using color increases the cost of printing, or reduces the quality if printed in black and white). This option has the same effect as the printed option from the confproc package.

citecolor

- citecolor=colorforcite uses the color colorforcite (to be defined by the used) for links to bibliography items cited;

linkcolor

- linkcolor=colorforlink uses the color colorforlink for links, such as from the index of authors, table of contents and general bibliography back-references;

urlcolor

- urlcolor=colorforurl uses the color colorforurl for URL, mainly in the general bibliography but also in the publishing information, for example;

debug

- debug prints more information from the `hyperref` package;

a4paper,letterpaper

- a4paper | letterpaper are passed to `hyperref`;

bookmarksopen

- bookmarksopen=[true] | false: opens/closes the bookmarks of the PDF file.
  N.B.: a change of this option will only be reflected in the PDF after two pdfLaTeX runs.

bookmarksopenlevel

- bookmarksopenlevel=0 | [1] | 2: bookmarks are opened at level 0 (parts=days), 1 (chapters=sessions) or 2 (sections=papers). The 'session' level is a good choice, as they are not too many sessions, so all lines should display at once when opening the PDF, whereas the 'paper' level would require more lines than papers (100? 200?).
  N.B.: a change of this option will only be reflected in the PDF after two pdfLaTeX runs.

The expert side Other `hyperref` options change hyperlink properties such as back-references. Their values are then given by default to the `hyperref` package to ensure the electronic version of the proceedings is ok. They should be changed with a lot of care[11]:

pdftex

- pdftex: to set up `hyperref` for use with the pdftex program.

raiselinks

- raiselinks: in the hypertex driver, the height of links is normally calculated by the driver as simply the base line of contained text; this options forces `\special` commands to reflect the real height of the link (which could contain a graphic).

hyperindex

- hyperindex: makes the text of index entries into hyperlinks. It is used for the index of authors, to link back to their various papers.

backref

- backref: allows for back-references in the general bibliography.

pagebackref

- pagebackref: adds 'backlink' text to the end of each item in the bibliography, as a list of page numbers (this can only work properly if there is a blank line after each `\bibitem`).

plainpages

- plainpages=false: forces page anchors to be named by the arabic form of page number, rather than the formatted form. This is useful since `confproc` uses the `book` class, and uses a front matter (publishing information, welcome letters, table of contents, etc) before inserting papers.

pdfpagelabels

- pdfpagelabels: sets PDF page labels, to be able to link to them.

breaklinks

- breaklinks: allows links to break over lines by making links over multiple lines into PDF links to the same target. This is particularly useful for 2-columns table of contents with the option linktocpage=false (not the default); and for long URLs in the general bibliography.

linktocpage

- linktocpage: makes page number (instead of text) to be the link to table of contents (as well as list of figures and list of tables, but they are not often used for proceedings).

pdfstartview=XYZ

- pdfstartview=XYZ: opens the PDF in Acrobat with zoom=100% instead of full screen; especially useful if working with a big screen (*e.g.* 30 inches).

4.2.9 Passed to geometry

geometry

New [v0.7]

As for `hyperref`, an option passed options to the `geometry` package: geometry={option list}. This simplifies a lot the setting of the proceedings layout variables, that previously was poorly

[11] Please refer to the `hyperref` documentation [10] for more complete, accurate and up-to-date descriptions.

managed (part of it was hard-coded in the `confproc.cls` file, and the left-over neglected and left to the user's skills). Default values are:

```
geometry={text={6.9in,9in},inner=0.8in,top=1in,bottom=1in,%
  headsep=7.05mm,footskip=10mm,voffset=-5mm}
```

4.2.10 Default option values

By default, the set of options used (if not defined by the user) produces a 2-side 10pt and letter size final electronic proceedings (ie. no debug) without binding and with blue links:

- document formatting, also passed to book: letterpaper, 10pt, twoside;
- proceedings-specific formatting: twosidepapers, electronic, binding=0mm, papers=final, headers=allpages, bib=none;
- list of inserted papers: paperselec=all;
- lists formatting: onecoltoc, tocnum=left, threecolindex, twocolbib;
- checking layout and paper data: checktitle=false, checkauthor=false, showmarginlines=false, showpapernumber=false, movepagenumber=false, colorheaders=black;
- printing package debug: verbose=false, debug=false;
- generating the `pdftk` data file: pdftk=false;
- passed to `hyperref`: colorlinks=true, linkcolor=red, citecolor=blue, urlcolor=blue, bookmarksopen=true, bookmarksopenlevel=1;
- passed to `geometry`: text=6.9in,9in, inner=0.8in, top=1in, bottom=1in, headsep=7.05mm, footskip=10mm, voffset=-5mm.

4.2.11 Back to the example: loading the cass with options

```
⟨*example2custom⟩
\documentclass[letterpaper,10pt,twoside,%
  electronic,% [printed] | electronic
  papers=final,% empty | draft | [final] | countpages
  headers=exceptpdf,% none | pdfonly | exceptpdf | [allpages]
  paperselec=all, %[all] | p_001 | p_fake
  colorheaders=red,%
  verbose,%
  pdftk,%
  pdftkfolder={pdftk_info/},%
  pdftksubject={DAFx-06 Conference},%
  hyperref={bookmarksdepth=1,bookmarksopen,bookmarksopenlevel=0,%
    citecolor=colorforcite,linkcolor=colorforlink,urlcolor=colorforurl},%
```

4.2.12 Document layout

We used the following `geometry` options for tuning page attributes:

```
  geometry={text={175truemm,226truemm},% A4 & letter
    inner=0.805in,top=29.15mm,bottom=24.5mm,footskip=9.68mm,voffset=-5mm},%letter
]{confproc}

```

so that the proceedings layout can perfectly match the one of individual papers. This means that you have to check for those values in your template.

Option / category	Package(s)	Function
Document formatting		
[letterpaper] | a4paper	`hyperref`, `confproc`	European A4 | North American paper format
[10pt] | 11pt | 12pt	`book`, `confproc`	set normal font size
[twoside] | oneside	`book`, `confproc`	for two/one-side documents (2-side: new chapters start on odd & right pages)
[twosidepapers] | onesidepapers	`confproc`	for papers to be locally considered as two/one-side documents
Proceedings-specific formatting		
[electronic] | printed	`confproc`	links with/without colors. Identical to colorlinks=true | false from `pdfpages`
binding=Xmm [0mm]	`confproc`	amount of horizontal space added to printed version (independent from printed)
papers=[final]	`confproc`	inserts PDF papers from page 1 to N (slow)
draft	`pdfpages`, `confproc`	fakes PDF papers inclusion from page 1 to N, but checks page existence (faster)
empty	`confproc`	fakes PDF papers inclusion from page 1 to N without checking page existence (fastest)
countpages	`confproc`	include full PDF papers (slowest; ignore N and breaks blibliography)
headers=[allpages]	`confproc`	header/footer for all pages
pdfonly	`confproc`	header/footer only for PDF papers
exceptpdf	`confproc`	header/footer for all pages except PDF papers
none	`confproc`	no header/footer for any pages
bib=[none]	`confproc`	no general bibliography
last	`confproc`	for the final compilation with general bibliography (breaks back-references)
merge	`confproc`	only includes 1st and last page (faster run) for merging bib items in the general bibliography
backref	`confproc`	prepares back-references before final run
List of inserted papers		
paperselec=[all] | p_001...	`confproc`	indicate the papers to be inserted (1 or all)
Lists formatting		
[onecoltoc] | twocoltoc	`confproc`	one/two column(s) table of contents
tocnum=[left] | right	`confproc`	left/right page numbering table of contents
[onecolbib] | twocolbib	`confproc`	one/two column(s) general bibliography
[threecolindex] | twocolindex	`confproc`	three/two columns index of authors
Help for checking data and layout		
checktitle ([false])	`confproc`	overlays the title onto each paper's 1st page
checkauthor ([false])	`confproc`	overlays the author list onto each paper's 1st page
showpapernumber ([false])	`confproc`	adds paper number below page number
movepagenumber ([false])	`confproc`	moves page number down by a few millimeters
showmarginlines ([false])	`confproc`	shows margin lines of paper template
colorheaders=[black]	`confproc`	changes color of header/footer

Table 1: *List of options 1/2*

Option / category	Package(s)	Function
Verbose and `pdftk` options		
debug ([false])	`hyperref`	sets debug=true for `hyperref`
verbose ([false])	`confproc`	sets debug=true + adds `confproc` specific debug
pdftk ([false])	`confproc`	generates confproc.pdftk with `pdftk` commands for metadata of individual papers
pdftksubject ([Conference])	`confproc`	set PDF subject metadata for `pdftk`
pdftkproducer ([pdftk 1.12 + Ghostscript 8.71])	`confproc`	set PDF producer metadata for `pdftk`
pdftkcreator ([LaTeX2e + confproc 0.7])	`confproc`	set PDF creator metadata for `pdftk`
Passed to `hyperref` using hyperref={option list}		
backref	`hyperref`	add reference page number and link to each bibliographic item
breaklinks	`hyperref`	allows links to break over lines by making links over multiple lines into PDF links to the same target (great for table of contents and bibliography in two columns)
citecolor=[blue]	`hyperref`	use the user-defined colorforcite color for links to bibliography items cited
colorlinks=[true] | false	`hyperref`	links without/without colors. Equivalent to electronic | printed
hyperindex	`hyperref`	author index entries pages = hyperlinks to each paper
linkcolor=[red]	`hyperref`	color to use for links (from index, TOC, and bibliography back-references)
linktocpage=[true]	`hyperref`	TOC link is the page number, not the text
pdfpagelabels=[true]	`hyperref`	set PDF page labels: compulsory for creating any link to page!
pdfstartview=[XYZ]	`hyperref`	open the PDF file in Acrobat with zoom=100% instead of full screen
pdftex=[true]	`hyperref`	set up `hyperref` for use with pdfTeX
plainpages=[false]	`hyperref`	forces page anchors to be named by the arabic form of the page number, rather than the formatted form
raiselinks=[true]	`hyperref`	forces `\special` commands to reflect the link real height (may contain a graphic)
urlcolor=blue	`hyperref`	use the blue color for URL (general bibliography, publishing information)
Passed to `geometry` using geometry={option list}		
text={height=21cm,width=15cm}	`geometry`	see `geometry`

Table 2: *Alphabetical list of all options 2/2*

4.2.13 Example: loading extra packages

Then, we define the other packages to be used.

Important note: any package that redefines LaTeX macros should be inserted before `hyperref`. At present, `confproc` does not provide any mechanism for this, so adding other such packages may result in surprises. A temporarily solution is to add them in the class definition itself...

We specify the input and font encodings, to allow for running LaTeX on a document with accents (in the list of authors and paper titles):

```
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
```

We change the default LaTeX font to 'Times', as it displays better in PDF files:

```
\usepackage{mathptmx}
```

N.B.: the `mathptmx` package better behaves than `times`: it doesn't change the maths font, and does not load Helvetica and Courier at horrible sizes (they look even worse than the default sans and mono fonts in combination)[12].

The header of DAFx-06 proceedings had a '9th', that requires:

```
\usepackage[super]{nth}

```

4.3 Commands and customization

Here is a list of what proceedings element that can be customized: the document layout (see sec. 4.2.12); the PDF metadata (see sec. 4.3.3); the titles for special section (see sec. 4.3.6); the front page (see sec. 4.3.5); the document header/footer (see sec. 4.3.4); the publishing information; the welcome letter(s); the title/author style in the table of contents and bookmarks (see sec. 5.1.13); the color for links (see sec. 4.3.2); and of course how many columns for the table of contents (1 or 2), bibliography (1 or 2) and index of authors (2 or 3) using options.

N.B.: All these customizations are also used in the provided example file `example3optim.tex`; the explanations in section 5 are then similar (for people skipping one of the two).

4.3.1 PDF vertical and horizontal shifts

We set the default *x* (left/right) and *y* (up/down) shifts used to insert PDFs files:

```
\setlength{\LaTeXxShift}{0pt}
\setlength{\LaTeXyShift}{-3mm} %letter
%\setlength{\LaTeXyShift}{1mm} %A4
\setlength{\WordxShift}{10pt}
\setlength{\WordyShift}{-40pt}

```

The commented line is used with the A4 format (see also sec. 4.2.12). The values may differ depending if the papers were generated using a LaTeX template and a Word template, in the case your templates are not perfectly identical (which is often the case). The default values provided by the class are those used for the DAFx-06 proceedings, and were tested for both letter and A4 format.

Those global values are used by default for each paper, except if `\procpaper` has other values specified for the option arguments: xshift=... and yshift=....

4.3.2 Define colors for internal and external hyper-links

When inserting the document class, we have used home-made colors for the links (citecolor, linkcolor and urlcolor); those colors remain to be defined in the document preamble before being used:

[12] Thanks to Will Robertson for this information.

```
\definecolor{colorforlink}{rgb}{0,0,0.8}
\definecolor{colorforcite}{rgb}{0,0.8,0}
\definecolor{colorforurl}{rgb}{0,0,1}

```

There are a few things you need to know about it:

- the way colors are declared is explained in the `color` package;
- the colorforlink color is used for all links in the table of contents and index of authors, as well as back-references;
- the colorforurl color is useful only if you include URL(s) in you preamble, or in the general bibliography (if any);
- the colorforcite color is useful only in two cases:
  - without a general bibliography: if you cite any document form the preamble (not from a paper);
  - with a general bibliography: it is only used during the merging process. After this process and when generating the final document, all citations will disappear, as the last page of the paper is properly inserted.

4.3.3 PDF metadata

As `confproc` is to be used with pdfLaTeX and generates a PDF, it makes sense to customize the PDF metadata, that correspond to the PDF file information that are provided from the operating system. At least three metadata should be set, given here with their default values. To do so, we start by defining generic strings that will be used:

`\DAFxname`

- firstly, the conference name

```
\newcommand{\DAFxname}{Proc.~of the \nth{9} %
  Int.~Conference on Digital Audio Effects (DAFx-06)}
```

`\DAFxdate`

- secondly, the conference date(s):

```
\newcommand{\DAFxdate}{September 18-20, 2006}
```

`\DAFxaddress`

- thirdly, the conference address:

```
\newcommand{\DAFxaddress}{Montreal, Canada}

```

We can then define the 3 PDF metadata as follows:

`\procpdfauthor`

- use `\procpdfauthor` to change the PDF author (default: '[Proceedings author/editor]'):

```
\renewcommand{\procpdfauthor}{Vincent Verfaille, McGill University}
```

`\procpdftitle`

- use `\procpdftitle` to change the PDF short title (default: '[Proceedings title]'):

```
\renewcommand{\procpdftitle}{DAFx-06 Proceedings - \DAFxaddress}
```

`\procpdfsubject`

- use `\procpdfsubject` to change the PDF subject (default: '[Proceedings description]'):

```
\renewcommand{\procpdfsubject}{Conference proceedings}

```

\hypersetup Those values are used to set the \hypersetup command. They are only evaluated when the document begins, and can also be replaced by redefining \hypersetup in the document preamble.

N.B.: depending on the PDF viewer, only the first 30 or so characters of those metadata may be displayed (eg. Apple's Preview, Adobe Reader). It is then advised to use the shortest possible texts.

\hypersetup N.B.: an alternative way to change the PDF metadata consist in using the \hypersetup command (see the hyperref package).

4.3.4 Header and footer

The paper templates often have a header and footer, so we may want to use the same headers/footers for the whole proceedings (using the headers option). This is costumized by redefining the left

\proclhead \procchead \proclhead and centre \procchead header commands:

```
126 \renewcommand{\procchead}{}%

127 %%\renewcommand{\proclhead}{{\em \small \procpdfsubject}}
128 \renewcommand{\proclhead}{{\em \small \DAFxname, \DAFxaddress, \DAFxdate}}
```

\proccfoot as well as \proccfoot for the central footers (the page number being centered by default):

```
129 \renewcommand{\proccfoot}{\small DAFX-\thepage}
```

\procfootvskip The \procfootvskip length adjusts the footer vertical position:

```
130 \setlength{\procfootvskip}{1.2mm}
```

\procoptfootvskip When checking the page numbering for papers which template already includes the page number, we may change the amount of vertical shift applied to move down by a few millimeters (default: 3mm) the footer when using the movepagenumber option. Then, use the \procoptfootvskip command:

```
131 %%\setlength{\procoptfootvskip}{4mm}
132
```

As soon as you remove the movepagenumber option, the footer comes back to its normal position.

4.3.5 Define front page and title commands

LaTeX commands To generate the front page with \maketitle, redefine usual LaTeX commands:

\author • the proceedings' author/editor:

```
133 %%\author{Vincent Verfaille, McGill University}
```

\title • the proceedings' title:

```
134 %%\title{Proc. of the \nth{9} Int. Conf. on Digital Audio Effects\\
135 %%  Montreal, Quebec, Canada}
```

\date • the proceedings' date:

```
136 %%\date{Sept.~18--20, 2006}
```

We can alternatively use the \procpdfauthor and \procpdftitle commands that we previously defined (if set to what is expected) to set:

\author • the proceedings' author/editor:

```
137 \author{\procpdfauthor}
```

\title • the proceedings' title:

```
138 %%\title{\procpdftitle}
139 \title{\DAFxname\\ \DAFxaddress}
```

\date

- the proceedings' date:

```
%%\date{\today}
\date{\DAFxdate}
```

Then, some fine tuning of all the parameters of this page is needed so that it looks as you wish (potentialy with logos, images, etc).

Import a PDF page from another editor As for the DAFx-06 proceedings, we may insert with `pdfpages` the cover, generated as another document (for instance with XeLaTeX):

```
%%\includepdf[noautoscale,pages=1,link]{\PICTPATH ex_1stpage.pdf}

```

Indeed, we found it easier to design our very own cover using XeTeX, or any other tool.
N.B.: the PDF file named `ex_1stpage.pdf` is provided in the `.zip` archive of the package but not generated by the package.

4.3.6 Special section titles (toc, index, biblio)

To change special section titles, redefine the LaTeX commands already modified by `confproc`:

\contentsname

- title of the table of contents (default: 'Conference Program'):

```
%%\renewcommand{\contentsname}{List of Sessions}
```

\bibname

- title of the general bibliography (default: 'Full Bibliography'):

```
%%\renewcommand{\bibname}{General Bibliography}
```

\indexname

- title of the index (default: 'Index of Authors'):

```
%%\renewcommand{\indexname}{List of Authors}
```

4.3.7 Declare paths to pictures, papers, texts...

\PAPERPATH

We then define paths to papers (with both PDF papers and related folders to batch re-generate them all) that the `example2custom.tex` file uses, *i.e.*:

```
\renewcommand{\PAPERPATH}{papers/}
```

4.3.8 Chapter and section styles

Chapter and section styles can also be modified (for instance with the `titlesec` package) to adapt them to your needs. Remember that days appear as parts in the TOC and bookmarks, sessions as chapters, and papers as sections.

4.3.9 Title/author layout

\texorpdfstring

The `\texorpdfstring` command allows for a different text in LaTeX and for the PDF (which is good for having different bookmark titles and table of contents entries). It is then used by default to add a line break between the paper title and the authors' names in the table of contents. You can

\papertitlestyle

customize the title font style using the `\papertitlestyle` command as in:

```
%%\renewcommand{\papertitlestyle}{\texorpdfstring{}{\scshape}}
```

\paperauthorstyle

that defines the paper's title in small capitals. The `\paperauthorstyle` command is used to customize the author font style. For instance, to replace the line break (between paper title and list of authors, in the table of contents) by a comma in the table of contents only (not in the PDF bookmark):

```
%%\renewcommand{\paperauthorstyle}{\texorpdfstring{, }{\break}}
```

In case we want to check each title and author list, that can be useful to specify how the overlayed text is formatted.

\confstylechecktitle To do so, we first define the format of the check title:

```
\renewcommand{\confstylechecktitle}{\vspace*{0.3cm} %
  \bf \sc \Large \noindent \centerline}
```

\confstylecheckauthor and then the format of the check author list:

```
\renewcommand{\confstylecheckauthor}{\large \it \noindent \centerline}
```

4.3.10 Make the index

The last step of the preamble is to make the index:

```
\makeindex
```

4.4 Front matter: cover page, index and table of contents

We can now start the document and its front matter by using:

```
%%%==========  PROCEEDINGS  ==========
\begin{document}

\frontmatter
```

New [v0.7] Since version 0.5, switching to the front matter does not anymore automatically changes the style of table of contents entries.

\frontmattertocstyle We then explicitely use the optional style:

```
\frontmattertocstyle
```

We then force the cover (ie. the first page of the proceedings) to be numbered '1':

```
\setcounter{page}{1}
```

4.4.1 Cover page

We now add a bookmark chapter in the front matter:

```
\pdfbookmark[0]{Preamble}{preamble}
```

That way, we ensure that all the sections in the front matter/preamble (cover page, welcome letters, etc) except the table of contents appear in a same bookmark as sub-items, thus reducing the number of lines appearing that do not deal with days, sessions, papers, etc. Note that we do it by hand. This is not as beautiful and general as if the class was doing it for you (which could have been done); however, not automatizing this bookmark entry allows the proceedings editor to decide if he wishes to link to the first pages or not.

We then include the first page and generate its bookmark entry:

```
\pdfbookmark[1]{Cover}{cover}
```

and produce the first page:

```
\maketitle
\newpage
```

4.4.2 Table of contents

To ensure next page is numbered and has proper headers/footers, use:

```
\otherpagestyle
```

We now insert the conference program (table of contents):

```
\tableofcontents
```

N.B.: the bookmark entries are automatically generated from the table of contents.

4.5 Main matter: the papers

We can then beginning inserting the papers.

```
%%%==== BEGINNING OF PAPERS ====
```

Before doing so however, we use the following line to set a counter that is used by the pdftk option to properly count page numbers taking into account the preamble.

```
\setcounter{npagespreamble}{\arabic{page}-1} % only useful for the 'pdftk' option
```

Why do we need to do so? Since page numbers change from roman to arabic, page numbers in the document do restart at 1, whereas for the PDF document (when opened into a viewer), page numbers are different and go on increasing! Therefore, the export of individual papers with `pdftk`[13] could not work without this trick, that I still did not figure out how to hide it from the user.

`\mainmatter` We now (and only after last line) switch to the main matter and to arabic page numbering:

```
\mainmatter
```

New [v0.7] Since version 0.5, switching to the main matter does not anymore automatically changes the style of table of contents entries.

`\mainmattertocstyle` We then explicitely use the optional style:

```
\mainmattertocstyle
```

We then insert the papers, day-by-day and session-by-session.

`\procday` Remember that from the bookmark structure point of view, a day appears as a part:

```
\procday{Day 1}
%    \color{black}
%  \begin{macro}{\session}
%    a session appears as a chapter:
%  \end{macro}
%    \color{black!50}
%    \begin{macrocode}
 \session{Oral Session 1}
```

`\procpaper` and a paper appears as a section:

```
 \procpaper[xshift=\LaTeXxShift{}, yshift=\LaTeXyShift{}, switch=45, npages=6,%
   title={Templates for One Author},%
   author={Alfred Alabama},%
   index={\index{Alabama, Alfred}},%
 ]{p_001}

 \procpaper[xshift=\LaTeXxShift{}, yshift=\LaTeXyShift{}, switch=21, npages=5,%
```

[13] Get `pdftk` at: `http://www.accesspdf.com/pdftk/`

```
    title={Templates for One Author with Two Affiliations},%
    author={Bob Boogie-Woogie},%
    index={\index{Boogie-Woogie, Bob}},%
  ]{p_003}

```

We also give an example of how the older interface to load paper can still be used:

```
%\procinsertpaper{\LaTeXxShift{} \LaTeXyShift}{5}{p_003}%
%    {Templates for One Author with Two Affiliations}% paper title
%    {Bob Boogie-Woogie}% list of authors
%    {\index{Boogie-Woogie, Bob}}% authors index entries
%    {Serra:1996:sms,Moorer:2000:AES:audio:millenium,%
%      Arfib:1998:DAFx,Haykin:1991:adaptive:filter}%
%    {21}{\pdfbookmark[2]{Bob Boogie-Woogie}{21.author1}}

\session{Poster Session 1}

  \procpaper[xshift=\LaTeXxShift{}, yshift=\LaTeXyShift{}, switch=33, npages=4,%

    title = {Templates for Two Authors},%
    author={Alfred Alabama, Chris Christmas},%
    index={\index{Alabama, Alfred}\index{Christmas, Chris}},%
  ]{p_005}

\procday{Day 2}
\session{Oral Session 2}

  \procpaper[xshift=\LaTeXxShift{}, yshift=\LaTeXyShift{}, switch=75, npages=6,%

    title={Templates for Three Authors},%
    author={Bob Boogie-Woogie, Chris Christmas, Don Didon},%
    index={\index{Boogie-Woogie, Bob}\index{Christmas, Chris}%
      \index{Didon, Don}},%
  ]{p_007}

  \procpaper[xshift=\LaTeXxShift{}, yshift=\LaTeXyShift{}, switch=27, npages=7,%

    title={Templates f\'or F\`o\"ur ÃĂuthors},%
    author={J\o{}hn J\"oe, K\'e\~{n}t K\^{\i}ng, L\`ou L\'ou,%
      M\`anfr\'ed J. M\^ost\u{e}k\i},
    index={\index{J\"oe, J\o{}hn}\index{K\^{\i}ng, K\'e\~{n}t}%
      \index{L\'ou, L\`ou}\index{M\^ost\u{e}k\i, M\`anfr\'ed J.}},
  ]{p_009}

%%%==== END OF PAPERS ====
```

4.6 Back matter: index of authors

`\backmatter` After all papers are inserted, we switch to the document back matter (bibliography & index):

```
\backmatter
```

Once again, we use the optional style for table of contents entries:

```
\backmattertocstyle
```

that we may have redefined it in the preamble (it uses the `titletoc` package).

`\insertindex` We finally insert the index:

```
\insertindex
\end{document}
```

⟨/example2custom⟩

4.7 LaTeX runs

To build this example, run the following LaTeX steps:

1. generates the first `.aux` and `.idx` files (use option papers=empty to go faster, as this time the number of pages for each paper has been defined):
   `pdflatex example2custom.tex`
2. generates the author index:
   `makeindex -s confproc2.ist example2custom.idx`
3. inserts table of contents and index, and update their page numbers for next run (use option papers=final to ensure internal links are correct):
   `pdflatex example2custom.tex`
4. final LaTeX run inserting table of contents and index with proper page numbers; useful only if the table of contents is longer than a single page (papers=final):
   `pdflatex example2custom.tex`

Next section provide another working example (`example3optim.tex`) with several tricks to help optimizing the proceedings building process, and that was tested by re-generating the DAFx-06 proceedings (several years after the conference). The resulting PDFs were identical (appart from the improvements for bookmarks managements), but the work line is much easier to use and read. To generate the example proceedings, run `confproc.ins` through LaTeX. Better, run the `bash` script called `buildproc` (see sec. 7.5): it will run all the steps for you.

5 Example 3 (`example3optim.tex` & others) Full working example

Incremental learning... Compared to the previous customization example in sec. 4, this third example illustrates a complete and working example with: the use of some more tools: various package option sets depending on the compilation step, the publishing informations, a welcome letter, a paper switch to simplify the paper insertion and program changes, and a general bibliography. Morevoer, the conference program is directly generated from a `.csv` file by the `generateswitch.pl` Perl script. Tools to ensure that all pages of each PDF are inserted without having to run LaTeX on the whole document (quite slow) are provided.

5.1 `example3optim.tex`: Main file

The main file is named `example3optim.tex`. Once again, lines of code that do no differ from previous example (ie. `example2custom.tex`) appear in grey color.

⟨∗example3optim⟩

5.1.1 Using the `confproc` class

The class is to be called as would have been the `book.cls`. For this example, the exhaustive list of option values is given (as well as a comment with possible values and default values between squared brackets):

```
\documentclass[letterpaper,% [letterpaper] | a4paper
  10pt,%                     [10pt] | 11pt | 12pt
  twoside,%                  [twoside] | oneside
  twosidepapers,%            [twosidepapers] | onesidepapers
  electronic,%               [electronic] | printed
  binding=0mm,%              [0mm]
  papers=final,%             empty | draft | [final]
  headers=exceptpdf,%        none | pdfonly | exceptpdf | [allpages]
  bib=backref,%              [none] | merge | backref | final
  paperselec=all,%           [all] | p_001 | paper_2 | 3 ...
  onecoltoc,%                [onecoltoc] | twocoltoc
  tocnum=left,%              [left] | right
  twocolbib,%                [twocolbib] | onecolbib
  threecolindex,%            [threecolindex] | twocolindex
  checktitle=false,%         true | [false]
  checkauthor=false,%        true | [false]
  showpapernumber=false,%    true | [false]
  movepagenumber=false,%     true | [false]
  showmarginlines=false,%    true | [false]
  colorheaders=black,%       [black] | red | any color!
  debug=false,%              true | [false]
  verbose=false,%            true | [false]
  pdftk=true,%               true | [false]
  hyperref={bookmarksdepth=1,bookmarksopen,bookmarksopenlevel=0,%
    linkcolor=colorforlink,urlcolor=colorforurl}, % [blue, blue, blue]
  geometry={text={175truemm,226truemm},% A4 & letter
    inner=0.805in,top=29.15mm,bottom=24.5mm,footskip=9.68mm,voffset=-5mm},%letter
  ]{confproc}
%%%    inner=0.69in,top=33.9mm,bottom=38mm,footskip=10.4mm,voffset=-4.8mm},%A4
```

To simplify the proceedings building, we can use the `bash` scripts in sec. 6.1 to insert a file containing this command with the list of options:

```
%%\input{exclass}
```

Inserting options that way allows to switch between class options for various LaTeX runs by using 2 files (`exclasspre.tex` and `exclasslast.tex`). In those two files, the class is defined with different options set, and each one is temporary renamed as `exclass.tex` before being inserted.

5.1.2 Loading extra packages

We define the other packages to be used.

Important note: any package that redefines LaTeX macros should be inserted before `hyperref`. At present, `confproc` does not provide any mechanism for this, so adding other such packages may result in surprises. A temporarily solution is to add them in the class definition itself...

We use `setspace` to change the line spacing in the welcome letter (which text is not as dense as the papers themselves):

```
\usepackage{setspace}
```

The `xkeyval` package ensures further compatibility with other packages using it:

```
\usepackage{xkeyval}
```

[BIB] We then use a bibliography style for the general bibliography:

```
\usepackage{newapave}
```

We then specify the input and font encodings, to allow for running LaTeX on a document with accents (in the list of authors and paper titles):

```
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
```

We change the default LaTeX font to 'Times', as it displays better in PDF files:

```
\usepackage{mathptmx}
```

The header of DAFx-06 proceedings had a '9th', that requires:

```
\usepackage[super]{nth}

```

5.1.3 Fine tuning the document layout

To refine the document layout, we may use the `layout` package:

```
%\usepackage{layout}
```

as well as `layouts` to refine the table of contents layout:

```
%\usepackage{layouts}
```

N.B.: When the table of contents layout is inserted too early, it is not properly displayed!

5.1.4 PDF vertical and horizontal shifts

We set the default *x* (left/right) and *y* (up/down) shifts used to insert PDFs files:

```
\setlength{\LaTeXxShift}{0pt}
\setlength{\LaTeXyShift}{-3mm} %letter
%\setlength{\LaTeXyShift}{1mm} %A4
\setlength{\WordxShift}{10pt}
\setlength{\WordyShift}{-40pt}

```

Those global values are used by default for each paper, except if `\procpaper` has other values specified for the option arguments: xshift=... and yshift=....

5.1.5 Define colors for internal and external hyper-links

When inserting the document class, we have used home-made colors for the links (citecolor, linkcolor and urlcolor); those colors remain to be defined in the document preamble before being used:

```
\definecolor{colorforlink}{rgb}{0,0,0.8}
\definecolor{colorforcite}{rgb}{0,0.8,0}
\definecolor{colorforurl}{rgb}{0,0,1}

```

See section 4.3.2 for complementary information.

5.1.6 Customize PDF metadata

At least three PDF metadata should be set, given here with their default values. To do so, we start by defining generic strings that will be used:

\DAFxname

- firstly, the conference name

```
\newcommand{\DAFxname}{Proc.~of the \nth{9} %
 Int.~Conference on Digital Audio Effects (DAFx-06)}
```

\DAFxdate

- secondly, the conference date(s):

```
\newcommand{\DAFxdate}{September 18-20, 2006}
```

\DAFxaddress

- thirdly, the conference address:

```
\newcommand{\DAFxaddress}{Montreal, Canada}

```

We can then define the 3 PDF metadata as follows:

\procpdfauthor

- use \procpdfauthor to change the PDF author (default: '[Proceedings author/editor]'):

```
    \renewcommand{\procpdfauthor}{Vincent Verfaille, McGill University}
```

\procpdftitle

- use \procpdftitle to change the PDF short title (default: '[Proceedings title]'):

```
    \renewcommand{\procpdftitle}{DAFx-06 Proceedings - \DAFxaddress}
```

\procpdfsubject

- use \procpdfsubject to change the PDF subject (default: '[Proceedings description]'):

```
    \renewcommand{\procpdfsubject}{Conference proceedings}

```

\hypersetup

Those values are used to ser the \hypersetup command. They are only evaluated when the document begins, and can also be replaced by redefining \hypersetup in the document preamble.

\hypersetup

N.B.: depending on the PDF viewer, only the first 30 or so characters of those metadata may be displayed (eg. Apple's Preview, Adobe Reader). It is then advised to use the shortest possible texts.

N.B.: an alternative way to change the PDF metadata consist in using the \hypersetup command (see the hyperref package).

5.1.7 Header and footer

The paper templates often have a header and footer, so we may want to use the same headers/footers for the whole proceedings (using the headers option). This is costumized by redefining the left \proclhead and centre \procchead header commands:

\proclhead
\procchead

```
\renewcommand{\procchead}{} %
\renewcommand{\proclhead}{{\em \small \DAFxname, \DAFxaddress, \DAFxdate}}
```

\proccfoot

as well as \proccfoot for the central footers (the page number being centered by default):

```
\renewcommand{\proccfoot}{{\small DAFX-\thepage}}
```

\procfootvskip

The \procfootvskip length adjusts the footer vertical position:

```
\setlength{\procfootvskip}{1.2mm}
```

\procoptfootvskip

When checking the page numbering for papers which template already includes the page number, we may change the amount of vertical shift applied to move down by a few millimeters (default: 3mm) the footer when using the movepagenumber option. Then, use the \procoptfootvskip command:

```
\setlength{\procoptfootvskip}{4mm}

```

As soon as you remove the movepagenumber option, the footer comes back to its normal position.

5.1.8 Define front page and title commands

To generate the front page with \maketitle, we redefine usual LaTeX commands:

\author

- the proceedings' author/editor:

```
\author{\procpdfauthor}
```

\title

- the proceedings' title:

```
\title{\DAFxname\\ \DAFxaddress}
```

\date

- the proceedings' date:

```
\date{\DAFxdate}

```

using the \procpdfauthor and \procpdftitle commands previously defined.

5.1.9 Special section titles (toc, index, biblio)

To change special section titles, redefine the LaTeX commands already modified by confproc:

\contentsname

- title of the table of contents (default: 'Conference Program'):

```
\renewcommand{\contentsname}{Day-by-Day Conference Program}
```

\bibname

- title of the general bibliography (default: 'Full Bibliography'):

```
\renewcommand{\bibname}{General Bibliography}
```

\indexname

- title of the index (default: 'Index of Authors'):

```
\renewcommand{\indexname}{List of Authors}
```

5.1.10 Declare paths to pictures, papers, texts...

We then define paths to different resources that the `example3optim.tex` file uses, *i.e.*:

\PAPERPATH
- papers (containing both PDF papers and related folders to re-generate them):

```
\renewcommand{\PAPERPATH}{papers/}
```

\PICPATH
- pictures (*e.g.* logos used in your first page and welcome letters):

```
\newcommand{\PICTPATH}{pictures/}
```

\TEXTPATH
- texts (*e.g.* publishing informations, welcome letters, the paper switch):

```
\newcommand{\TEXTPATH}{}
```

\BIBPATH
- bibliographies (*e.g.* 3 files as explained in sec. 5.7):

```
\newcommand{\BIBPATH}{}
```

5.1.11 Declare bibliographic files

\procbibfile We now define the file name of the main bibliography:

```
\newcommand{\procbibfile}{\BIBPATH exbiblio}
```

Several `.bib` files may be used to build the general bibliography (see sec. 5.7).

5.1.12 Chapter and section styles

Chapter and section styles can also be modified (for instance with the `titletoc` package) to adapt them to your needs. Remember that days appear as parts in the TOC and bookmarks, sessions as chapters, and papers as sections. Here is an example taken from the ICMC 2009 proceedings, where a TOC with left-aligned numbers is added with horizontal lines of different thickness before days and sessions. To do so, we first define our horizontal rulers:

```
\newcommand{\myaddhruletotoc}{\vspace*{0.1cm}%
  \noindent\protect\hrulefill\par\vspace*{-0.15cm}}
\newcommand{\myaddthickhruletotoc}{\vspace*{0.5cm}%
  \noindent\protect\hrule height 0.6ex \hfill\par\vspace*{0.1cm}}
```

\procday We then redefine the `\procday` command to add an empty line in the TOC before each day:

```
\renewcommand{\procday}[1]{%
  \phantomsection%
  \addcontentsline{toc}{part}{#1}} % \centerline{#1}
```

\session as well as the `\session` command to add an empty line in the TOC before each session:

```
\renewcommand{\session}[1]{%
  \phantomsection%
  \addcontentsline{toc}{chapter}{#1}}
```

\mainmattertocstyle We can finally use the `titletoc` package and syntax to modify the TOC layout by adding the horizontal rulers. Note that given the way it is redefined, it will work properly for both left-aligned and right-aligned TOC numbers. Here is the definition for the main matter (`\mainmattertocstyle`):

```
%% idem for left/right numbering
\renewcommand{\mainmattertocstyle}{
  \titlecontents{part}[0pt]%
    {\addvspace{3mm}}%
    {\myaddthickhruletotoc\Large\bfseries}%
    {\myaddthickhruletotoc\Large\bfseries}%
    {}%
```

```
    [\addvspace{0.5mm}]%
  \titlecontents{chapter}[0pt]%
    {\addvspace{2mm}}%
    {\myaddhruletotoc\large\bfseries\itshape}%
    {\myaddhruletotoc\large\bfseries\itshape}%
    {}%
    [\addvspace{0.5mm}]%
}
```

5.1.13 Title/author layout

\texorpdfstring

The \texorpdfstring command allows for a different text in LaTeX and for the PDF (which is good for having different bookmark titles and table of contents entries). It is then used by default to add a line break between the paper title and the authors' names in the table of contents. We

\papertitlestyle

customize the title font style using the \papertitlestyle command as in:

```
%%\renewcommand{\papertitlestyle}{}
\renewcommand{\papertitlestyle}{\texorpdfstring{}{\scshape}}
```

\paperauthorstyle

that defines the paper's title in small capitals (default with the comment). The \paperauthorstyle command is used to customize the author font style. For instance, to replace the line break (between paper title and list of authors, in the table of contents) by a comma in the table of contents only (and not in the PDF bookmark) and have names right-flushed[14]:

```
%%\renewcommand{\paperauthorstyle}{\texorpdfstring{\newline\itshape}{\break}}
\renewcommand{\paperauthorstyle}{\texorpdfstring{, \hfill}{\break}}
```

Again, the default appears with the comment sign.

\proctoctitleauthor

Consider now that we prefer to invert the paper title and the author list positions. We first redefine \proctoctitleauthor, that defines how the paper title and author list are ordered and formatted in the TOC:

```
\renewcommand{\proctoctitleauthor}[2]{%
  \texorpdfstring{{\paperauthorstyle #2}{\papertitlestyle #1}}%
  {{\papertitlestyle #1}}}
```

We then redefine the formatting of the author list (italic) and paper title (boldface), and also which of the two commands is responsible for the line break (before the paper title):

```
\renewcommand{\paperauthorstyle}{\texorpdfstring{\itshape}{}}
\renewcommand{\papertitlestyle}{\texorpdfstring{\newline}{\break}}
```

\confstylechecktitle

In the case we want to check each title and author list, that can be useful to specify how the overlayed text is formatted. To do so, we define the format for the check title:

```
\renewcommand{\confstylechecktitle}{\vspace*{0.3cm} %
  \bf \sc \Large \noindent \centerline}
```

\confstylecheckauthor

and for the check author list:

```
\renewcommand{\confstylecheckauthor}{\large \it \noindent \centerline}
```

[14] Such option usually works well for short titles and author list:
My short title, FirstName1 LastName1, FirstName2 LastName2
but longer titles/author list may unfortunately give a different result, for instance:
My very long title that I like and do not want to shorten, FirstName1 LastName1, FirstName2 LastName2, FirstName3 LastName3, FirstName4 LastName4

5.1.14 Make the index

The last step of the preamble is to build the index:

```
\makeindex
```

5.1.15 Start the document: front matter (cover page and table of contents)

We can now start the document and its front matter by using:

```
%%%==========  PROCEEDINGS  ==========
\begin{document}

\frontmatter
```

5.1.16 Display the document layout

To check the document layout (thanks to the `layout` package), uncomment:

```
%%\layout
```

To specifically check the table of contents layout (thanks to the `layouts` package), uncomment:

```
%%\begin{figure}
%%    \setlayoutscale{0.8} \tocdiagram
%%    \caption{Table of Contents entry parameters} \label{fig:tocp}
%%\end{figure}
%%\begin{figure}
%%    \setlayoutscale{0.8} \currenttoc \tocdesign
%%    \caption{Typical Table of Contents entry for this document}
%%    \label{fig:thistoc}
%%\end{figure}
```

Then can either be inserted at the end of the document (not changing page numbering, but may be forgetten as we do not so often check the last page) or at its beginning (changing page numbering but being the first page you see when opening it).

`\clearsingleordoublepage` To go to the next page (right-opening page in two-side mode), use:

```
%%\clearsingleordoublepage
```

We then force the cover (ie. the first page of the proceedings) to be numbered '1':

```
\setcounter{page}{1}
```

5.1.17 Cover page

We now add a bookmark chapter in the front matter:

```
\pdfbookmark[0]{Preamble}{preamble}
```

That way, we ensure that all the sections in the front matter/preamble (cover page, welcome letters, etc) except the table of contents appear in a same bookmark as sub-items, thus reducing the number of lines appearing that do not deal with days, sessions, papers, etc. Note that we do it by hand. This is not as beautiful and general as if the class was doing it for you (which could have been done); however, not automatizing this bookmark entry allows the proceedings editor to decide if he wishes to link to the first pages or not.

We then include the first page and generate its bookmark entry:

```
\pdfbookmark[1]{Cover}{cover}
```

and produce the first page:

```
\maketitle
```

An alternative to the \maketitle command consists in inserting the first page as a PDF 1-page document generated with another software:

```
%%\includepdf[noautoscale,pages=1,link]{\PICTPATH ex_1stpage.pdf}
```

5.1.18 Customize TOC formatting

We can now change some of the TOC formatting (centered, lower and small capitals TOC name) with the following hack:

(i) add a 1cm vertical space:

```
\addtocontents{toc}{\vskip 1cm}
```

(ii) add the TOC name in the text of the TOC contents, with the formatting we want:

```
\addtocontents{toc}{\centerline{\huge\textsc{Conference Program}}}
```

(iii) remove the TOC name in the TOC only (not in the PDF bookmarks):

```
\renewcommand{\contentsname}{\texorpdfstring{}{Conference Program}}
```

5.1.19 Publishing informations

Publishing informations are then given on page 2, inside the cover:

```
\newpage
\vspace*{1.7cm}
\pdfbookmark[1]{Publishing informations}{publishing}
```

with no header nor footer on this page:

```
\thispagestyle{empty}
```

We then provide the publishing information itself:

```
\noindent {\bf Published by:}\\ Laboratory Name\\ Department name\\
School Name\\ University Name\\
\url{http://www.conferencesite.com}\\
```

We also indicate the ISBN number:

```
\vspace*{0.15cm}\newline
\noindent {\bf ISBN: X-XXXX-XXXXXX}\\
```

and the credits:

```
\vspace*{0.35cm}\newline
\noindent {\bf Credits:}\\
Cover design: Firstname Lastname\\
Logo photo: Firstname Lastname\\
\LaTeX{} editor: Firstname Lastname\\
```

If you think confproc is a time-saving solution, that's a good place to spread the word[15].

```
using \LaTeX's 'confproc' package, version 0.8 (optional: by V. Verfaille)\\
```

We then indicate where and when the proceedings were printed:

```
\vspace*{0.35cm}\newline
\noindent  Printed in City by Print-Company --- Month Year
```

[15] Do not forget to send me a postcard too :-).

5.1.20 Welcome letters

To ensure next page is numbered and has proper headers/footers, use:

```
\otherpagestyle
```

Roman page numbers now start to appear. We include all welcome letters[16]:

```
%%%-- Welcome letters
\clearsingleordoublepage
\vspace*{0.6cm}
\thisotherpagestyle
```

We create the bookmark entry by hand (so that you can remove it):

```
\pdfbookmark[1]{Welcome from Firstname Lastname}{welcome}
```

and the corresponding section (and table of contents entry):

```
\section*{Welcome from Firstname Lastname, Conference Chair}
```

Depending on the text length, you may use either 1.5 line spacing:

```
\vspace*{1.1cm}
\onehalfspace
\begin{center}
  \begin{minipage}[h]{14cm}
    Text of the welcome letter, with 1.5 lines spacing, blah blah...
    Text of the welcome letter, with 1.5 lines spacing, blah blah...
    Text of the welcome letter, with 1.5 lines spacing, blah blah...
    Text of the welcome letter, with 1.5 lines spacing, blah blah...
  \end{minipage}
\end{center}
```

or double line spacing (both are using the `setspace` style):

```
\doublespace
\begin{center}
  \begin{minipage}[h]{14cm}
    Text of the welcome letter, with 2 lines spacing, blah blah...
    Text of the welcome letter, with 2 lines spacing, blah blah...
    Text of the welcome letter, with 2 lines spacing, blah blah...
    Text of the welcome letter, with 2 lines spacing, blah blah...
  \end{minipage}
\end{center}
\singlespace
```

5.1.21 Table of contents

We now insert the conference program (table of contents):

```
\tableofcontents

```

N.B.: the bookmark entries are automatically generated from the table of contents.

5.1.22 Main matter: the papers

We can then beginning inserting the papers.

```
%%%==== BEGINNING OF PAPERS ====
```

Before doing so however, we use the following line to set a counter that is used by the pdftk option to properly count page numbers taking into account the preamble.

```
\setcounter{npagespreamble}{\arabic{page}-1} % only useful for the 'pdftk' option
```

[16] There is only one welcome letter in this example, but there usually are many others: from the faculty dean, the department dean, the conference chair, etc.

Why do we need to do so? Since page numbers change from roman to arabic, page numbers in the document do restart at 1, whereas for the PDF document (when opened into a viewer), page numbers are different and go on increasing! Therefore, the export of individual papers with `pdftk`[17] could not work without this trick, that I still did not figure out how to hide from the user.

`\mainmatter` We now (and only after last line) switch to the main matter and to arabic page numbering:

```
416 \mainmatter
```

New [v0.7] Since version 0.5, switching to the main matter does not anymore automatically changes the style of table of contents entries.

`\mainmattertocstyle` We then explicitely use the optional style:

```
417 \mainmattertocstyle
```

and can even redefine it in the preamble (it uses the `titletoc` package). Then, we include the file with all papers' information organized with a switch:

```
418 \input{\TEXTPATH expapersswitch}
```

We then insert the papers, day-by-day and session-by-session.

`\procday` Remember that from the bookmark structure point of view, a day appears as a part:

```
419 \procday{Day 1}
```

`\session` a session appears as a chapter:

```
420   \session{Oral Session 1}
```

`\paperid` and a paper appears as a section. Each paper is inserted by the `\paperid{ID}{PDFname}` command, that uses the switch in `expapersswitch.tex` to get the information about paper ID:

```
421     \paperid{45}{p_001}
422     \paperid{21}{p_003}
```

We also insert a poster session with one paper:

```
423   \session{Poster Session 1}

424     \paperid{33}{p_005}
```

and a second oral presentation session with two more papers:

```
425 \procday{Day 2}
426   \session{Oral Session 2}

427     \paperid{75}{p_007}
428     \paperid{27}{p_009}

429
430 %%%==== END OF PAPERS ====
```

5.1.23 Back matter

`\backmatter` After all papers are inserted, we switch to the document back matter (bibliography & index):

```
431 \backmatter
```

Once again, we use the optional style for table of contents entries:

```
432 \backmattertocstyle
```

that we may have redefine it in the preamble (it uses the `titletoc` package).

[17] Get `pdftk` at: `http://www.accesspdf.com/pdftk/`

5.1.24 General bibliography

The general bibliography is inserted with the following style:

```
\bibliographystyle{newapave}
```

This DAFx-06 style is a modification of the `newapa` style: the year is indicated at the end, before the back-references, instead of being between parenthesis right after the list of authors. If you prefer the `newapa` style (or any other style), simply replace this last line by:

```
%%\bibliographystyle{newapa}
```

and comment the line that inserts the `newapave` style.

The bibliography is then inserted:

```
{\footnotesize\bibliography{\procbibfile}}
```

Note that the general bibliography may be very long. Changing the font size (for instance to `\footnotesize` as in the previous line) may then be a good idea.

5.1.25 Index of authors

`\insertindex` We finally insert the index:

```
\insertindex
\end{document}
```

⟨/example3optim⟩

5.2 `expapersswitch.tex`: Paper switch!

Let us now take a look at the paper switch, which is central to this version of the proceedings. It contains a switch to proceedings papers, allowing to work on the document without needing to know yet the final order of papers (which is useful when working in parallel on the document and on the conference program).

New [v0.7] `\procpaper` Version 0.7 of `confproc` redefines the paper insertion interface with `\procpaper`, that uses key-values options and thus clarifies the LaTeX code. We define the `\paperid` command:

`\paperid`

⟨*expapersswitch⟩

```
\newcommand{\paperid}[2]{
```

`\paperswitch` Inside the switch, the `\paperswitch` command is set to the paper reference:

```
\renewcommand{\paperswitch}{#1}
```

Papers can still be inserted using the pre-version 0.5 `\procinsertpaper` command (see next section), but it is strongly recommanded to forget it and use the new interface of the `\procpaper` command (version 0.5 and above).

5.2.1 First and old way: pre-version 0.5 interface

To insert the first paper (ID=45) with the pre-version 0.5 command (compatibility check), we first define commands to help making this old code readible:

```
%========== PAPER ID = 45 ==========
\ifnum\paperswitch=45 {
  \newcommand{\papertitle}{Templates for One Author}
  \newcommand{\paperauthors}{Alfred Alabama}
  \newcommand{\paperindex}{\index{Alabama, Alfred}}
  \newcommand{\paperref}{\paperswitch}
```

```
\newcommand{\paperpagenum}{6}
\newcommand{\papercite}{Serra:1996:sms,%
  Moorer:2000:AES:audio:millenium,Arfib:1998:DAFx,%
  Mitra:Kaiser:1993:DSP:handbook}
```

\procinsertpaper

We then use the old \procinsertpaper command to insert papers[18]:

```
\procinsertpaper{\LaTeXxShift{} \LaTeXyShift}{\paperpagenum}%
  {\paperref}{\papertitle}{\paperauthors}{\paperindex}{\papercite}%
  {#2}{\pdfbookmark[2]{Alfred Alabama}{#2.author1}}}
\fi
```

As we can see, it still works; this is however too ugly. Why would you still want to use such an bad interface to set arguments?

5.2.2 Second and preferred way: shorter and more readible

\procpaper
New [v0.7]

Thanks to the new \procpaper command, it is much simpler and more readible to insert papers: except the paper file name, the other arguments are optional, and their name is clear enough! The lines of next code correspond to what is generated by the generateswitch.pl Perl script (see sec. 7.4), which converts the .csv data into LaTeX code to insert in this current file:

```
\ifnum\paperswitch=21
  \procpaper[xshift=\LaTeXxShift{}, yshift=\LaTeXyShift{}, npages=5, switch=21,%
    title={Templates for One Author with Two Affiliations},%
    author={Bob Boogie-Woogie},%
    index={\index{Boogie-Woogie, Bob}},%
    cite={Serra:1996:sms,Moorer:2000:AES:audio:millenium,Arfib:1998:DAFx,%
      Haykin:1991:adaptive:filter},%
    bookmark={\pdfbookmark[2]{Bob Boogie-Woogie}{#2.author1}}%
  ]{#2}
\fi

\ifnum\paperswitch=27
  \procpaper[xshift=\LaTeXxShift{}, yshift=\LaTeXyShift{}, npages=7, switch=27,%
    title={Templates f\'or F\‘o\"ur ÃĂuthors},%
    author={J\o{}hn J\"oe, K\’e\~{n}t K\^{\i}ng, L\‘ou L\’ou,%
      M\‘anfr\’ed J. M\^ost\u{e}k\i},%
    index={\index{J\"oe, J\o{}hn}\index{K\^{\i}ng, K\’e\~{n}t}%
      \index{L\’ou, L\‘ou}\index{M\^ost\u{e}k\i, M\‘anfr\’ed J.}},%
    cite={Serra:1996:sms,Moorer:2000:AES:audio:millenium,Dutilleux:1991,%
      Fitz:Haken:2003:Web:morphing:loris},
    bookmark={\pdfbookmark[2]{J\o{}hn J\"oe}{#2.author1}%
      \pdfbookmark[2]{K\’e\~{n}t K\^{\i}ng}{#2.author2}%
      \pdfbookmark[2]{L\‘ou L\’ou}{#2.author3}%
      \pdfbookmark[2]{M\‘anfr\’ed J. M\^ost\u{e}k\i}{#2.author4}}%
  ]{#2}
\fi

\ifnum\paperswitch=33
  \procpaper[xshift=\LaTeXxShift{}, yshift=\LaTeXyShift{}, npages=4, switch=33,%
    title = {Templates for Two Authors},%
    author={Alfred Alabama, Chris Christmas},%
    index={\index{Alabama, Alfred}\index{Christmas, Chris}},%
    cite={Serra:1996:sms,Moorer:2000:AES:audio:millenium,%
```

[18] This command has 9 arguments that must be used in this exact order: (i) X and Y shifts (with a space in between, as in ‘10 12’); (ii) the number of pages; (iii) the paper reference; (iv) the title; (v) the list of authors; (vi) the index entries; (vii) the citations for the general bibliography; (viii) the name of the PDF file to insert; (ix) the bookmark entries for the authors. Hopefully, Andreas Matthias suggested me to use key-values options, and it is no more useful to do it this old way!

```
      Arfib:1998:DAFx,Askenfelt:1976:automatic:transcription},
    bookmark={\pdfbookmark[2]{Alfred Alabama}{#2.author1}%
      \pdfbookmark[2]{Chris Christmas}{#2.author2}}%
  ]{#2}
\fi

\ifnum\paperswitch=75
  \procpaper[xshift=\LaTeXxShift{}, yshift=\LaTeXyShift{}, npages=6, switch=75,%
    title={Templates for Three Authors},%
    author={Bob Boogie-Woogie, Chris Christmas, Don Didon},%
    index={\index{Boogie-Woogie, Bob}\index{Christmas, Chris}%
      \index{Didon, Don}},%
    cite={Serra:1996:sms,Moorer:2000:AES:audio:millenium,%
      Arfib:1998:DAFx,Egozy:1995:MIT:features:gesture},%
    bookmark={\pdfbookmark[2]{Bob Boogie-Woogie}{#2.author1}%
      \pdfbookmark[2]{Chris Christmas}{#2.author2}%
      \pdfbookmark[2]{Don Didon}{#2.author3}}%
  ]{#2}
\fi
```

We're done with the papers data, but **we must not forget to close the curly brace!**

```
}
```
⟨/expapersswitch⟩

5.3 `expages.tex`: Get page numbers and recompile all papers

In the case where inserted papers have headers/footers, we may have to recompile them all with the proper page numbers. Before doing so, compile the proceedings enough times so that the table of contents is generated and inserted. Then, use the page number indicated for each paper to edit accordingly the `expages.tex` file. An example is provided here:

⟨*expages⟩
```
\newcommand{\setpagenumber}[1]{
  \newcommand{\paperswitch}{#1}
    \ifnum\paperswitch=45 {\setcounter{page}{1}}\fi
    \ifnum\paperswitch=21 {\setcounter{page}{7}}\fi
    \ifnum\paperswitch=27 {\setcounter{page}{13}}\fi
    \ifnum\paperswitch=33 {\setcounter{page}{17}}\fi
    \ifnum\paperswitch=75 {\setcounter{page}{23}}\fi
}
```
⟨/expages⟩

You may then recompile all papers (use the `buildpapers` Unix script, see sec. 7.12), provided that they all have the corresponding line in their preamble:
`\input{../../expages.tex}\setpagenumber{01}`
where 01 is the paper reference (to be changed for each paper). Using the following:
`\setcounter{page}{1}`
would of course have the equivalent effect, except that you would have to re-edit each paper after changing your program order.

5.4 `exsessions.tex`: Organize the conference program by sessions/day

Depending on the conference duration, the program may feature a few sessions during 2 or 3 days, or many sessions during 4 to 7 days (or even more). Then, the table of contents and the bookmarks may be organized:

- by sessions and then by related papers (short conferences): see sec. 5.4.1;

- by day, then by sessions and then by papers (long conferences, to avoid a too long list of sessions in the PDF bookmark): see sec. 5.4.2.

The mechanism used in `confproc` to build the table fo contents and bookmarks is based on section levels: days are inserted in the table of contents and bookmarks as parts, whereas sessions are inserted as chapters and papers as sections.

Note that the `confproc` **does not handle programs with parallel sessions**. It is then up to you to decide in which order they may appear in the table of contents.

5.4.1 `exsessions.tex`: Program organized by sessions

For a small size conference, if not using days (comment the `\procday` lines in the example), you will obtain the table of contents corresponding to Tab 3. The corresponding bookmark is depicted closed in Tab. 4, opened at its first level in Tab. 5, and opened at its second level in Tab. 6.

Conference Program

Oral Session 1

1 Templates for One Author
Alfred Alabama

7 Templates for One Author with Two Affiliations
Bob Boogie-Woogie

Poster Session 1

11 Templates for Two Authors
Alfred Alabama, Chris Christmas

Oral Session 2

15 Templates for Three Authors
Bob Boogie-Woogie, Chris Christmas, Don Didon

21 Templates fór Fòür Àuthors
John Jöe, Kéñt Kîng, Lòu Lóu, Mànfréd J. Môstěki

27 **Full Bibliography**

28 **Index of Authors**

Table 3: *Example of table of contents for a conference organized by sessions.*

▶ Preamble
Program
▶ Oral Session 1
▶ Poster Session 1
▶ Oral Session 2
Full Bibliography
Index of Authors

Table 4: *Closed bookmarks for a conference organized by sessions.*

- ▼ Preamble
  - Cover
  - Publishing informations
  - Welcome from Firstname Lastname
- Program
- ▼ Oral Session 1
  - ▶ Template for One Author
  - ▶ Template for One Author with Two Affiliations
- ▼ Poster Session 1
  - ▶ Template for Two Authors
- ▼ Oral Session 2
  - ▶ Template for Three Authors
  - ▶ Template fór Fòür Àuthors
- Full Bibliography
- Index of Authors

Table 5: *First-level opened bookmarks for a conference organized by sessions.*

- ▼ Preamble
  - Cover
  - Publishing informations
  - Welcome from Firstname Lastname
- Program
- ▼ Oral Session 1
  - ▼ Template for One Author
    - Alfred Alabama
  - ▼ Template for One Author with Two Affiliations
    - Bob Boogie-Woogie
- ▼ Poster Session 1
  - ▼ Template for Two Authors
    - Alfred Alabama
    - Chris Christmas
- ▼ Oral Session 2
  - ▼ Template for Three Authors
    - Bob Boogie-Woogie
    - Chris Christmas
    - Don Didon
  - ▼ Template fór Fòür Àuthors
    - John Jöe
    - Kéñt Kîng
    - Lòu Lóu
    - Mànfred J. Môstěkı
- Full Bibliography
- Index of Authors

Table 6: *Second-level opened bookmarks for a conference organized by sessions.*

5.4.2 Program organized by days

In the case of bigger conferences with a program organized by day, you will get the table of contents corresponding to Tab 7. The corresponding bookmark is depicted closed in Tab. 8, opened at its first level in Tab. 9, and opened at its second level Tab. 10.

Table 7: *Example of table of contents for a conference organized by day.*

Table 8: *Closed bookmarks for a conference organized by days.*

- ▼ Preamble
  - Cover
  - Publishing informations
  - Welcome from Firstname Lastname
- Program
- ▼ Day 1
  - ▶ Oral Session 1
  - ▶ Poster Session 1
- ▼ Day 2
  - ▶ Oral Session 2
- Full Bibliography
- Index of Authors

Table 9: *First-level opened bookmarks for a conference organized by days.*

- ▼ Preamble
  - Cover
  - Publishing informations
  - Welcome from Firstname Lastname
- Program
- ▼ Day 1
  - ▼ Oral Session 1
    - ▶ Template for One Author
    - ▶ Template for One Author with Two Affiliations
  - ▼ Poster Session 1
    - ▶ Template for Two Authors
- ▼ Day 2
  - ▼ Oral Session 2
    - ▶ Template for Three Authors
    - ▶ Template fór Fòür Àuthors
- Full Bibliography
- Index of Authors

Table 10: *Second-level opened bookmarks for a conference organized by days.*

5.5 `exprogram.csv`: Generate the conference program from a CSV file

It may be easier to collect data about the papers from a server, manipulate them in a spreadsheet software (for example M$ Excel), and then generate the program from a `.csv` file. The provided `generateswitch.pl` Perl script (see sec. 7.4) generates the corresponding `expaperswitch.tex` and `exsessions.tex` files for the example. First, let's take a look at the following CSV file, that contains the example's conference program[19]:

520 ⟨∗exprogram⟩
```
521 Type,Paper Number,PC Decision,Pages,Title,File Name,Generated,Citations,Auth1 First Name,Auth1 L
522 Type,-2,0,,,,,,First Name,Last Name,First Name,Last Name,First Name,Last Name, F.Name, L.Name,,
523 Day,0,,,"Day 1: September 18, 2007",,,,,,,,,,,,
524 Session,0,,,Oral Session 1,,,,,,,,,,,,
525 paper,45,0,6,Templates for One Author,p_001,LaTeX,"Serra:1996:sms,Moorer:2000:AES:audio:milleniu
526 paper,21,0,5,Templates for One Author with Two Affiliations,p_003,LaTeX,"Serra:1996:sms,Moorer:2
527 Poster Session,0,,,Poster Session 1,,,,,,,,,,,,
528 paper,33,P,4,Templates for Two Authors,p_005,LaTeX,"Serra:1996:sms,Moorer:2000:AES:audio:millen
529 Day,0,,,"Day 2: September 19, 2007",,,,,,,,,,,,
530 Session,0,,,Oral Session 2,,,,,,,,,,,,
531 paper,75,0,6,Templates for Three Authors,p_007,LaTeX,"Serra:1996:sms,Moorer:2000:AES:audio:mille
532 paper,27,0,7,Templates fÃşr FÃšÃijr ÃĂuthors,p_009,LaTeX,"Serra:1996:sms,Moorer:2000:AES:audio:m
533 ⟨/exprogram⟩
```

As expected fromt its first line, it contains the following columns:

1. **Type:** the script will accept the following values:
   - use `Type` for the items/lines to be ignored;
   - Day: use `Day`;
   - Session: for oral session, use `Session`, `Paper Session` or `Oral Session`; for poster sessions, use `poster session`; for demonstration sessions, use `Demo Session`;
   - Paper: use `paper` or `oral` for oral presentation; `poster` for poster presentation; `demo` for demo. The 3 output identical code anyway: the different values only help to organize the program!.

   N.B.: theses values are not case sensitively processed by the `generateswitch.pl` Perl script.
2. **Number:** paper number or reference, often generated by the submission system. It will be used for paper insertion, for ordering the program, etc.
3. **PC Decision:** `oral` or `poster`. It does not change the LaTeX generated code, so you may not use it;
4. **Pages:** number of pages;
5. **Title:** paper/session title;
6. **File Name:** name of the corresponding `.pdf` file;
7. **Generated:** software used to generate the `.pdf` files (e.g.: `LaTeX`, `Word`): this allows to use different *X* and *Y* offset values (we however used the same value for all papers of one kind);
8. **Citations:** list of bibliography items for the general bibliography (ex: `\cite{bibitem1,bibitem2,bibitem3}`); blank if no general bibliography;
9. **Auth1 First Name:** first name of author 1;

[19] This is normal that this text goes on after the margin. Please check the generated file if you wish to read each line.

10. **Auth1 Last Name:** last name of author 1;
11. **Auth2 First Name:** first name of author 2, blank if none;
12. **Auth2 Last Name:** last name of author 2, blank if none;
13. **Auth3 First Name:** first name of author 3, blank if none;
14. **Auth3 Last Name:** last name of author 3, blank if none;
15. **Auth4 First Name:** first name of author 4, blank if none;
16. **Auth4 Last Name:** last name of author 4, blank if none;
17. **comments:** there is an extra column, that is not used by the script.

5.6 `exbiblio.bib`: Common bibliography items

Let us now take a look at the common bibliographic items of this example:

534 ⟨∗exbiblio⟩
```
535 %-- references to a book
536 @book{Mitra:Kaiser:1993:DSP:handbook,
537   Author = {S.~K. Mitra and J.~F. Kaiser},
538   Title = {Handbook for Digital Signal Processing},
539   Publisher = {J. Wiley {\&} Sons},
540   Year = {1993}}
541
542 @book{Haykin:1991:adaptive:filter,
543   Author = {Simon Haykin},
544   Title = {Adaptive Filter Theory},
545   Publisher = {Prentice Hall},
546   Address = {Englewood Cliffs},
547   Edition = {Second},
548   Year = {1991}}
549
550 %-- reference to a book chapter
551 @inbook{Serra:1996:sms,
552   Author = {X. Serra},
553   Chapter = {Musical Sound Modeling with Sinusoids plus Noise},
554   Publisher = {G. D. Poli, A. Picialli, S. T. Pope and C. Roads,%
555     Eds.~Swets~\&~Zeitlinger},
556   Title = {Musical Signal Processing},
557   Pages = {91--122},
558   Year = {1996}}
559
560 %-- reference to a journal paper
561 @article{Moorer:2000:AES:audio:millenium,
562   Author = {James A. Moorer},
563   Title = {Audio in the New Millennium},
564   Journal = {Journal of the {AES}},
565   Volume = 48,
566   Number = 5,
567   Year = 2000,
568   Month = may,
569   Pages = {490--498}}
570
571 %-- reference to a proceeding paper
572 @inproceedings{Arfib:1998:DAFx,
```

```
573   Author = {D. Arfib},
574   Booktitle = {Proc. of the COST-G6 Workshop on Digital Audio Effects %
575     (DAFx-98)},
576   Title = {Different Ways to Write Digital Audio Effects Programs},
577   Address = {Barcelona, Spain},
578   Pages = {188--91},
579   Year = {1998}}
580
581 %-- reference to a technical report
582 @techreport{Askenfelt:1976:automatic:transcription,
583   Author = {A. Askenfelt},
584   Title = {Automatic notation of played music (status report)},
585   Institution = {{STL-QPSR, Vol. 1, pp. 1--11}},
586   Year = {1976}}
587
588 %-- reference to a master thesis
589 @mastersthesis{Egozy:1995:MIT:features:gesture,
590   Author = {E.~B. Egozy},
591   title = {Deriving musical control features from a real-time timbre %
592     analysis of the clarinet},
593   School = {Massachusetts Institute of Technology},
594   Year = {1995}}
595
596 %-- reference to a PhD thesis
597 @phdthesis{Dutilleux:1991,
598   Author = {P. Dutilleux},
599   School = {University of Aix-Marseille II},
600   Title = {Vers la machine \‘a sculpter le son, modification en %
601     temps-r\’eel des caract\’eristiques fr\’equentielles et temporelles%
602     des sons},
603   Year = {1991}}
604
605 %-- reference to a web page
606 @unpublished{Fitz:Haken:2003:Web:morphing:loris,
607   Author = {K. Fitz and L. Haken},
608   Title = {{Current Research in Real-time Sound Morphing}},
609   Note = {Available at \href{http://www.cerlsoundgroup.org/RealTimeMorph/}%
610     {http://www.cerlsoundgroup.org/RealTimeMorph/}},
611   Year = {Accessed March 08, 2006}}
612 ⟨/exbiblio⟩
```

Please refer to sec. 5.7 for details about the bibliography merging process.

5.7 General bibliography

5.7.1 Making the general bibliography

For the DAFx-06 proceedings (but not for the provided example), we worked with three bibliography files in order to simplify the bibliography merging process:

- `exbibconcat.bib` containing all citations for all papers;
- `exbibcommon.bib` containing common bibliography items, added one by one during the merging process;
- `exbibstrings.bib` containing all common strings (conference names, journal names, etc), to ensure coherence among citations from same sources (journal, conference).

Here is how those files were created and used:

1. create the complete bibliography:
   (a) for each paper, change its bib item tags to a tag that cannot be common to 2 papers (we used a format starting with the paper number: `paperID:originaltag`)[20];
   (b) ensure that each paper has a proper list of bibliography items using those new tags;
   (c) add individual paper bibliography items into the file named `exbibconcat.bib`;
   (d) set the proceedings bibliography file to:

   ```
   \renewcommand{\procbibfile}{\BIBPATH exbibconcat.bib}
   ```

   (e) run LaTeX with the complete bibliography (using the compil=bibmerge option that uses `\nocite{*}`) so bib items are include twice: by the paper and globally. You are now ready to merge bibliographies.
2. merge the bibliographic items (long step):
   (a) first, add the `exbibcommon.bib` file to the list of bibliography files by setting the proceedings bibliography files to:

   ```
   \renewcommand{\procbibfile}{\BIBPATH exbibcommon.bib,%
     \BIBPATH exbibconcat.bib}
   ```

   (b) for each item appearing multiple times:
      i. create a corresponding entry in the `exbibcommon.bib` file;
      ii. remove all of its appearances in `exbibconcat.bib`;
      iii. take this opportunity to correct inconsistent information (title, list of authors, page numbers, etc)! Note that this process requires a lot of time, as it is the slowest in the bibliography merging process.
3. merge the bibliography strings:
   (a) add `exbibstrings.bib` to the list of bibliography files:

   ```
   \renewcommand{\procbibfile}{\BIBPATH exbibstrings.bib,%
     \BIBPATH exbibcommon.bib,\BIBPATH exbibconcat.bib}
   ```

   (b) merge the common strings. For each string shared by several items:
      i. define the corresponding string in the `exbibstring.bib` file. For instance, for the IEEE Transactions on Acoustics, Speech, and Signal Processing, add:

      ```
      @string{IEEE-TASSP = "{IEEE Trans. Acoust., Speech,
        and Signal Proc.}"}
      ```

      ii. use such definitions (*e.g.* `IEEE-TASSP`) to replace any of its appearance in `exbibconcat.bib`. For instance, use:

      ```
      @article{paper027:Mcaulay86,
        Author = {Robert J. McAulay and Thomas F. Quatieri},
        Title = {Speech Analysis/Synthesis Based on a%
          Sinusoidal Representation},
        Journal = IEEE-TASSP,
        Volume = {34},
        Number = {4},
        Pages = {744-754},
        Year = {1986}}
      ```

[20] You may ask authors to do so when sending them editor's notes.

4. optimal: update each paper once the general bibliography is ok:
   (a) for each paper:
      i. generate a new bibliography file (*e.g.* `p_027.bib` for `p_027.tex`) that only includes its non-common bibliography items remaining in `exbibconcat.bib`;
      ii. edit each paper so that it uses both this new bibliography file (`p_027.bib`) together with `exbibcommon.bib` and `exbibstrings.bib`. This will provide common and coherent contents to both local and general bibliographies. Since `p_027.tex` is placed in the `papers/pdftex/p_027/` folder, its bibliography insertion will then become something like:

```
\bibliography{../../exbibstrings.bib,%
    ../../exbibcommon.bib,p_027.bib}
```

   (b) re-run LaTeX on all papers, using the `buildpapers` Unix script (see sec. 7.12). This script also copies all resulting PDFs to the right place.
   (c) if you did not use the previous script, copy all PDF papers to the `papers/` folder. The `buildcppdfpapers` Unix script (see sec. 7.13) can do it for you, for instance if you changed some of the papers but not all, and do not remember which were to be copied.

You are now done with bibliography merging, and are ready to re-run LaTeX on the proceedings using the compil=backref options as many times as necessary to provide proper back-references and page numbering.

5.7.2 Which bib styles for the templates?

Concerning the paper bibliography style, each conference has its own style, often derived from other ones. For instance, the DAFx-06 templates were using the `IEEEbib.bst` style. It however is quite old (1993), and not as compact as the latest `IEEEtran.bst`. As the DAFx proceedings use the order of appearance and not alphabetical sorting (as do the IEEE publications it was inspired from), the more recent `IEEEtranS.bst` style was not suited. The DAFx-06 templates were corrected so as to use `IEEEtran.bst` instead of `IEEEbib.bst` before insertion of papers into the proceedings.

5.7.3 Which bib styles for the general bibliography?

The general bibliography style may be a bit different, as it does not require item numbering in order to not get confused with each paper's bibliography item number. Moreover, alphabetical order is more suited as it simplifies the search for any particular author cited. Therefore, we need to use a bibliographic style on its own, other than the paper templates one!

The style to use has to look more like APA style, with the first author's last name coming first. For that reason, we used the `newapa` style, and derived the `newapave` style with minor cosmetic tweaking (those styles have no numbering, the author list is like "Lastname, F.", etc).

5.7.4 Right-flushing the biblio back-references

Back-references provided by the `hyperref` package are a list of numbers at the end of the bibliographic items (after the last dot). The example using the `newapa` bibliographic style would provide:

> Arfib, D. (1998). Different ways to write digital audio effects programs. In *Proc. of the COST-G6 Workshop on Digital Audio Effects (DAFx-98)*, Barcelona, Spain, (pp. 188–91). 6, 11, 16, 22, 29

We modified `newapa.bst` (resp. `newapa.sty`) by making slight changes (but in many places), and renamed it `newapave.bst` (resp. `newapave.sty`) for the DAFx-06 proceedings. This modification process was carried out to provide some changes and adjustements in the bibliography style and layout[21], as well as right-flushed back-references. Using the `newapa` bibliographic style, the previous example is then modified into something like:

> Arfib, D. Different ways to write digital audio effects programs. In *Proc. of the COST-G6 Workshop on Digital Audio Effects (DAFx-98)*, Barcelona, Spain, pp. 188–91. 1998. 6, 11, 16, 22, 29

Together with the color links, back-references are easier to see when they are right-flushed that when they are left-flushed.

To apply the right-flushed back-references to another style, here is the only trick to keep from the hack. Edit the function that displays the last item of the bibliographic element list (`output.year.check` in our case, because it was reformatted) so as to add a `\hfill` at the end of that command (the year definition in our example):

```
FUNCTION {output.year.check}
{ year empty$
 { ``empty year in `` cite$ * warning$ }
 { write$
   `` (" year * extra.label * `)" *
     mid.sentence 'output.state :=
   }
 if$
}
```

Important note: if the last displayed item (in our case, the year) was not in last position, you also need to edit the following functions defined under the `FUNCTION {name}` format (not exhaustive list): `article`, `book`, `booklet`, `inbook`, `incollection`, `inproceedings`, `manual`, `masterthesis`, `misc`, `phdthesis`, `proceedings`, `techreport`, and `unpublished`. For instance:

```
FUNCTION {misc}
{ output.bibitem
  format.authors output
  author format.key output            % added
  output.year.check                   % added
  title howpublished new.block.checkb
  format.title output
  new.block
  howpublished output
  new.block
  note output
  fin.entry
```

was replaced with:

```
FUNCTION {misc}
{ output.bibitem
  format.authors output
  author format.key output
  title howpublished new.block.checkb
  format.title output
  new.block
  howpublished output
```

[21] no parenthesis around page numbers nor around the year; and year is placed at the end.

```
new.block
note output
output.year.check                    % !!! MOVED !!!
fin.entry
```

The Unix `diff` command may help you to compare the original (`newapa.bst`) and modified (`newapave.bst`) versions of the bibliography style files.

5.7.5 Ensuring that the biblio back-references are right-flushed

With this hack in the bibliography style, all bibliography back-references should appear as right-flushed. However, it sometimes does not work, due to some LaTeX formatting mechanisms I am not competent to identify. Then, sometimes, a list of numbers will see its last item appearing alone on next line, even though there obviously was enough space on the previous line where the other numbers appear. I noticed that some minor reformatting of the concerned bibliographic item could solve this issue. There is no way to automatically do this, nor general rule, only a few tricks I found efficient to solve this issue in 6 items of the DAFx-06 proceedings' general bibliography:

- moving from optional to compulsory a bib item field;
- replacing a `---` by a `--` (arg! so ugly...);
- adding a missing space (*e.g.* between the thesis number and the URL);
- using hyphenation at your advantage: you may sometimes get a reference for which the layout will not hyphen the end of the title, just before the last line (I suspect this is what messes the whole process behind the `\hfill` command).

5.8 LaTeX runs

We now provide the LaTeX compilation steps to build this example (wich is also done by the `buildproc.sh` script provided in section 7.5):

1. generates the first `.aux`, and `.idx` files (use option papers=empty to go faster, and bib=backref for the bibliography):
   `pdflatex example3optim.tex`
2. generates the general bibliography and `.brf`, `.bbl` files:
   `bibtex example3optim`
3. generates the author index:
   `makeindex -s confproc2.ist example3optim.idx`
4. inserts table of contents and index, update their page numbers and internal links for next run (use option papers=final,bib=backref to ensure internal links are correct):
   `pdflatex example3optim.tex`
5. insert table of contents and index with proper page numbers, and remove links to bibliography for next run (papers=final,bib=final):
   `pdflatex example3optim.tex`

6 More about building conference proceedings (for warriors)

6.1 Compilation steps: class option switch

When built, the current example with general bibliography requires various LaTeX runs with different option sets. Since LaTeX-runs can only be automatized with Unix scripts when options are not changed, it is proposed to automatize the option sets switch too. Two files are created, that the script renames when needed, so that the main example file inserts the proper file.

The `example4optim.tex` example file is generated from `example3optim.tex`, after removing the first lines that declare the document class and its options.

6.1.1 Options set for non-final LaTeX runs: `exclasspre.tex`

This first file (`exclasspre.tex`) is used for all LaTeX runs except the final one. In this example, it adds headers on all pages (headers=allpages), and move the footer (movepagenumber) so that we can check page numbers. Also, the option compil=bibbackref creates proper back-references.

```
⟨*exclasspre⟩
\documentclass[letterpaper,10pt,twoside,twosidepapers,%
  electronic,% [printed] | electronic
  papers=countpages,% empty | draft | [final] | countpages
  headers=allpages,% none | pdfonly | exceptpdf | [allpages]
  paperselec=all, %[all] | p_001 | p_fake
  bib=backref,%
  colorheaders=red,%
  verbose,%
  pdftk,%
  pdftkfolder={pdftk_info/},%
  pdftksubject={DAFx-06 Conference},%
  movepagenumber,%
  hyperref={bookmarksdepth=1,bookmarksopen,bookmarksopenlevel=0,%
    citecolor=colorforcite,linkcolor=colorforlink,urlcolor=colorforurl},%
  geometry={text={175truemm,226truemm},% A4 & letter
    inner=0.805in,top=29.15mm,bottom=24.5mm,footskip=9.68mm,voffset=-5mm},%letter
]{confproc}
⟨/exclasspre⟩
```

As previously said, the draft option of `pdfpages` does not generate the bookmark data. So, we do not use it for any of those final LaTeX runs. You can of course use it any time during the layout fine tuning, conference program definition, etc.

6.1.2 Options set for final LaTeX run on the paperback version: `exclasslastpb.tex`

This second file is used for the final LaTeX run: it removes options such as movepagenumber, and uses headers only on the pages where it is necessary (using headers=exceptpdf, as you may have finished the page numberings before). It also uses the compil=last option, in order to insert the last page of each paper with proper back-references generated during the previous LaTeX runs:

```
⟨*exclasslastpb⟩
\documentclass[letterpaper,10pt,twoside,twosidepapers,%
  printed,% [printed] | electronic
  papers=final,% empty | draft | [final] | countpages
  headers=exceptpdf,% none | pdfonly | exceptpdf | [allpages]
  paperselec=all, %[all] | p_001 | p_fake
  bib=last,%
  pdftk,%
  pdftkfolder={pdftk_info/},%
```

```
  pdftksubject={DAFx-06 Conference},%
  binding=5mm,% [0mm]  -> adjust the binding depending on the proceedings thickness
  hyperref={bookmarksdepth=1,bookmarksopen,bookmarksopenlevel=0,%
    citecolor=colorforcite,linkcolor=colorforlink,urlcolor=colorforurl},%
  geometry={text={175truemm,226truemm},% A4 & letter
    inner=0.805in,top=29.15mm,bottom=24.5mm,footskip=9.68mm,voffset=-5mm},%letter
]{confproc}
⟨/exclasslastpb⟩
```

6.1.3 Options set for final LaTeX run on the electronic version: `exclasslastel.tex`

This third file differs from the second by its electronic option, and the geometry settings (the document has no inner margin):

```
⟨*exclasslastel⟩
\documentclass[letterpaper,10pt,twoside,twosidepapers,%
  electronic,% [printed] | electronic
  papers=final,% empty | draft | [final] | countpages
  headers=exceptpdf,% none | pdfonly | exceptpdf | [allpages]
  paperselec=all, %[all] | p_001 | p_fake
  bib=last,%
  pdftk,%
  pdftkfolder={pdftk_info/},%
  pdftksubject={DAFx-06 Conference},%
  binding=0mm,% [0mm] -> no binding for electronic version
  hyperref={bookmarksdepth=1,bookmarksopen,bookmarksopenlevel=0,%
    citecolor=colorforcite,linkcolor=colorforlink,urlcolor=colorforurl},%
  geometry={text={175truemm,226truemm},% A4 & letter
    inner=0.0in,top=29.15mm,bottom=24.5mm,footskip=9.68mm,voffset=-5mm},%letter
]{confproc}
⟨/exclasslastel⟩
```

6.2 Option management: examples of option combinations

6.2.1 Fasten compilation steps

`papers=draft` `papers=empty`

As the LaTeX run may be long when only making a small change, we may want to speed up the process by using the papers=draft or even better with papers=empty (see sec. 4.2.3). This is useful for instance when making layout changes, editing the welcome letters, or working on generating proper page numbering. This will replace each PDF page by an almost blank page. The difference between the two is that papers =draft uses the draft mode of `pdfpages`, whereas the papers =empty options uses an internal command, and displays more information about each `.pdf` file: paper title, author list, file name, tags of bibliography items, and page number. Note that both preserve the bookmark, and empty is much faster than draft, which is already way faster than last.

`final`

The other possible compilation option is final. Note that it is configured by default depending on the compil option you used, but can be modified anyway.

6.2.2 Verbose/debug

`verbose` `debug`

Also, the verbose or debug option adds some debug comments in the LaTeX console, both from `confproc` and `hyperref` packages, that might help to track problems if any. It can be used at any compilation step, of course!

6.2.3 Clear single/double page

Depending on wether your document is oneside or twoside, you may want to force it to always:

cleardoublepage

- clear double page after each paper in 1-side mode using cleardoublepage (used with oneside);

clearsinglepage

- not clear double page after each paper in 2-side mode using clearsinglepage (used with twoside).

6.3 Steps to generate the final version of the proceedings

Now that all options and commands to build proceedings are know, we need more insight about what to do with which option/command and when. This is the purpose of this section, that proposed some building stages (as depicted in the diagram in Fig. 1) to produce both the electronic and paperback final versions of the provided example proceedings with the following constraints:

- the template for papers has a header and footer, so the proceedings must have the same header/footer;
- a general bibliography is used;
- the final PDF papers must be named after their first page number.

6.3.1 Generate the program and the paper switch

Once the general proceedings structure is there (see the `example3optim.tex` file), the first step is to generate the conference program and its corresponding paper switch:

- by hand (read sec. 5.2 for an example);
- using the `generateswitch.pl` Perl script described in sec. 7.4 to generate both the `exsessions.tex` and `expapersswitch.tex` files from your `exprogram.csv` program file.

6.3.2 Make sure each PDF is fully inserted

At some point, we want to make sure all pages of each paper are inserted. The simplest way is to always use the papers=countpages, but this is unfortunately very time consuming when working on setting various layout aspects, the program, and also when adding contents to the preamble. Therefore, if using papers=empty for instance, we want to make sure no paper has been truncated by error. Checking this by hand is a long and fastidious task, so two solutions are offered: the `countnbpages.sh` script (see sec. 7.7) and the `*.np*` files generated when running LaTeX with either papers=countpages or papers=empty.

For Unix users:

1. launch once the `countnbpages.sh` script (see sec. 7.7) to generate `example.npc`, which contains each paper's real number of pages;

papers=empty

2. run pdflatex onto `example.tex` with papers=empty (faster mode to emulate paper insertion with your settings such as the number of pages); it generates a `example.nps` file with the user-defined number of pages;
3. compare the content of both files, using

   ```
   diff example.nps example.npc
   ```

4. each paper appearing in the `diff` command output has a page number discrepancy, that has to be corrected.

Once all discrepancies have been corrected, re-do this procedure once for a last check, just in case!

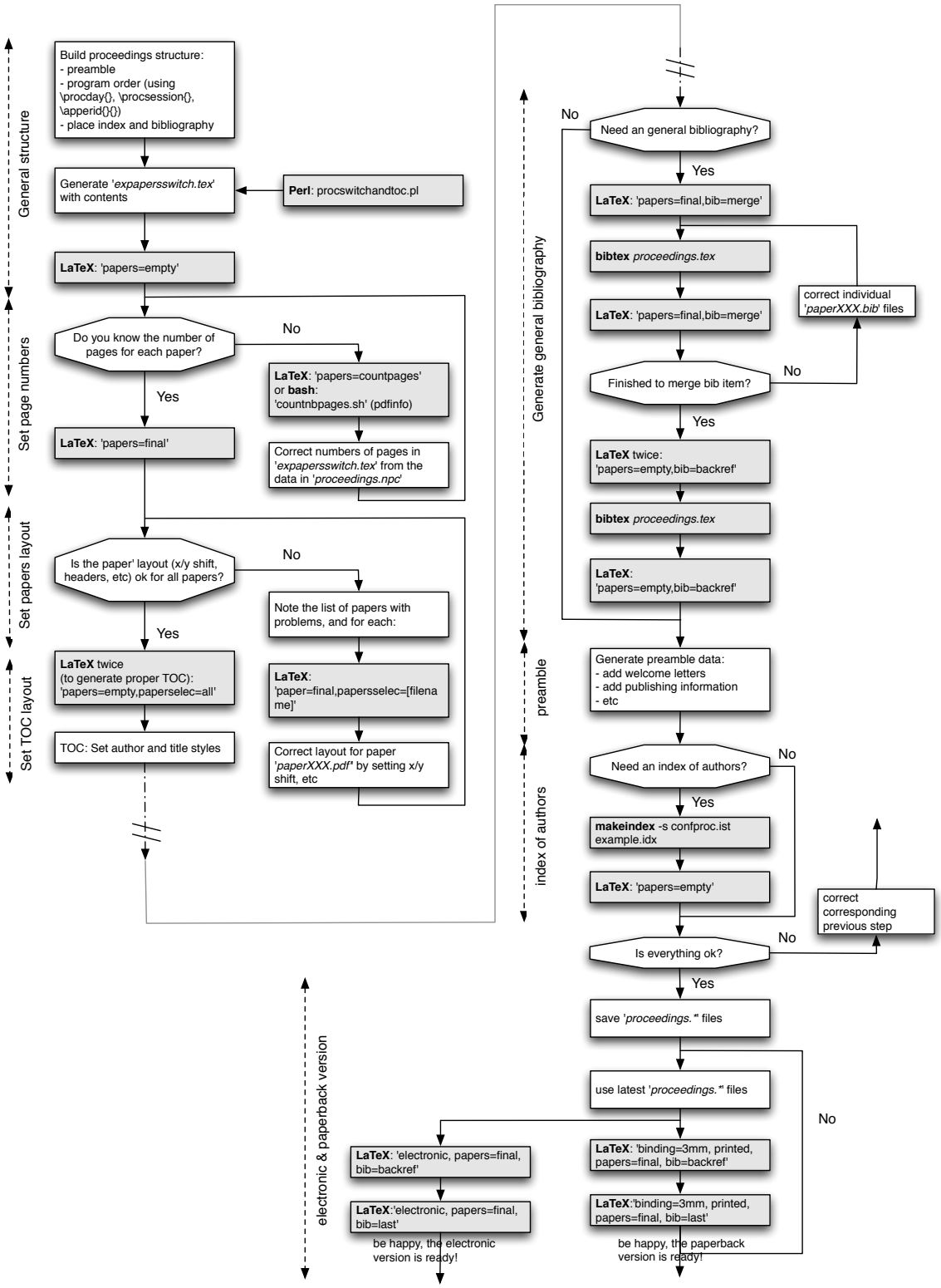

Figure 1: Suggested stages to build proceedings

For non-Unix users (sorry, it's much slower!):

papers=countpages

1. run pdflatex onto `example3optim.tex` with papers=countpages (real full paper insertion); it generates `example.npc` (real number of pages);

papers=empty

2. run pdflatex onto `example3optim.tex` with papers=empty (faster mode to emulate paper insertion with your selected number of pages); it generates `example.nps` (supposed number of pages);
3. compare both files (with your favorite text editor's 'compare file' function).

If some page numbers indicated in `expapersswitch.tex` are not identical to the real page number, we just edit the file to correct the error, and can now g on with building the proceedings.

6.3.3 Changing papers' first page number

When the paper template includes the page number in the footer, we need to correct each individual paper's first page number as it appears in the proceedings' table of contents[22]. To do so:

1. make at least two runs with the following options:

   ```
   \documentclass[a4paper,10pt,twoside,twosidepapers,%
     compil=last,headers=allpages,movepagenumber,electronic]{confproc}
   ```

   to include all papers and build a table of contents with proper page numbers.

2. prepare each paper for insertion. There are two ways to do this:
   (a) lazy way: use the `\setcounter{page}{XXX}` command in each paper, with `XXX` replaced by the real number;
   (b) alternative way (simplifies program changes): centralize page numbers in the `expages.tex` file, organized by the paper ID. Then, the two steps are:
   - add the following in the preamble of each paper:
     `\input{../../expages.tex}\setpagenumber{04}`
     Here, the ID paper is `04`, and has to be updated for each paper.
   - update the `expages.tex` file for each paper: set its first page number as it appears in the table of contents.

   By doing so, we can update the program to re-build the table of contents as many times as wanted, without having to re-edit each paper.
3. once the program and the corresponding paper order are ok, (re)generate each paper independently with proper first page number (using the `buildpapers.sh` script in sec. 7.12);
4. check that no error was made when numbering the first page. Run LaTeX with at least the headers=allpages,movepagenumber options. In case of error, re-do step 2–3 till the page numbers are ok.

6.3.4 Renaming papers

Renaming all papers according to their first page number (*e.g.* `p_NNN.pdf` if only PDF files are renamed) is very helpful to ensure the proceedings' CD version is ISO compliant, and has file names with less than 8 characters (+ extensions). This can only be done once the program is definitive. Then file names can be changed accordingly in the `.csv` file; the `expapersswitch.tex` file has to be re-generated, and the proceedings rebuilt. It is easily done using the Unix scripts.

[22] When clicking on a paper, the PDF file of this paper will open with the same first page number. Also, if the conference papers are available on the web, knowing the proper page numbers will help readers to properly cite them.

6.4 Quality and production

We present here some other ideas dealing with the production and the quality of the proceedings. Indeed, to provide the best possible quality proceedings, individual papers may be edited (see sec. 6.4.1), which can be simplified by sending notes to authors before they submit the final version (see sec. 6.4.2). Using only LaTeX to typeset may require to convert some Word files to LaTeX, in the case where the proceedings templates are provided in the 2 formats to authors (see sec. 6.4.3). The last comments are about the graphical quality (sec. 6.4.4) and the necessary font embedding in the PDF images (see sec. 6.4.5).

6.4.1 Editing the papers

To provide the best possible quality proceedings for DAFx-06, each individual paper was edited by checking the following items (non-exhaustive list):

- proper page format used (US letter instead of A4 format), as the authors were from various continents around the worlds;
- break the title line at the right place using `\break`;
- affiliation type is properly indicated;
- affiliation is properly layed out;
- author's email exists and works;
- captions are italic, with a "." at the end;
- all figures are referenced in the text;
- bibliographic items have a volume and number, as well as page number or preprint number (AES convention);
- bibliographic items use generally defined strings, and are identical each time they are cited;
- math units: Physics convention is roman, not italic (*i.e.* not LaTeX's math style). Ex: 5 Hz, and not $5Hz$.

So as to ensure papers look uniform as possible, we changed for each:

- the URL font to sans-serif, as its default font is too wide. We added the following command in the preamble of each paper:
  `\usepackage{url}\urlstyle{sf}`
- all `\href{}{}` commands related to URL (*i.e.* all except emails) where converted to URL, as it is more apropriated (it does the hyphenations for you and most of the time it does it better).

Some not-so-minor comments:

- to do a valid line break in the paper title (at least with the `dafx06.sty` style, but not only) use `\break`, instead of `\newline`, `\\`, or `\linebreak` (that creates unbalanced titles). That way, it works similarly for both the title and the pdftitle in the metadata.
- using the `balance.sty` package allows to well balance the last page, which is especially useful for the bibliography.

6.4.2 Sending editing notes to authors to improve the layout quality

In order to improve aspects of the quality of the proceedings, we listed many common errors and gave a feedback to authors of all accepted papers, that they were kindly asked to take care of. This is how we proceeded:

1. examine all papers and list the common errors and electronic paper info (PDF version, PDF generator, valid hyperref, etc.) (10h);
2. in an `.csv` file, indicate all problems, paper's title, index and author's email (0.5h);
3. column by column, fill in the data (30h) with errors detected in each paper;
4. use a combination of Perl script and an AppleScript to convert info in this file into usual sentences and indications of what to do in order to improve the paper quality (4h), and then into a series of emails, ready to be sent to authors (4h).

N.B.: Those scripts are not provided in the package, but can be obtained on demand.

6.4.3 Manual Word to LaTeX conversion

To automatize all the processes in proceedings making, we may want to convert all non-LaTeX generated documents into LaTeX documents. If that cannot be asked to the conference authors, we need to do it ourselves. Here is an example of the steps to follow:

1. copy and paste the whole text;
2. update the header (author, title, affiliation);
3. add sections, subsections, etc. according to the original text;
   update labels and references for sections, subsections, etc.;
4. insert figures and tables with the proceedings template style;
   update labels and references for figures and tables;
5. update captions with the proceedings template style;
6. edit equations (inside the text and as separated formulae);
   update labels and references for equations ;
7. replace all Word quotes by LaTeX quotes (double "", and single '' quotes) to avoid they disappear (Unicode-related issue);
8. correct any specific formatting such as italic, capitals, bold, etc;
9. remove useless hyphenations "-" produced as line breaks by Word;
10. replace remaining hyphens by the proper corresponding one: dash '-' (hyphen), semi-quadratin '–' (ex: number range) and quadratin '—'.

6.4.4 How to ensure the graphical quality?

The best way to ensure excellent quality for you graphics in the electronic version of you proceedings consists in using vectorial images, *i.e.* postscript (`.ps` or `.eps`) or `.pdf` files. It should be the same for the printed version, except that the font problem with Matlab described in sec. 6.4.5 may imply to convert vectorial images to bitmap images (such as `.png` or `.gif`).

6.4.5 How to ensure your fonts are embedded in the PDF?

Various important things have to be checked to ensure a great PDF file quality. (Please read www.prepressure.com/pdf/basics/preflight for more information). One of these important things is the font embedding. If not checked, you may end up with a document where fonts will disappear and be replaced by random characters (which are not random in fact)[23]. Unfortunately for non-experts user of Matlab, the system fonts such as Arial or Helvetica are not embedded by default in the `.pdf` nor in the `.eps` file. This can be checked by converting any of the two into another format using Ghostscript. For instance, converting a `.pdf` to `.ps` using `pdf2ps` will show the following log info:

```
**** Warning: Fonts with Subtype = /TrueType should be embedded.
             The following fonts were not embedded:
                    Arial-ItalicMT
                    ArialMT

**** This file had errors that were repaired or ignored.
**** The file was produced by:
**** >>>> pdfTeX-0.14h <<<<
**** Please notify the author of the software that produced this
**** file that it does not conform to Adobe's published PDF
**** specification.
```

This can also be checked by processing a PDF files produced by Matlab using Acrobat Distiller ($), and you will get the same errors...

A way to correct this in Matlab is use the `print_pdf.m`[24] file by Oliver Woodford to save the figure as PDF with embedded fonts.

If you cannot have it fixed by the author, then you need to use Acrobat Professional to do the job for you. Utilities like `pdfinfo` and `pdffonts` are helpful to detect such problems as well as missing bounding boxes, with commands like:

```
pdffonts [filename.pdf]
pdfinfo -box [filename.pdf]
```

7 Various scripts and utilities to save time

7.1 `buildcls.sh`: Build the class documentation and files

Name `buildcls.sh`

Type Unix/bash script

Purpose generates the class files and documentation, and prepares the example-related files.

Note first set the path to LaTeX 2ε binaries before using it!

```
666 ⟨*buildcls⟩
667 #!/bin/sh
668
669 wd=‘pwd‘
670
```

[23] Indeed, when printing on a system that is not yours (*e.g.* in a professional print center), the printer may be set such as not to replace a missing font by a similar one. This is why, Matlab text can be totally scrapped, replaced by other numbers, letters, and so on!

[24] get it at: www.mathworks.com/matlabcentral/fileexchange/22018-printpdf)

First, you may set the path to LaTeX 2ε binaries:

```
#-- set path to LaTeX binaries
LaPath="/usr/texbin/" # TeXLive

```

and then, only if necessary, change the names to the LaTeX compilers:

```
#-- set names of LaTeX and related compilers
Latex=$LaPath"pdflatex"
Index=$LaPath"makeindex"
```

as well as the document and example target names:

```
Target="confproc"  #- set document's name
extarget="example/" #- set the example folder name

```

We can start building the documentation and the `.ins` file:

```
#-- build doc, class and example files
$Latex $Target.dtx #- build doc. and .ins file
$Latex $Target.ins #- build class and example files

```

We then rename the bibliography style (HACK: the file is generated under the `newapave2.sty`, because if there already exist a `newapave.sty` on your system, it will not be generated again under that name; we now have to properly rename it):

```
#-- HACK: rename newapave2.sty
mv newapave2.sty newapave.sty

```

Once it is done, we can finish the documentation. this full sequence is only necessary if you generate the implementation, index and changes history:

```
cd $wd/
#-- finish to build the documentation
$Latex $Target.dtx #- re-run doc for toc update
$Latex $Target.dtx #- re-run doc for proper back-references
$Index -s gind.ist $Target  #- with \CodelineIndex of \PageIndex
$Index -s gglo.ist -o $Target.gls $Target.glo  #- with \RecordChanges
$Latex $Target.dtx #- insert index & list of changes, re-number
$Latex $Target.dtx #- last run with proper page numbers

```

Since there are 2 scripts, one to install (this one) and one to clean up all the mess (mainly used by me during building tests), we also prepare the latter:

```
#-- prepare scripts for cleaning package
cd $wd
chmod +x cleancls.sh

```

We then create the example folder and move all related files thanks to the `prepareexample.sh` script (sec. 7.3):

```
#-- prepare scripts for building example
chmod +x prepareexample.sh
./prepareexample.sh

```

By uncommenting the last line, you will also build the example!

```
#-- build example
cd $extarget
#./buildproc.sh
```
⟨/buildcls⟩

This script is generated by the first LaTeX run on `confproc.dtx`. You then have to change its permission in the `bash` shell to make it executable:

```
chmod +x buildcls
```

Then, you can run it from the `bash` shell:

```
./buildcls
```

7.2 `cleancls.sh`: Clean up the class folder

Name `cleancls.sh`

Type Unix/bash script

Purpose cleans up the folder where the class was generated.

```
⟨*cleancls⟩
#!/bin/sh

```

Create a back up folder:

```
mkdir backup #--- move the files to be kept
```

Move original class documentation materials:

```
mv confproc.dtx backup/
mv confproc_diag.pdf backup/
mv confproc-short.tex backup/
mv confproc-short.pdf backup/
```

Move building and cleaning scripts:

```
mv buildcls.sh backup/
cp cleancls.sh backup/
```

Cleanup and place back the original minimal file set:

```
rm *.* #--- clean up!
mv backup/* . #--- move the backed up files
rm -r backup #--- remove the temporary backup folder
⟨/cleancls⟩
```

You may want to use it to re-generate the whole package from the `.dtx` file. Note that this script too is generated by the first LaTeX run on the `confproc.dtx` file.

7.3 `prepareexample.sh`: Prepare the example files, scripts and folders

Name `prepareexample.sh`

Type Unix/bash script

Purpose prepares the example-related files, scripts and folders

Note first set the path to LaTeX 2_ε binaries before using it!

```
%    \changes{0.7}{2010/08/05}{added \file{prepareexample.sh}}
⟨*prepareexample⟩
#!/bin/sh

wd=‘pwd‘

```

Set the example folder name:

```
extarget="example" #- set the example folder name
```

Then create the example folder:

```
#-- prepare scripts for building example
mkdir $extarget #- create the folder

#-- generate the program session files
perl generateswitch.pl<exprogram.csv
```

and move the example-related files and scripts:

```
mv ex*.* $extarget/ #- move all other example files into proper folder
mv buildproc* $extarget/ #- move scripts into it
mv buildcppdfpapers* $extarget/
mv buildpapers* $extarget/
mv countnbpages.sh $extarget/
mv generateswitch.pl $extarget/
mv papersinfo.sh $extarget/
mv paperssplitpreamble.sh $extarget/
mv removeLaTeXcmds.sh $extarget/
```

Also, copy folder and files that are common to the class documentation and the example:

```
cp -r pictures $extarget/ #- copy pictures into it
cp -r papers $extarget/ #- copy papers into it
cp confproc.cls $extarget/ #- copy the class into it
cp confproc*.ist $extarget/ #- copy the index style into it
cp newapave* $extarget/ #- copy the newapave bib style files
```

Move the `expages.tex` generated file to the right place, and prepare the `pdftk` information folder:

```
#-- prepare building scripts and generate needed directories
cd $wd/$extarget
mkdir pdftk_info/
mv expages.tex papers/
```

Change the permission of the example-related scripts:

```
cd $wd/$extarget
chmod +x buildproc*
chmod +x generateswitch.pl
chmod +x exportIndividualPDFs.sh
chmod +x papersinfo.sh
chmod +x paperssplitpreamble.sh
cd ..
```

By uncommenting the last line, we also build the example!

```
#-- build example
cd $wd/$extarget
#./buildproc.sh
```
⟨/prepareexample⟩

N.B.: This script is generated by the first LaTeX run on `confproc.dtx`. You then have to change its permission in the `bash` shell to make it executable:

```
chmod +x prepareexample.sh # change its permissions
```

Then, you can run it from the bash shell:

```
./prepareexample.sh
```

7.4 generateswitch.pl: Generate the paper switch and program [Perl]

Name generateswitch.pl

Type Perl script

Purpose generates TOC-related example files: paper switch in expapersswitch.tex and papers order by sessions and days in exsessions.tex.

to use it, type in a bash terminal: perl generateswitch.pl<exprogram.csv (or any other program file).

```
⟨*generateswitch⟩
#!/usr/bin/perl -w

# generateswitch.pl
#    created as dafxproctoc.pl by Marc Zadel, 2006-04-28
#    modified for confproc.cls by Vincent Verfaille, 2007-08-08 (v0.4) & 2009-10-30 (v0.7)
# Execute as
# ./generateswitch.pl < intputfile.txt >

use strict;
use Text::ParseWords;
open(SWI, ">expapersswitch.tex"); #open for write, overwrite
open(SESSIONS, ">exsessions.tex"); #open for write, overwrite

# ----- Configuration
# field separator for the input file
my $fieldseparator=',';

# mac line endings: "\r"  / Unix line endings: :\n"
$/ = "\n";  # line endings for the input file
$\ = "\n";  # line endings for the output file

# ----- Subroutines
# -- split one line of input into a hash with named fields
sub parseinputline {
  my ($inputline) = @_;

  # escape single quotes on the input line: they interfere with quotewords()'s
  # quote handling (ie, they start to quote stuff)
  $inputline =~ s/'/\\'/g;

  # parse the input line
  my @wordlist = &quotewords($fieldseparator, 0, $inputline);

  # replace accented characters with latex escaped equivalents. Use it after
  # quotewords() so the '\' don't get interpreted by quotewords() as escapes
  foreach my $word ( @wordlist ) {
    if ( $word ) { $word = &latexifyaccentedcharacters($word); }
  }

  # extract the fields into local variables. Author names stored as a list
```

```
  my ($type, $number, $pcdecision, $nbpages, $title, $filename,
      $generatedfrom, $cite) = @wordlist;

  # remove the first 8 elements (just parsed out), leaving only author names.
  # reminder: list of 8 scalars, though some may be "" if less than 4 authors
  splice( @wordlist, 0, 8 );

  # store the author names as a list of lists. We end up with a list that looks
  #  like ((Udo,Zoelzer),(Daniel,Arfib))
  my @authors = ();
  while ( $wordlist[0] ) {
    push( @authors, [splice( @wordlist, 0, 2 )] );
    # "splice( @wordlist, 0, 2 )": cuts the first 2 scalars off of @wordlist
    # and returns them; calling [splice(@wordlist,0,2)] returns a *reference*
    # to a list containing the first two scalars. (see perldoc perldsc.)
  }

  # create a hash reference containing the named fields and return it
  my $fields = {
    type    => $type,
    number    => $number,
    pcdecision  => $pcdecision,
    nbpages     => $nbpages,
    title     => $title,
    generatedfrom => $generatedfrom,
    filename  => $filename,
    cite    => $cite,
    authors   => \@authors,
  };
  return $fields;
}

# -- takes a string in Mac OS Roman encoding and encode the accented
#  characters with latex escapes (only for a subset of available characters).
sub latexifyaccentedcharacters {
  # for mapping between unicode and mac os western encoding, see:
  # http://www.unicode.org/Public/MAPPINGS/VENDORS/APPLE/ROMAN.TXT
  my ($inputstring) = @_;
  $inputstring =~ s/\x8a/\\"a/g;  # \"a: unicode 0xe4, mac os western 0x8a
  $inputstring =~ s/\x87/\\'a/g;  # \'a: unicode 0xe9, mac os western 0x87
  $inputstring =~ s/\x88/\\`a/g;  # \`a: unicode 0xe8, mac os western 0x88
  $inputstring =~ s/\x8e/\\'e/g;  # \'e: unicode 0xe9, mac os western 0x8e
  $inputstring =~ s/\x8f/\\`e/g;  # \`e: unicode 0xe8, mac os western 0x8f
  $inputstring =~ s/\x91/\\"e/g;  # \"e: unicode 0xeb, mac os western 0x91
  $inputstring =~ s/\x97/\\'o/g;  # \'o: unicode 0xf3, mac os western 0x97
  $inputstring =~ s/\x98/\\`o/g;  # \`o: unicode 0xf2, mac os western 0x98
  $inputstring =~ s/\x9a/\\"o/g;  # \"o: unicode 0xf6, mac os western 0x9a
  $inputstring =~ s/\x99/\\^o/g;  # \^o: unicode 0xf4, mac os western 0x99
  $inputstring =~ s/\xbf/\\o /g;  # \o:  unicode 0xf8, mac os western 0xbf
  $inputstring =~ s/\x96/\\~n /g;  # \~n:  unicode 0xF1, mac os western 0x96
  $inputstring =~ s/\x94/\\^{\\i}/g;  # \^{\i}: unicode 0xee, mac os western 0x94
  $inputstring =~ s/\x/\\i/g;  # \i: unicode , mac os western
  $inputstring =~ s/\x9f/\\"u/g;  # \"u: unicode 0xfc, mac os western 0x9f
  $inputstring =~ s/\x5c/\\/g;  # \: unicode 0x5C, mac os western 0x5C

  return $inputstring;
```

```
}

# -- output the information for a day
sub outputdaylatex {
  my ($fields) = @_;
  my $sessiontitle = $fields->{'title'};
  open(SESSIONS, ">>exsessions.tex"); #open for append
  print SESSIONS ' ';
  print SESSIONS '%%%== Day';
  print SESSIONS '\procday{', $sessiontitle, '}'
}

# -- output the information for a session line
sub outputsessionlatex {
  my ($fields) = @_;
  my $sessiontitle = $fields->{'title'};
  open(SESSIONS, ">>exsessions.tex"); #open for append
  print SESSIONS ' ';
  print SESSIONS '%%%-- session';
  print SESSIONS '\session{', $sessiontitle, '}'
}

# -- in: ref. to a list of lists of author names ((Udo,Zoelzer),(Daniel,Arfib))
# out: ref. to a Perl list w/ entries "Udo Zoelzer" and "Daniel Arfib" (no quotes)
sub authorsbyfirstname {
  my ($authors) = @_;
  # generate a list of full "first last" author names
  my @authorlistbyfirstname = map { "$_->[0] $_->[1]" } @$authors;
  return \@authorlistbyfirstname; # return a ref. to the new list of authors
}

# -- in: ref. to a list of lists of author names ((Udo,Zoelzer),(Daniel,Arfib))
# out: ref. to a Perl list w/ entries "Zoelzer, Udo" and "Arfib, Daniel"
sub authorsbysurname {
  my ($authors) = @_;
  # generate a list of authors with surnames written first
  my @authorlistbysurname = map { "$_->[1], $_->[0]" } @$authors;
  return \@authorlistbysurname; # return a ref. to the new list of authors
}

# -- in: ref. to a list of author names: "Zoelzer, Udo" and "Arfib, Daniel"
# out: LaTeX index entries: "\index{Zoelzer, Udo}\index{Arfib, Daniel}"
sub genindex {
  my ($authorsbysurname) = @_;
  my @indexentries = map { "\\index{$_}" } @$authorsbysurname;
  return join('', @indexentries);
}

# -- in: ref. to a list of author names: "Zoelzer, Udo" and "Arfib, Daniel"
# out: bookmarks cmds: "\pdfbookmark[2]{Udo Zoelzer}{#2.Udo Zoelzer}
# \pdfbookmark[2]{Daniel Arfib}{#2.Daniel Arfib}"
sub genbookmark {
  my ($authorsbyfirstname) = @_;
  my @indexentries = map { "\\pdfbookmark[2]{$_}{#2.$_}" }
      @$authorsbyfirstname;
  return join('', @indexentries);
```

```
}

# -- output the information for a paper line
sub outputpaperlatex {
  my ($fields) = @_;
  open(SWI, ">>expapersswitch.tex"); #open for append
  print SWI '%======= PAPER ID = ', $fields->{'number'}, ' =======';
  print SWI '\ifnum\paperswitch=', $fields->{'number'};
  print SWI '  \procpaper[xshift=\LaTeXxShift{}, yshift=\LaTeXyShift{}, npages=',
    $fields->{'nbpages'}, ', switch=\paperswitch,%';
  print SWI '    title={', $fields->{'title'}, '},% paper title';
  print SWI '    author={', join( ', ', @{&authorsbyfirstname($fields->{'authors'})}),
  '},% list of authors';
  print SWI '    index={', &genindex(&authorsbysurname($fields->{'authors'})),
  '},% authors index entries';
  print SWI '    cite={', $fields->{'cite'}, '},% cited bib items';
#  print SWI '  {#2}{\paperbookmark}';
  print SWI '    bookmark={', &genbookmark(&authorsbyfirstname($fields->{'authors'})),'}% for Pl
  print SWI '  ]{#2}';
  print SWI '\fi';
  print SWI ' ';
  open(SESSIONS, ">>exsessions.tex"); #open for write, overwrite
  print SESSIONS '\paperid{', $fields->{'number'}, '}{', $fields->{'filename'}, '}';
}

# ----- Main
# FIXME: parse a line, and confirm that all of the fields are set up properly
# --> correct number of fields, and the fields have the correct values
open(SWI, ">>expapersswitch.tex"); #open for write, overwrite
print SWI '\newcommand{\paperid}[2]{';
print SWI ' ';
print SWI '\renewcommand{\paperswitch}{#1}';
print SWI ' ';

while ( <> ) {
  chomp; # clear the newline character from the end of the line
  my $fields = &parseinputline($_);   # parse the line into fields
  # take some action depending on what type of line it is; case insensitive
  if ( lc($fields->{'type'}) eq lc('day') ) {
  &outputdaylatex($fields);
  } elsif ( lc($fields->{'type'}) eq lc('session')
      || lc($fields->{'type'}) eq lc('paper session')
      || lc($fields->{'type'}) eq lc('demo session')
      || lc($fields->{'type'}) eq lc('poster session') ) {
  &outputsessionlatex($fields);
  } elsif ( lc($fields->{'type'}) eq lc('oral')
      || lc($fields->{'type'}) eq lc('paper')
      || lc($fields->{'type'}) eq lc('demo')
      || lc($fields->{'type'}) eq lc('poster') ) {
  &outputpaperlatex($fields);
  } elsif ( lc($fields->{'type'}) eq lc('Type')) {
  } else { print '!!! a day, session or paper (',
      $fields->{'type'},') is lost by the script...';
  }
open(SWI, ">>expapersswitch.tex"); #open for append
}
```

```
print SWI '}';
close(SWI);
close(SESSIONS);
```
⟨/generateswitch⟩

7.5 `buildproc.sh`: Build the proceedings

Name `buildproc.sh`

Type Unix/bash script

Purpose describes all compilation steps to produce the final version of the proceedings

Note This script is the most important!

This script applies several LaTeX runs to create valid table of content, index, bibliography, index of authors, and proper back references from the bibliography. It also manages the renaming of the class insertion file, so we do not need anymore to run a last time by hand after changing the compil=backref option to compil=last (as this option change, and others, are in the `exclasspre.tex` and `exclasslast.tex` files).

⟨∗buildproc⟩
```
#!/bin/sh

```

We set the user-dependent the original file name (no extension):

```
#--- set user dependent file name
TEXFILE="example3optim"
```

Let us now set system-dependent variables: the path the the LaTeX distribution as well as the binaries:

```
#--- set system-dependent variables
LATEXPATH="/usr/texbin/" # for TexLive
#--- set compilers' paths
PDFLATEX=$LATEXPATH"pdflatex"
BIBTEX=$LATEXPATH"bibtex"
MAKEINDEX=$LATEXPATH"makeindex"
mkdir pdftk_info/

```

- we copy the class options (including bib=backref) to the file called by the core file:

```
#--- class settings: "empty" option and binding
cp exclasspre.tex exclass.tex
```

- we run LaTeX once to generate the table of contents:

```
#--- Compile
separator='--------------------------------------------------'
echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** PdfLaTeX: create toc (1/6) ***'
$PDFLATEX  $TEXFILE.tex

```

- we run bibTeX once to generate the bibliography:

```
echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** Bibtex: generate the general biblio. (2/6) ***'
$BIBTEX $TEXFILE

```

- we run `makeindex` once to generate the author index:

```
echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** Makeindex: create index of authors (3/6) ***'
$MAKEINDEX -s confproc2.ist $TEXFILE.idx

```

- we run LaTeX once to insert the table of contents, index and bibliography, and update their page numbers for next run:

```
echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** PdfLaTeX: add toc + insert index and bibliography (4/6) ***'
$PDFLATEX  $TEXFILE.tex

```

- we run LaTeX once again to insert update page numbers in the table of contents, index and bibliography:

```
echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** PdfLaTeX: createupdate toc, index and bib page numbers (5/6) ***'
$PDFLATEX $TEXFILE.tex

```

- we do the final LaTeX run:

```
echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** PdfLaTeX: mod. class insertion, for proper PDF links for full papers (6/6) ***'
$PDFLATEX $TEXFILE.tex
⟨/buildproc⟩
```

7.6 `buildprocelpb.sh`: Generate both paperback and electronic final versions

Name `buildprocelpb.sh`

Type Unix/bash script

Purpose describes all compilation steps to produce the final version of both paperback and electronic proceedings

Note This script helps to automatize the whole production process; its is described/commented as an incrementation from `buildproc.sh`.

We first set the user-dependent file names and folders, namely:

```
%    \changes{0.7}{2010/08/05}{added \file{buildprocelpb.sh}}
%    \color{black!50}
⟨*buildprocelpb⟩
#!/bin/bash

#--- set user dependent file name
```

- the original file name (no extension):

```
INTEXFILE="example4optim"
```

- the target file name (no extension):

```
TEXFILE="proceedings"
```

- the working directory:

```
TEXFILEPATH="example"
```

- the target directories for paperback and electronic versions:

```
PAPERBACKFOLDER="PDF_printed/"
ELECTRONICFOLDER="PDF_electronic/"

```

We then define the names of LaTeX files containing the class options before the last run (ie. with bibliography back-references):

```
#--- different class options for electronic vs paperback version
class_paperback_pre=exclasspre
```

and the names of LaTeX files containing the class options for the last run for both paperback and electronic versions:

```
class_paperback_final=exclasslastpb
class_electronic_final=exclasslastel

```

Let us now set system-dependent variables: the path the the LaTeX distribution as well as the binaries:

```
#--- set system-dependent variables
LATEXPATH="/usr/texbin/" # TexLive

#--- set compilers' paths
PDFLATEX=$LATEXPATH"pdflatex"
BIBTEX=$LATEXPATH"bibtex"
MAKEINDEX=$LATEXPATH"makeindex"

```

We then define temporary folders for splitting the proceedings back into separate files with proper PDF metadata:

```
#--- set script-specific paths
GPATH=‘pwd‘ # general proc path
PAPERBACKFOLDER=${GPATH}/${PAPERBACKFOLDER}
ELECTRONICFOLDER=${GPATH}/${ELECTRONICFOLDER}
PDFPATH="${ELECTRONICFOLDER}/papers"
PDFTKPATH="pdftk_info/"
INPATH="tmp/papersinfo/"
SPPATH="tmp/papers_split/"

```

Let us now create the needed folders:

```
#=== prepare output folders
mkdir -p ${PAPERBACKFOLDER}
mkdir -p ${ELECTRONICFOLDER}
rm -r ${ELECTRONICFOLDER}/papers/
mkdir -p ${ELECTRONICFOLDER}/papers/
mkdir -p $INPATH
mkdir -p $SPPATH
mkdir -p $PDFTKPATH

```

We can now move to the working directory and start building the paperback version:

```
#=== GO TO LaTeX FOLDER !!!
cd ${GPATH}

```

- we generate the LaTeX file by concatenating the class options (including bib=backref) with the file core:

```
#=== MAKE PAPERBACK VERSION
#--- class settings: "empty" option and binding
cat ${class_paperback_pre}.tex ${INTEXFILE}.tex >${TEXFILE}.tex

```

- we run LaTeX once to generate the table of contents:

```
#--- Compile
separator='------------------------------------------------------'
echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** PdfLaTeX: create toc (1/6) ***'
$PDFLATEX  $TEXFILE.tex

```

- we run bibTeX once to generate the bibliography:

```
echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** Bibtex: generate the general biblio. (2/6) ***'
$BIBTEX $TEXFILE

```

- we run `makeindex` once to generate the author index:

```
echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** Makeindex: create index of authors (3/6) ***'
$MAKEINDEX -s confproc2.ist $TEXFILE.idx

```

- we run LaTeX once to insert the table of contents, index and bibliography, and update their page numbers for next run:

```
echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** PdfLaTeX: add toc + insert index and bibliography (4/6) ***'
$PDFLATEX  $TEXFILE.tex

```

- we run LaTeX once again to insert update page numbers in the table of contents, index and bibliography:

```
echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** PdfLaTeX: createupdate toc, index and bib page numbers (5/6) ***'
$PDFLATEX $TEXFILE.tex

```

- we generate the final LaTeX file by concatenating the class options (with bib=final being the only diference, otherwise you may break more than just the back-references and get a non-functioning table of content and index series of links) with the file core:

```
#--- class settings: "final" option and binding
cat ${class_paperback_final}.tex ${INTEXFILE}.tex  >${TEXFILE}.tex

```

- we do the final LaTeX run:

```
echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** PdfLaTeX: mod. class insertion, for proper PDF links for full papers (6/6) ***'
$PDFLATEX $TEXFILE.tex

```

- and save the file in the appropriate folder:

```
#--- save PDF
cp ${TEXFILE}.pdf $PAPERBACKFOLDER/${TEXFILE}.pdf

```

The process is the exact same for the electronic version:

```
#=== MAKE ELECTRONIC VERSION FOR CD, FROM PAPERBACK VERSION
#--- class settings: "final" option and no binding
cd ${GPATH}/${TEXFILEPATH}
cat ${class_electronic_final}.tex ${INTEXFILE}.tex  >${TEXFILE}.tex

#--- Compile
echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** PdfLaTeX: create toc (1/6) ***'
$PDFLATEX  $TEXFILE.tex

echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** Bibtex: generate the general biblio. (2/6) ***'
$BIBTEX $TEXFILE

echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** Makeindex: create index of authors (3/6) ***'
$MAKEINDEX -s confproc2.ist $TEXFILE.idx

echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** PdfLaTeX: add toc (4/6) ***'
$PDFLATEX  $TEXFILE.tex

echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** PdfLaTeX: create toc + include index (5/6) ***'
$PDFLATEX $TEXFILE.tex

#--- class settings: "final" option and binding
cat ${class_paperback_final}.tex ${INTEXFILE}.tex  >${TEXFILE}.tex

echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** PdfLaTeX: mod. class insertion, for proper PDF links for full papers (6/6) ***'
$PDFLATEX $TEXFILE.tex

```

Once the PDF file is generated, is ts saved: and

```
mkdir ${ELECTRONICFOLDER}/papers/
#--- save PDF
echo "cmd: cp ${TEXFILE}.pdf ${GPATH}/${ELECTRONICFOLDER}/${TEXFILE}.pdf"
cp ${TEXFILE}.pdf $ELECTRONICFOLDER/${TEXFILE}.pdf

```

We can then generate individual PDFs with proper PDF metadata:

```
#=== EXPORT individual pdf papers back from the proceedings + hdr/footers/metadata
cd ${GPATH}
echo; echo; echo; echo; echo; echo; echo $separator; echo $separator;
echo '*** Export individual PDFs ***'
echo "cmd: ./exportIndividualPDFs.sh ${GPATH} ${TEXFILEPATH}/${TEXFILE} ${INPATH} ${SPPATH} ${PI
./exportIndividualPDFs.sh ${GPATH} ${TEXFILE} ${INPATH} ${SPPATH} ${PDFPATH} ${PDFTKPATH}

```

and clean up the temporary folder if uncommented:

```
# rm -r ${GPATH}/tmp/
⟨/buildprocelpb⟩
```

7.7 `countnbpages.sh`: Count the number of pages of individual PDFs

Name `countnbpages.sh`

Type Unix/bash script

Purpose counts each `.pdf`'s number of pages, and stores them in a `proceedings.npc` file.

It is basically used to check that there is no discrepancy between the number of pages indicated in the proceedings file (`proceedings.tex` or `expapersswitch.tex`) and the actual number of pages of each paper.

```
%    \changes{0.7}{2010/08/05}{added \file{countnbpages.sh}}
⟨*countnbpages⟩
#!/bin/sh

```

We first set the path to LaTeX binaries and names of LaTeX compiler:

```
#-- set path to LaTeX binaries
LATEXPATH="/usr/texbin/" # TeXLive
#-- set names of LaTeX and related compilers
PDFLATEX=$LATEXPATH"pdflatex"

```

We then set the document name and papers' folder name:

```
TEXFILE="simple_proceedings"  #- set document's name
PDFSPATH="papers" #- set the papers' folder name

```

We remove previous `.npt` file:

```
rm -f ${TEXFILE}.npt # count pages from terminal
cd ${PDFSPATH}
```

For each `.pdf` file, we count its number of pages using the `pdfinfo`[25] script:

```
for file in *.pdf
do
  pdfinfo -meta $file | grep "Pages:" > tmp0
  echo "file $file has `sed 's/.*\([0-9]\).*/\1/' < tmp0` pages" >> ../${TEXFILE}.npt
done
rm -f tmp0
cd ..
more ${TEXFILE}.npt
⟨/countnbpages⟩
```

papers=countpages

Alternative Non-Unix afficionados can use the papers=countpages option (results are stored in `proceedings.npc`); this is however much much much slower, as `pdfpages` has to insert each and every page for each and every paper.

7.8 `exportIndividualPDFs.sh`: Export individual PDF papers from the proceedings

Name `exportIndividualPDFs.sh`

Type Unix/bash script

Purpose exports each individual paper from the whole proceedings, provided that a `proceedings.pdftk` file exists

Dependencies `paperssplitpreamble.sh` and `papersinfo.sh`

Note It is only meant to be used once a valid PDF version of the electronic proceedings exist (ie. once all formatting issues have been solved and final content have been added), before creating a CD-ROM.

We first set macros from the script arguments (path, TEX file name without extension, paper info folder, split PDFs folder, PDF proceedings folder):

```
% \changes{0.7}{2010/08/05}{added \file{exportIndividualPDFs.sh}}
⟨*exportIndividualPDFs⟩
#!/bin/bash

args=("$@")
GPATH=${args[0]}  #= ~/e-proceedings
TEXFILE=${args[1]}  #= proceedings
INPATH=${args[2]}  #= papers_info
#mkdir -p $INPATH
SPPATH=${args[3]}  #= papers_split
#mkdir -p $SPPATH
PDFPATH=${args[4]}
PDFTKPATH=${args[5]}

PDFFILE=${TEXFILE}.pdf  # for use in the paper_split.sh and paper_info.sh scripts

echo "-PATH (working path):       $GPATH"
echo "-TeX file (orig. TeX proc): $TEXFILE"
echo "-PDF:                       $PDFFILE (original PDF proc)"
echo "-PDFPATH (indiv. PDFs):     $PDFPATH "
echo "-PDFTKPATH (pdftk info):    $PDFTKPATH"
echo "-INPATH (papers info):      $INPATH"
echo "-SPPATH (split papers):     $SPPATH"

```

we move to the PDFtk folder:

```
cd $PDFTKPATH
list=‘ls *.pdftk‘
```

Then for each file:

```
for tmpfile in $list
do
```

[25] `pdfinfo` can be downloaded from

we concatenate all lines by removing carriage returns:

```
  cp ${tmpfile} test.txt
  #-- 2-concat all lines, removing carriage returns
  sed -e :a -e '$!N;s/\n/LineBreak/;ta' -e 'P;D' test.txt >test2.txt
  perl -ne ' s/LineBreakInfoKey/\nInfoKey/g; print ' test2.txt >test3.txt
  perl -ne ' s/LineBreakInfoValue/\nInfoValue/g; print ' test3.txt >test4.txt
  perl -ne ' s/LineBreak//g; print ' test4.txt >test5.txt
  mv test5.txt $tmpfile
done

```

After a quick clean up:

```
rm -f tmp*
rm -f test*.txt

```

we generate the `paper_split.sh` bash script that will soon be used:

```
cd $GPATH
echo "__________"
echo "__ split PDFs: generate bash script file"
pwd
echo "cmd: cat paperssplitpreamble.sh $TEXFILE.pdftk >tmp.sh"
cat paperssplitpreamble.sh $TEXFILE.pdftk >tmp.sh
mv tmp.sh ${GPATH}/papers_split.sh

```

to which we add the Perl command that helps to adds echo command to each `pdftk`[26] command in `paper_split.sh`:

```
echo "__________"
echo "__ split PDFs: Perl to add echo lines to 'papers_split.sh' script"

#echo "cmd: Perl to copy/add 'echo' cmd to each pdftk command, in 'papers_split.sh'"
perl -p -e 's/^pdftk(.*[\n\r])/echo \"pdftk $1\"\npdftk $1/gm' ${GPATH}/papers_split.sh >tmp.tx
mv tmp.txt ${GPATH}/papers_split_all.sh

```

We can now launch the `papers_split_all.sh` bash script:

```
echo; echo "__________"
echo "__ split PDFs: launch bash script file"
#echo "cmd: chmod +x papers_split_all.sh"
chmod +x papers_split_all.sh

echo "cmd: ./papers_split_all.sh"
#echo "    ./papers_split_all.sh ${GPATH} ${TEXFILE} ${INPATH} ${SPPATH} ${PDFPATH}"
./papers_split_all.sh ${GPATH} ${TEXFILE} ${INPATH} ${SPPATH} ${PDFPATH}
# rm ${SPPATH}/*.ps #useful only if 'pdf2ps -> ps2pdf', not useful with 'gs'


```

and once all individual PDFs have been split, we generate new individual PDFs with corrected and homogeneous metadata:

```
#--- generate PDF with corrected metadata
echo "__________"
echo "__ Correct PDF metadata with papersinfo.sh"
./papersinfo.sh ${GPATH} ${TEXFILE} ${INPATH} ${SPPATH} ${PDFPATH} ${PDFTKPATH}

```

[26] Get `pdftk` at: `http://www.accesspdf.com/pdftk/`

we're now good for a real spring cleaning:

```
##--- clean
#rm -r ${INPATH}
#rm -r ${SPPATH}
#rm papers_split.sh
#rm -r tmp
```
⟨/exportIndividualPDFs⟩

7.9 `removeLaTeXcmds.sh`: Convert LaTeX strings for PDF metadata

Name `removeLaTeXcmds.sh`

Type Unix/bash script that makes use of Perl commands

Purpose converts LaTeX strings for PDF metadata

Note to be used when using the metadata in `proceedings.pdftk` (generated by the pdftk option), as we need to remove all LaTeX commands and accents that are not recognized by the PDF format.

We first get the script arguments (path, input file name, output file name):

⟨*removeLaTeXcmds⟩
```
#!/bin/bash

# arg 0: path, arg 1: input file; arg 2: output file

#-- save arguments for use
args=("$@")
path=${args[0]}
file=${args[1]}
outputfile=${args[2]}
```

We go to the folder where we want to work, and copy the input file in order to work on a copy:

```
cd ${path}
cp ${file} tmp.txt
#echo "__ ORIGINAL: $file ___"
#cat tmp.txt

```

We then remove accents with Perl commands::

```
#echo "  "
#echo "__ removed accents: __"
perl -p -i -e " s/\\'e/e/g " tmp.txt
perl -p -i -e " s/\\'{e}/e/g " tmp.txt
perl -p -i -e " s/\\`e/e/g " tmp.txt
perl -p -i -e ' s/\\"e/e/g ' tmp.txt
perl -p -i -e " s/\\`{e}/e/g " tmp.txt
perl -p -i -e " s/\\'a/a/g " tmp.txt
perl -p -i -e " s/\\`a/a/g " tmp.txt
perl -p -i -e " s/\\`{a}/a/g " tmp.txt
perl -p -i -e ' s/\\"{o}/oe/g ' tmp.txt
perl -p -i -e ' s/\\"o/o/g ' tmp.txt
perl -p -i -e ' s/\\o{}/o/g ' tmp.txt
perl -p -i -e ' s/\\^o/o/g ' tmp.txt
perl -p -i -e " s/\\'o/o/g " tmp.txt
perl -p -i -e " s/\\`o/o/g " tmp.txt
perl -p -i -e " s/\\'u/u/g " tmp.txt
```

```
perl -p -i -e ' s/\\u //g ' tmp.txt
perl -p -i -e ' s/\\"u/u/g ' tmp.txt
perl -p -i -e ' s/\\i /i/g ' tmp.txt
perl -p -i -e ' s/\\i/i/g ' tmp.txt
perl -p -i -e " s/\\\'{i}/i/g " tmp.txt
perl -p -i -e ' s/\\"{i}/i/g ' tmp.txt
perl -p -i -e ' s/\\c {c}/c/g ' tmp.txt

```

We also remove LaTeX text formatting commands (such as `\textit`, `\textbf`,, etc), simple math symbols ($\sim$, μ for instance; this list is to be customized depending on the proceedings data) and remove backquotes (\):

```
#echo "  "
#echo "__ removed textit, texbf, {, }: __"
perl -p -i -e ' s/\--/-/g ' tmp.txt
perl -p -i -e " s/\\ss/ss/g " tmp.txt
perl -p -i -e ' s/\\textsuperscript //g ' tmp.txt
perl -p -i -e " s/\\&/&/g " tmp.txt
perl -p -i -e ' s/\\mu/mu\:/g ' tmp.txt
perl -p -i -e ' s/\\sim\s//g ' tmp.txt
perl -p -i -e ' s/\\sim//g ' tmp.txt
perl -p -i -e ' s/\s:/:/g ' tmp.txt

perl -p -i -e ' s/\$//g ' tmp.txt
perl -p -i -e " s/textit //g " tmp.txt
perl -p -i -e " s/textbf //g " tmp.txt
```

We finally remove other left-overs LaTeX text formatting commands only at the end (such as {, }, etc), and accents reminders (', `, ", etc):

```
perl -p -i -e " s/\{//g " tmp.txt
perl -p -i -e " s/\}//g " tmp.txt
perl -p -i -e ' s/\`\`/"/g ' tmp.txt
perl -p -i -e " s/\'\'/\"/g " tmp.txt

#echo "  "
#echo "__ removed \: ___"
perl -pi -e 's/\\//g' tmp.txt
perl -pi -e 's/\s{2,10}\{\}/ /g' tmp.txt
perl -pi -e 's/\s{2,10}/ /g' tmp.txt

cp tmp.txt $outputfile
```
⟨/removeLaTeXcmds⟩

7.10 `paperssplitpreamble.sh`: Preamble of `papersplit.sh`

Name `paperssplitpreamble.sh`

Type Unix/bash script preamble

Purpose preamble for the `papersplit.sh` script, that is generated by running `exportIndividualPDFs.sh`

Note it is not meant to be directly used by the user.

We set macros from the script arguments (path, TeX file name without extension, paper info folder, split PDFs folder, PDF proceedings folder):

```
⟨*paperssplitpreamble⟩
#!/bin/bash
```

```
1321
1322 args=("$@")
1323 GPATH=${args[0]}
1324 TEXFILE=${args[1]} # example1
1325 INPATH=${args[2]} # papers_info
1326 SPPATH=${args[3]}  #papers_split
1327 PDFPATH=${args[4]}
1328
1329 cd ${GPATH}
1330 SPPATH=${GPATH}/${SPPATH}
1331 PDFFILE=${GPATH}/${TEXFILE}.pdf # PDF proceedings
1332 echo "PDF proc used for individual PDFs extraction:\n  --> $PDFFILE"
1333 echo "saving tmp .ps and .pdf files into\n  --> $SPPATH"
1334 ⟨/papersplitpreamble⟩
```

7.11 `papersinfo.sh`: Generate individual PDFs with proper metadata

Name `papersinfo.sh`

Type Unix/bash script

Purpose adds proper metadata to each paper

Dependencies `removeLaTeXcmds.sh` and working `pdftk`

Note to be used once individual papers have been exported from the whole PDF proceedings.

```
1335 ⟨*papersinfo⟩
1336 #!/bin/bash
1337
```

We first get the script arguments, which are in the order:

```
1338 args=("$@")
```

1. the folder where individual PDFs are stored:

```
1339 GPATH=${args[0]} #= ~/proceedings/e-proceedings
```

2. the finished proceedings (PDF file, without extension):

```
1340 TEXFILE=${args[1]} #= ICMC2009_proceedings
```

3. the folder where `pdftk` information files are stored (same as in `buildprocelpb.sh`):

```
1341 INPATH=${args[2]} #= papers_info
```

4. the folder (same as in `buildprocelpb.sh`) where are stored individual `pdf` files (obtained from splitting the proceedings):

```
1342 SPPATH=${args[3]} #= papers_split
```

5. the folder where final individual `pdf` files with proper PDF metadata will be stored:

```
1343 PDFPATH=${args[4]} #= ~/proceedings/e-proceedings
```

6. the folder where individual `.pdftk` files were generated by LaTeX (ex: `pdtk_info/`):

```
1344 PDFTKPATH=${args[5]} #= ~/pdftk_info
1345
```

Let's go! We build the proceedings file name, and list individual PDF files:

```
PDFFILE=${TEXFILE}.pdf     # for use in the paper_split.sh and paper_info.sh scripts

cd ${GPATH}/${SPPATH}
filelist=`ls *.pdf`
mkdir ${PDFPATH}

```

Then, for each PDF file:

```
cd ${GPATH}
chmod +x removeLaTeXcmds.sh

for file in $filelist
do
  base=${file%%.*}
```

- remove LaTeX commands (and accents) from the `pdftk` information files generated by the LaTeX run with the pdftk=true option:

```
  echo "removing LaTeX accents: ${base}.pdftk -> ${base}_clean.info"
#  echo "cmd: removeLaTeXcmds.sh ${GPATH} ${PDFTKPATH}/${base}.pdftk ${INPATH}/${base}_clean.in
  ${GPATH}/removeLaTeXcmds.sh ${GPATH} ${PDFTKPATH}/${base}.pdftk ${INPATH}/${base}_clean.info
```

- use `pdftk`[27] to update each `.pdf` file's info:

```
  echo "adding PDF metadata:    ${base}_clean.info -> ${base}.pdf"
#  echo "cmd: pdftk ${SPPATH}/${base}.pdf update_info ${INPATH}/${base}_clean.info output ${PDF
  echo "pdftk ${GPATH}/${SPPATH}/${base}.pdf update_info ${GPATH}/${INPATH}/${base}_clean.info
  pdftk ${GPATH}/${SPPATH}/${base}.pdf update_info ${GPATH}/${INPATH}/${base}_clean.info output
done
```

⟨/papersinfo⟩

7.12 `buildpapers.sh`: Re-compile all papers

Name `buildpapers.sh`

Type Unix/bash script

Purpose run LaTeX on each paper

Note Useful if you need to make modifications to all papers; for instance to force each individual paper to have the same first page number as the one it has in the proceedings (for papers with page numbers included in the footer).

```
⟨*buildpapers⟩
#!/bin/sh

# Compile all papers with 'pdflatex' of 'latex'
#   (depending if they are in 'sources_pdftex' or 'sources_tex')
# and copy resulting pdf files in the 'papers' folder.
# Expected tree structure:
#     proceedings/papers/sources_pdftex/
#     proceedings/papers/sources_tex/
# with this script in 'proceedings/'

#--- choose if you compile from scratch or only once
```

[27] Get `pdftk` at: `http://www.accesspdf.com/pdftk/`

```
#BUILD_TYPE=final      #recompile and re-do biblio
BUILD_TYPE=renumber   #recompile only once for re-numbering

#--- set system dependent variables
LATEXPATH="/usr/texbin/" # TeXLive

#--- paths
LATEX=$LATEXPATH"latex"
DVIPDF=/usr/local/bin/dvipdf
PDFLATEX=$LATEXPATH"pdflatex"
BIBTEX=$LATEXPATH"bibtex"
MAKEINDEX=$LATEXPATH"makeindex"
PROCSTY='dafx_06.sty'

#--- Compiling .tex files with pdfLaTeX
cd papers/sources_pdftex
for i in *; do
  echo; echo; echo '=====> Compiling' $i '.tex  with pdfLaTeX <====='
  cd $i
  # copy the paper style (in case you changed it)
  cp ../../$PROCSTY .
  echo; echo '   ---> 1st compilation of ' $i '.tex'
  $PDFLATEX $i
  if [ $BUILD_TYPE = final ]; then
    echo; echo '   ---> Compiling the bibliography ' $i '.tex'
    $BIBTEX $i
    echo; echo '   --- 2nd compilation of ' $i '.tex'
    $PDFLATEX $i
    echo; echo '   ---> 3rd compilation of ' $i '.tex'
    $PDFLATEX $i
  fi
  #--- copy the pdf where the proceedings will be assembled
  cp $i.pdf ../..
  cd ..
done
#--- Compiling .tex files with LaTeX (problems related with hyperref)
cd ../sources_tex
for i in *; do
  echo; echo; echo '=====> Compiling' $i '.tex  with LaTeX <====='
  cd $i
  #--- copy the paper proceedings style (if you changed the tree)
  cp ../../$PROCSTY .
  echo; echo '   ---> 1st compilation of ' $i '.tex '
  $LATEX $i.tex
  if [ $BUILD_TYPE = final ]; then
    echo; echo '   ---> Compiling the bibliography ' $i '.tex '
    $BIBTEX $i
    echo; echo '   ---> 2nd compilation of ' $i '.tex '
    $LATEX $i
    echo; echo '   ---> 3rd compilation of ' $i '.tex '
    $LATEX $i
  fi
  #--- produce the pdf from dvi
  $DVIPDF $i.dvi $i.pdf
  #--- copy the pdf where the proceedings will be assembled
  cp $i.pdf ../..
```

```
  cd ..
done
```
⟨/buildpapers⟩

7.13 `buildcppdfpapers.sh`: Copy all PDFs papers at the right place

Name `buildcppdfpapers.sh`

Type Unix/bash script

Purpose copies all PDF files at the right place (*i.e.* in 'papers/')

Note the previous Unix script already does it, but you may want to only copy the files, not re-run LaTeX them.

⟨∗buildcppdfpapers⟩
```
#!/bin/sh

cd papers/sources_tex
for i in *; do
  echo '*********' $i '*********'
  cp $i/$i.pdf ..
done
cd ../sources_pdftex
for i in *; do
  echo '*********' $i '*********'
  cp $i/$i.pdf ..
done
```
⟨/buildcppdfpapers⟩

Conclusion and copyright

It now seems that you have all the necessary files, scripts and information with a series of working and complete examples, in order to produce your v own conference proceedings! Have fun using `confproc`!!!

8 Implementation

Please note: The macros containing a '@' are internal commands. They do *not* belong to the user interface and should not be called directly by the end user! You may get unpredictable results if you don't know what you are doing. Internal macros may be changed by me without announcement or warning, so be careful. Use them at your own risk if you cannot resist...

8.1 Key-value option management

New [v0.7]

We use the following two packages to manage the options with the key=value format [12, 13]:

```
⟨*package⟩
\RequirePackage{kvoptions}
\RequirePackage{kvoptions-patch}
```

as well as the `xifthen` package for simpler if/then tests (related to the option management):

```
\RequirePackage{xifthen}
```

I know that next step is useless (it corresponds to the default values provided by `keyval` if no value is given)... but let's explicitly set the `keyval`-key to be used:

```
\SetupKeyvalOptions{family=confproc,prefix=confproc@}
```

and the `\confproc` command for managing key values:

```
\newcommand*{\confproc}[1]{\setkeys{confproc}{#1}}
```

8.2 Options declaration

8.2.1 Obsolete options

We then define pre-version 0.5 options that became obsolete (their name has changed or their functionality has disappeared):

```
\DeclareStringOption{compil}{\PackageWarning{confproc}{Option %
  "compil=bib*" ignored since v0.5; use "bib=*" instead of "compil=bib*"}}
\DeclareVoidOption{draft}{\PackageWarning{confproc}{Option "draft" %
  ignored since v0.5; use "papers=draft" instead}}
\DeclareVoidOption{final}{\PackageWarning{confproc}{Option "final" %
  ignored since v0.5; use "papers=final" instead}}
\DeclareVoidOption{tocnumleft}{\PackageWarning{confproc}{Option %
  "tocnumleft" ignored since v0.5; use "tocnum=left" instead}}
\DeclareVoidOption{tocnumright}{\PackageWarning{confproc}{Option %
  "tocnumright" ignored since v0.5; use "tocnum=right" instead}}
\DeclareVoidOption{cleardoublepage}{\PackageWarning{confproc}{Option %
   "cleardoublepage" ignored since v0.5; use "twosidepapers" instead}}
\DeclareVoidOption{clearsinglepage}{\PackageWarning{confproc}{Option %
  "clearsinglepage" ignored since v0.5; use "onesidepapers" instead}}
```

8.2.2 Options of the `book` package

The usual font sizes and paper size options are used to set the document parameters. Right now, only three font sizes are supported[28], namely 10pt, 11pt, and 12pt):

10pt
11pt
12pt

```
\DeclareVoidOption{10pt}{%
    \expandafter\PassOptionsToPackage
    \expandafter{\CurrentOption}{book}}
\DeclareVoidOption{11pt}{%
```

[28] It still wonder how bigger/smaller font sizes could be useful for proceedings; so if you need it, please ask!

```
    \expandafter\PassOptionsToPackage
    \expandafter{\CurrentOption}{book}}
\DeclareVoidOption{12pt}{%
    \expandafter\PassOptionsToPackage
    \expandafter{\CurrentOption}{book}}
```

letterpaper For paper size, we define only the two usual ones for proceedings, namely letterpaper:

```
\newif\if@proc@letterpaper
\DeclareVoidOption{letterpaper}{%
  \@proc@letterpapertrue
  \setlength\paperheight {11in}%
  \setlength\paperwidth {8.5in}%
  \setlength\oddsidemargin {-4.95truemm}%
  \setlength\evensidemargin {-4.95truemm}%
  \def\shiftsletterpaper{}%
  \PassOptionsToClass{\CurrentOption}{book}%
  \PassOptionsToPackage{\CurrentOption}{hyperref}}
```

a4paper and a4paper:

```
\DeclareVoidOption{a4paper}{%
  \@proc@letterpaperfalse
  \setlength\paperheight {297mm}%
  \setlength\paperwidth {210mm}%
  \setlength\oddsidemargin {-4.95truemm}%
  \setlength\evensidemargin {-10.95truemm}%
  \def\shiftsafourpaper{}%
  \PassOptionsToClass{\CurrentOption}{book}%
  \PassOptionsToPackage{\CurrentOption}{hyperref}}
```

They are used to set the document, and then passed to the `book` and `hyperref` packages. Both

oneside twoside oneside and twoside options are re-defined as they were in the `book` package, and then passed to it:

```
\DeclareVoidOption{oneside}{\@twosidefalse \@mparswitchfalse%
    \PassOptionsToPackage{\CurrentOption}{book}}
\DeclareVoidOption{twoside}{\@twosidetrue \@mparswitchtrue%
    \PassOptionsToPackage{\CurrentOption}{book}}
```

8.2.3 Proceedings-specific formatting options

twosidepapers onesidepapers We define a pair of complementary options (twosidepapers | onesidepapers) that allow for a double/single page clear after each paper:

```
\DeclareBoolOption[true]{twosidepapers}
\DeclareComplementaryOption{onesidepapers}{twosidepapers}
```

papers The papers option is a string defaulted to final:

```
\DeclareStringOption[final]{papers}
```

that can have three values: empty (very fast run, with faked insertion of user-selected pages by `confproc`, and without page existence check), draft (not too slow run, with page existence check and faked insertion of user-selected pages by `pdfpages`), final (slow run with real insertion of user-selected pages by `pdfpages`) and countpages (slow run, with real insertion of all pages by `pdfpages`).

electronic The electronic is a boolean defaulted to true:

```
\DeclareBoolOption[true]{electronic}
```

(that generates an electronic document with color hyperlinks)

printed and printed is declared as its complementary:

```
\DeclareComplementaryOption{printed}{electronic}
```

(and generates black color links).

binding New [v0.7] The binding is a string defaulted to 0mm:

```
\DeclareStringOption[0mm]{binding}
```

it in facts defines a length used for the inner binding.

headers The headers is a string defaulted to allpages:

```
\DeclareStringOption[allpages]{headers}
```

It can take four values: headers=no to add no headers on any page, headers=pdfonly to add headers on PDF papers only, headers=exceptpdf to add headers on all pages except the inserted PDFs, and headers=allpages to add headers on all pages.

bib New [v0.7] The bib is a string defaulted to none:

```
\DeclareStringOption[none]{bib}
```

that indicates the general bibliography status (four values): bib=none, bib=merge (merging process, only 1st and last page are inserted), bib=backref (generating proper back-references: citations still appear in green in the last page's header) and bib=final (removing last page's citation, but also breaking back-references in next LaTeX runs).

8.2.4 List of inserted papers

paperselec New [v0.7] The paperselec is a string defaulted to all:

```
\DeclareStringOption[all]{paperselec}
```

8.2.5 Lists formatting

twocoltoc The twocoltoc is a boolean defaulted to false:

```
\DeclareBoolOption[false]{twocoltoc}
```

that allows to produce a 2-columns table of contents.

onecoltoc The onecoltoc is its complementary:

```
\DeclareComplementaryOption{onecoltoc}{twocoltoc}
```

used to produce a usual 1-column table of contents.

tocnum The tocnum is a string defaulted to left:

```
\DeclareStringOption[left]{tocnum}
```

that allows to place TOC numbers either on the left or the right side.

twocolbib The twocolbib is a boolean defaulted to true:

```
\DeclareBoolOption[true]{twocolbib}
```

that produces either a tow- or a one-column(s) general bibliography.

onecolbib The onecolbib is its complementary:

```
\DeclareComplementaryOption{onecolbib}{twocolbib}
```

twocolindex The twocolindex is a boolean defaulted to true:

```
\DeclareBoolOption[true]{twocolindex}
```

that provides a 2-columns index of authors.

threecolindex The threecolindex is its complementary:

```
\DeclareComplementaryOption{threecolindex}{twocolindex}
```

and therefore provides a 3-columns index of authors.

8.2.6 Help for checking data and layout

`checktitle` The checktitle is a boolean defaulted to false:

```
\DeclareBoolOption[false]{checktitle}
```

`checkauthor` The checkauthor is a string defaulted to false:

```
\DeclareBoolOption[false]{checkauthor}
```

`colorheaders` The colorheaders is a string defaulted to black:

```
\DeclareStringOption[black]{colorheaders}
```

`showmarginlines` The showmarginlines is a boolean defaulted to false:

```
\DeclareBoolOption[false]{showmarginlines}
```

`showpapernumber` The showpapernumber is a boolean defaulted to false:

```
\DeclareBoolOption[false]{showpapernumber}
```

`movepagenumber` The movepagenumber is a boolean defaulted to false:

```
\DeclareBoolOption[false]{movepagenumber}
```

that is used to move the footer in order to check page numbers, when the PDF papers already have a page number before insertion.

8.2.7 Verbose and `pdftk` options

`debug` The debug is a boolean defaulted to false:

```
\DeclareBoolOption[false]{debug}
```

`verbose` The verbose is a boolean defaulted to false:

```
\DeclareBoolOption[false]{verbose}
```

`pdftk` The pdftk is a boolean defaulted to false:

```
\DeclareBoolOption[false]{pdftk}
```

When used, it allows to generate bash commands (in a file with `.pdftk` extension) that will call `pdftk`[29] to set the `.pdf` file's author, subject and producer/creator, and save thse data in a temporary folder for future use

`pdftksubject` The pdftkfolder is a string that indicates the (temporary) folder into which individual `.pdftk` files are writtne. Note that the folder has to exist prior to run LaTeX, otherwise error will be displayed:

```
\DeclareStringOption[.]{pdftkfolder}
```

`pdftksubject` The pdftksubject is a string that indicates the PDF subject to `pdftk`:

```
\DeclareStringOption[Conference]{pdftksubject}
```

`pdftkproducer` The pdftkproducer is a string that indicates the PDF creator (ie. the program generating the new PDF) to `pdftk`:

```
\DeclareStringOption[pdftk 1.12 + Ghostscript 8.71]{pdftkproducer}
```

`pdftkcreator` The pdftkcreator is a string that indicates the PDF creator (ie. the program that initially created the PDF) to `pdftk`:

```
\DeclareStringOption[LaTeX2e + confproc 0.7]{pdftkcreator}
```

[29] Get `pdftk` at: `http://www.accesspdf.com/pdftk/`

8.2.8 Options passed to `hyperref` and `geometry`

hyperref The hyperref is a string:

```
\DeclareStringOption{hyperref}[]
\DeclareStringOption{geometry}[]
\DeclareStringOption{afterhyperref}[]
\DeclareStringOption{beforehyperref}[]
```

8.2.9 Unknown options: passed to the `hyperref` package

In its very first version (0.1i), the `confproc` package was passing the following `hyperrref`-specific options to it: colorlinks, colorlinks and colorlinks=true, colorlinks=false, linkcolor, citecolor, urlcolor, pagecolor, bookmarksopen, bookmarksopen=true, bookmarksopen=false. Not knowing how to use the `keyval` package, I used a simple and dirty trick, re-defining and passing these options, but it was limitating the customization of `hyperref` to what I believed was useful. To remove this biais, versions 0.2 to 0.5 treated them as any unknown options, that were passed to the `hyperref` package. While this behavior is still true, *it will be discontinued in the next versions, so please refrain to using it.*

New [v0.7]

We print a warning message for unknown options, and pass them to `hyperref`:

```
\DeclareDefaultOption{\PackageWarning{confproc}{Unknown %
  option `\CurrentOption'; passed to 'hyperref'}%
  \PassOptionsToClass{\CurrentOption}{hyperref}}
```

8.3 Options processing

8.3.1 Default values for options

When not set by the user, options have the following default settings:

```
\ExecuteOptions{letterpaper,10pt,twoside,twosidepapers,%
  electronic,binding=0mm,papers=final,paperselec=all,headers=allpages,%
  onecoltoc,tocnum=left,threecolindex,twocolbib,bib=none,%
  checktitle=false,checkauthor=false,showmarginlines=false,%
  showpapernumber=false,movepagenumber=false,colorheaders=black,%
  verbose=false,debug=false,pdftk=false,%
%  beforehyperref={},afterhyperref={},%
  hyperref={colorlinks=true,linkcolor=red,citecolor=blue,urlcolor=blue,%
    bookmarksopen=true,bookmarksopenlevel=1}%
  geometry={text={6.9in,9in},%
    inner=0.8in,top=1in,bottom=1in,%
    headsep=7.05mm,footskip=10mm,voffset=-5mm}}
```

8.3.2 Options processing

Options can now be processed thatnks to `kvoptions`:

```
\ProcessKeyvalOptions*
```

8.4 Application of option values

8.4.1 Package information messages and option settings

The electronic version only makes use of visible hyper-references (clickable links):

```
\ifconfproc@electronic%
  \PassOptionsToPackage{colorlinks=true}{hyperref}%
  \PackageInfo{confproc}{use color links with hyperref}%
\else
```

```
  \PassOptionsToPackage{colorlinks=false}{hyperref}%
  \PackageInfo{confproc}{does not use color links with hyperref}%
\fi
```

The binding option manages the binding length for both the `geometry` package and paper insertion with `pdfpages`:

```
\newlength{\proc@binding}
\ifthenelse%
  {\equal{\confproc@binding}{}}
  {\setlength{\proc@binding}{0mm}
    \PackageInfo{confproc}{setting binding to default (0mm)}}
  {\setlength{\proc@binding}{\confproc@binding}
    \PackageInfo{confproc}{setting binding to \confproc@binding}}
```

With papers=, papers are inserted either normally from page 1 to a user-defined value:

```
\newif\if@proc@IncludePDFs
```

replaced from page 1 to a user-defined value:

```
\newif\if@proc@ReplacePDFs
```

or fully inserted (discarding the user-defined paper number of pages):

```
\newif\if@proc@IncludeFullPDFs
```

We define a new file into which the number of pages will be counted:

```
\newwrite\npagesfile
```

For papers=empty, papers are replaced by a `confproc`-specific layout:

```
\ifthenelse%
  {\equal{\confproc@papers}{empty}}%
  {\PackageInfo{confproc}{replacing PDF files by information pages}%
    \@proc@ReplacePDFstrue
    \@proc@IncludeFullPDFsfalse
    \@proc@IncludePDFsfalse
    \immediate\openout\npagesfile=\jobname.nps}
  {\ifthenelse%
```

For papers=draft, papers are replaced by a `pdfpages`-specific layout (also called draft):

```
    {\equal{\confproc@papers}{draft}}%
    {\PackageInfo{confproc}{not including PDF files with 'pdfpages'}%
      \PassOptionsToPackage{draft}{pdfpages}%
      \@proc@ReplacePDFsfalse
      \@proc@IncludeFullPDFsfalse
      \@proc@IncludePDFsfalse
      \immediate\openout\npagesfile=\jobname.nps}
```

For papers=final, papers are normally inserted from page 1 to the user-defined number of pages:

```
    {\ifthenelse%
      {\equal{\confproc@papers}{final}}
      {\PackageInfo{confproc}{including PDF files with 'pdfpages'}%
        \PassOptionsToPackage{final}{pdfpages}%
        \@proc@ReplacePDFsfalse
        \@proc@IncludeFullPDFsfalse
        \@proc@IncludePDFstrue
        \immediate\openout\npagesfile=\jobname.nps}
```

For papers=countpages, papers are fully inserted, discarding the user-defined number of pages:

```
      {\ifthenelse%
        {\equal{\confproc@papers}{countpages}}
```

```
        {\PackageInfo{confproc}{counting each paper's number of %
          pages by including all its pages}
%          \PassOptionsToPackage{draft}{pdfpages}
          \@proc@ReplacePDFsfalse
          \@proc@IncludeFullPDFstrue
          \@proc@IncludePDFsfalse
          \immediate\openout\npagesfile=\jobname.npc}
```

In case of unknown value for papers=, the default is to insert papers using papers=final:

```
        {\PackageWarning{confproc}{unknown option %
          `papers=\confproc@papers' ; using `papers=final'}%
          \@proc@ReplacePDFsfalse{}
          \@proc@IncludeFullPDFsfalse
          \@proc@IncludePDFstrue
          \immediate\openout\npagesfile=\jobname.nps}
      }
    }
  }

```

To manage fancy headers, we need two new ifs so as to process differently pages with papers and pages without papers.

```
\newif\if@proc@FancyHeadersOnPapers
\newif\if@proc@FancyHeadersExceptPapers
```

For headers=none, no headers/footers are added:

```
\ifthenelse%
  {\equal{\confproc@headers}{none}}
  {\PackageInfo{confproc}{no headers}%
  \@proc@FancyHeadersOnPapersfalse
  \@proc@FancyHeadersExceptPapersfalse}
```

For headers=pdfonly, headers/footers are only added to inserted PDFs:

```
  {\ifthenelse%
    {\equal{\confproc@headers}{pdfonly}}
    {\PackageInfo{confproc}{headers on inserted PDFs only}%
      \@proc@FancyHeadersOnPaperstrue
      \@proc@FancyHeadersExceptPapersfalse}
```

For headers=exceptpd, headers/footers are added to all pages but inserted PDFs:

```
    {\ifthenelse%
    {\equal{\confproc@headers}{exceptpdf}}
      {\PackageInfo{confproc}{headers for all pages except PDFs}%
        \@proc@FancyHeadersOnPapersfalse
        \@proc@FancyHeadersExceptPaperstrue}
```

For headers=allpages, headers/footers are added to all pages including inserted PDF:

```
      {\ifthenelse%
      {\equal{\confproc@headers}{allpages}}
        {\PackageInfo{confproc}{headers on all pages, PDFs included}%
          \@proc@FancyHeadersOnPaperstrue
          \@proc@FancyHeadersExceptPaperstrue}
        {\PackageWarning{confproc}{unknown %
          'headers=\confproc@headers' option (using 'headers=allpages')}}
       }
    }
  }

```

To manage the general bibliography, we need two new ifs: one for the existence of the general bibliography and one for the appearance of the bibliographic items at the last page of each paper's header.

```
\newif\if@proc@BibNone
\newif\if@proc@BibRemoveCiteHdr
```

For bib=merge, each paper is replaced by its first and last page only, where the bibliography is displayed:

```
\ifthenelse%
  {\equal{\confproc@bib}{merge}}%
  {\PackageInfo{confproc}{bib: display 1st+last page of each paper}%
    \@proc@BibNonefalse%
    \def\conf@BibMerge{}%
    \@proc@BibRemoveCiteHdrfalse}%
```

For bib=backref, each paper is inserted as defined by the user, and back-references appear on the last page (upper left, in green):

```
  {\ifthenelse%
    {\equal{\confproc@bib}{backref}}%
    {\PackageInfo{confproc}{bib: displays back references}%
      \@proc@BibNonefalse%
      \def\conf@BibBackRef{}%
      \@proc@BibRemoveCiteHdrfalse}%
```

For bib=final, each paper is inserted as defined by the user, and back-references do not more appear (which means they are broken on successive LaTeX runs):

```
    {\ifthenelse%
      {\equal{\confproc@bib}{final}}%
      {\PackageInfo{confproc}{bib: hide bib items using 'nocite'}%
        \@proc@BibNonefalse%
        \@proc@BibRemoveCiteHdrtrue}%
```

For bib=none, no bibliography is displayed at the end of the proceedings:

```
      {\PackageInfo{confproc}{bibliography: none}%
         \@proc@BibNonetrue}%
  }%
}
%\confproc@bib\@empty
%  \typeout{confproc/bibliography: no setting for "bib="! 5
%    Please use one of: none, backref, merge of final.}%
```

\proccite
New [v0.7]

We define the \confcite citation function that can be changed depending on the citation function used by the chosen bibliography style:

```
\if@proc@BibRemoveCiteHdr
  \newcommand{\confcite}[1]{\nocite{#1}}
  \PackageInfo{confproc}{removing citations (with nocite{*}): %
    next run should be the last (will loose hyperlinks)}
\else \newcommand{\confcite}[1]{\cite{#1}}
\fi
```

To manage the left or right numbergin of the table of contents, we need a new if:

```
\newif\if@proc@TocNumberingRight
\ifthenelse%
  {\equal{\confproc@tocnum}{left}}%
  {\PackageInfo{confproc}{TOC numbering on left}%
    \@proc@TocNumberingRightfalse}
  {\ifthenelse%
```

```
    {\equal{\confproc@tocnum}{right}}%
    {\PackageInfo{confproc}{TOC numbering on right}%
      \@proc@TocNumberingRighttrue}
    {\PackageWarning{confproc}{unknown tocnum=\confproc@tocnum %
      (using 'right' instead)}%
      \@proc@TocNumberingRighttrue}
  }

```

Then come various options for checking the document layout and content. When willing to check spelling errors in the papers' author list:

```
\ifconfproc@checkauthor
  \PackageInfo{confproc}{add 'author list' field on the paper's 1st page}%
\else \PackageInfo{confproc}{do not add author list}%
\fi
```

and in the paper titles:

```
\ifconfproc@checktitle
  \PackageInfo{confproc}{add title field on the paper's 1st page}%
\else \PackageInfo{confproc}{do not add title}%
\fi
```

We also show margin lines:

```
\ifconfproc@showmarginlines
  \PackageInfo{confproc}{show margin lines to check template-complience}%
\else \PackageInfo{confproc}{do not show margin lines}%
\fi
```

add the paper number (useful when not using the papers=empty option):

```
\ifconfproc@showpapernumber
  \PackageInfo{confproc}{add paper number below page number}%
\else \PackageInfo{confproc}{do not add paper number below page number}%
\fi

```

With the twosidepapers option, papers always start on the right side (like book chapters):

```
\ifconfproc@twosidepapers
  \PackageInfo{confproc}{papers opening on right (odd) side}
\else
  \PackageInfo{confproc}{papers opening on any side}
\fi

```

With the verbose option, more information about paper insertion is provided to the user:

```
\newif\if@proc@verbose
\@proc@verbosefalse
\ifconfproc@verbose
  \@proc@verbosetrue
  \PackageInfo{confproc}{verbose mode turned on}
\else \PackageInfo{confproc}{verbose mode turned off}
\fi
```

With the debug option, even more information is provided to the class developer:

```
\ifconfproc@debug
  \@proc@verbosefalse
  \PackageInfo{confproc}{verbose mode turned off and debug turned on}
  \PassOptionsToPackage{debug}{hyperref}
\else \PackageInfo{confproc}{debug mode turned on}
\fi
```

8.4.2 Package information messages and option settings

We pass options to specific packages, possibly overwriting previous settings:

```
\PassOptionsToPackage{\confproc@hyperref}{hyperref}
\PassOptionsToPackage{\confproc@geometry}{geometry}
```

8.4.3 Print option values (hard debug mode)

New [v0.7]

With the verbose=true option, all option values and defaults are printed in the `.log` file and window:

```
\ifconfproc@verbose
  \typeout{_________________}
  \if@proc@letterpaper
    \typeout{| | Document formatting:}
    \typeout{| | ____ letterpaper}
  \else
    \typeout{| | Document formatting:}
    \typeout{| | ____ a4paper}
  \fi
  \iffalse\@twoside \typeout{| | ____ twoside=false (=oneside)}
  \else \typeout{| | ____ twoside=true}
  \fi
  \ifconfproc@twosidepapers \typeout{| | ____ twosidepapers=true}
  \else \typeout{| | twosidepapers=false (=onesidepaper)}
  \fi
  \typeout{| | Proceedings-specific formatting:}
  \ifconfproc@electronic \typeout{| | ____ electronic=true (file version)}
  \else \typeout{| | ____ electronic=false (printed)}
  \fi
  \typeout{| | ____ binding=\confproc@binding  (for printed version)}
  \typeout{| | ____ papers=\confproc@papers  (paper insertion)}
  \typeout{| | ____ headers=\confproc@headers  (header add to pages)}

  \typeout{| | List of papers:}
  \typeout{| | ____  paperselec=\confproc@paperselec}

  \typeout{| | Lists (toc, bib, index):}
  \ifconfproc@twocoltoc \typeout{| | ____ twocoltoc=true}
  \else \typeout{| | ____ twocoltoc=false (=onecoltoc)}
  \fi
  \ifthenelse{\equal{\confproc@tocnum}{left}}%
    {\typeout{| | ____ tocnum=left}}
    {\typeout{| | ____ tocnum=right}}
  \ifconfproc@twocolbib \typeout{| | ____ twocolbib=true}
  \else \typeout{| | ____ twocolbib=false (=onecolbib)}
  \fi
  \typeout{| | ____ bib=\confproc@bib}
  \ifconfproc@twocolindex \typeout{| | ____ twocolindex=true}
  \else \typeout{| | ____ twocolindex=false (=threecolindex)}
  \fi

  \typeout{| | Help for layout design:}
  \ifconfproc@checkauthor
    \typeout{| | ____ checkauthor=true (add author list to 1st page)}
  \else
    \typeout{| | ____ checkauthor=false (do not add author list to 1st page)}
  \fi
```

```
\ifconfproc@checktitle
  \typeout{| | ____ checktitle=true (add title to 1st page)}
\else
  \typeout{| | ____ checktitle=false (do not add title to 1st page)}
\fi

\ifconfproc@showpapernumber
  \typeout{| | ____ showpapernumber=true (add paper number)}
\else
  \typeout{| | ____ showpapernumber=false (do not add paper number)}
\fi
\ifconfproc@movepagenumber
  \typeout{| | ____ movepagenumber=true (move paper number for checking)}
\else
  \typeout{| | ____ movepagenumber=false (do not move paper number)}
\fi
\ifconfproc@showmarginlines
  \typeout{| | ____ showmarginlines=true (add template format)}
\else
  \typeout{| | ____ showmarginlines=false (do not add template format)}
\fi
\typeout{| | ____ colorheaders=\confproc@colorheaders (color for header/footer)}

\typeout{| | Verbose:}
\ifconfproc@debug \typeout{| | ____ debug=true (for hyperref)}
\else \typeout{| | ____ debug=false (for hyperref)}
\fi
\ifconfproc@verbose \typeout{| | ____ verbose=true (for confproc+hyperref)}
\else \typeout{| | ____ verbose=false (for confproc+hyperref)}
\fi
\ifconfproc@pdftk
  \typeout{| | ____ pdftk=true (for use with pdftk to add PDF metadata)}
  \typeout{| | ____ pdftkfolder=\confproc@pdftkfolder (folder where .pdftk files are saved)}
  \typeout{| | ____ pdftksubject=\confproc@pdftksubject (subject for individual PDF metadata)}
  \typeout{| | ____ pdftkproducer=\confproc@pdftkproducer (producer for individual PDF metadata)}
  \typeout{| | ____ pdftkcreator=\confproc@pdftkcreator (creator for individual PDF metadata)}
\else \typeout{| | ____ pdftk=false (for use with pdftk to add PDF metadata)}
\fi
\typeout{| | passed to hyperref: \confproc@hyperref}
\typeout{| | passed to geometry: \confproc@geometry}
\typeout{_________________}
\typeout{   }
\fi
```

With the pdftk=true option, we also generate a general .pdftk file containing ann pdftk commands:

```
\ifconfproc@pdftk
  \newwrite\pdftkinfoall
  \immediate\openout\pdftkinfoall=\jobname.pdftk
  \newwrite\pdftkinfofile
\fi
```

8.5 Initialization

As you can see, this package is based on the book package for all its layout aspects.

```
\LoadClass[10pt,letterpaper]{book}
```

8.6 Required packages

Several packages are included, among which many are required.
Use `graphicx` to insert logos (first page, welcome letters):

```
\RequirePackage{graphicx}
```

Use `pdfpages` (core of this class) to insert individual papers as PDF documents, page-by-page:

```
\RequirePackage{pdfpages}
```

Use `fancyhdr` to customize the headers and footers (for instance so that they match those of the paper templates):

```
\RequirePackage{fancyhdr}
```

Use `tocbibind` to change the `\indexname` command; its options disable the automatic insertion in the table of contents (hand made insertion instead):

```
\RequirePackage[nottoc,notbib,notindex]{tocbibind}
```

Use `titletoc` (part of the `titelsec` package) to change the table of contents layout (order of text, numbers, fonts, etc.):

```
\RequirePackage[rightlabels]{titletoc}
```

Use `multitoc` with the toc option for a two columns table of contents:

```
\ifconfproc@twocoltoc
  \RequirePackage[toc]{multitoc}
\fi
```

Use the `index` package to enable the creation of the index of authors:

```
\RequirePackage{index}
```

Use the `multitoc` package for a multi-columns table of contents or index:

```
\RequirePackage{multicol}
```

`\theindex` Also, when asking for a 2 or 3 columns index, redefine the `\theindex` environment (modified from the `gatech-thesis-index.sty` package) as:

```
\ifconfproc@twocolindex
  \renewenvironment{theindex}{%
    \if@twocolumn \@restonecolfalse
    \else \@restonecoltrue \fi
    \vspace*{-0.8cm}
    \section*{{\indexname}}
    \let\item\@idxitem
    \columnseprule \z@
    \columnsep 35\p@
    \begin{multicols}{2}[%
      \ifx\index@prologue\@empty\else
        \index@prologue
        \bigskip
      \fi]%
      \parindent\z@
      \parskip\z@ \@plus .3\p@\relax
  }{\end{multicols}%
    \if@restonecol \onecolumn
    \else \clearpage \fi}
\else
  \renewenvironment{theindex}{%
    \if@twocolumn \@restonecolfalse
    \else \@restonecoltrue \fi
    \vspace*{-0.8cm}
    \section*{{\indexname}}
```

```
    \let\item\@idxitem
    \columnseprule \z@
    \columnsep 35\p@
    \begin{multicols}{3}[%
      \ifx\index@prologue\@empty\else
        \index@prologue
        \bigskip
      \fi]%
      \parindent\z@
      \parskip\z@ \@plus .3\p@\relax
  }{\end{multicols}%
    \if@restonecol \onecolumn
    \else \clearpage \fi }
\fi
```

Use the `sectsy` package to change the sections font in the table of contents:

```
\RequirePackage{sectsty}
```

Use the `newapave` style for the general bibliography:

```
\RequirePackage{newapave}
```

If you do not wish to use the one developed for DAFx-06 but prefer to use the original `newapa` style, replace this last line in `confproc.cls` by:

```
\RequirePackage{newapa}
```

Links in the PDF files require to use the `color` package:

```
\RequirePackage{color}
```

We predefine here the names and values for the color links, so that they can be used:

```
\definecolor{colorforlink}{rgb}{0,0,0.5}
\definecolor{colorforpage}{rgb}{0,0,0.5}
\definecolor{colorforcite}{rgb}{0,0.5,0}
\definecolor{colorforurl}{cmyk}{0,1,0,0}
```

together with the `hyperref` package with the following default options:

```
%\confproc@beforehyperref{}   %^^A TODO: not functioning yet
\RequirePackage[pdftex,raiselinks,hyperindex,backref,pagebackref,%
    plainpages=false,pdfpagelabels,breaklinks,linktocpage,%
    pdfstartview=XYZ]{hyperref}
%\confproc@afterhyperref{}   %^^A TODO: not functioning yet
```

and with the `bookmark` package:

```
%%\RequirePackage[figure,table]{hypcap}
\RequirePackage{bookmark}
```

8.7 Proceedings specific commands

We now define the default values of some proceedings-specific commands.

8.7.1 PDF metadata

`\procpdfauthor` Define commands to set the PDF metadata: `\procpdfauthor` for the author:

```
\newcommand{\procpdfauthor}{[Proceedings author/editor]}
```

`\procpdftitle` `\procpdftitle` for the title:

```
\newcommand{\procpdftitle}{[Proceedings title]}
```

\procpdfsubject and \procpdfsubject for the subject:

```
\newcommand{\procpdfsubject}{[Proceedings short title] %
  ([Proceedings Acronym]), [City], [Country], [Dates]}
```

\hypersetup These commands are used in the \hypersetup command that is evaluated only when the document begins (so that you can redefine its author, title and subject):

```
\AtBeginDocument{
  \hypersetup{
    pdfauthor = \procpdfauthor,
    pdftitle = \procpdftitle,
    pdfsubject = \procpdfsubject,
    pdfkeywords = {},
    pdfcreator = {LaTeX + confproc v0.7},
    pdfproducer = {pdfLaTeX}}}
```

8.7.2 Page layout with geometry

The proceedings default page layout is defined thanks to geometry:

```
\iffalse\@twoside
  \usepackage[bindingoffset=\proc@binding]{geometry}
\else%
  \usepackage[twoside,bindingoffset=\proc@binding]{geometry}
\fi
```

Those values may be changed in the preamble, depending on your paper template.

8.7.3 Special section names

\contentsname We redefine the names of the table of contents (as it should appear in itself):

```
\renewcommand{\contentsname}{Conference Program}
```

\bibname the general bibliography as it appears in the document and in the table of contents:

```
\renewcommand{\bibname}{Full Bibliography}
```

\indexname and the index of authors as it appears in the document and in the table of contents:

```
\renewcommand{\indexname}{Index of Authors}
```

8.7.4 Header and footer

\proclhead We first define the default left header:

```
\newcommand{\proclhead}{}
```

\procchead the default central header:

```
\newcommand{\procchead}{{\color{red}Proceedings of the... \hfill %
  01--29 February, 2001}}
```

\procrhead and the default right header:

```
\newcommand{\procrhead}{}
```

\proclfoot Similarly, we define the default left footer (empty):

```
\newcommand{\proclfoot}{}
```

\proccfoot the default central footer:

```
\newcommand{\proccfoot}{{\small \color{red} Proc-\thepage}}
```

\procrfoot and the default right footer (empty):

```
\newcommand{\procrfoot}{}
```

We now define the default page styles for use with headers:

```
\pagestyle{fancyplain}
```

\headrulewidth together with the corresponding rule width for the headers:

```
\renewcommand{\headrulewidth}{0pt}
```

\footrulewidth and for the footers:

```
\renewcommand{\footrulewidth}{-5mm}
```

\procfootvskip We also define the vertical skip length to be applied to the footer:

```
\newlength{\procfootvskip}
\setlength{\procfootvskip}{0cm}
```

\procoptfootvskip and the option vertical skip length added to the footer with the movepagenumber option:

```
\newlength{\procoptfootvskip}
\ifconfproc@movepagenumber \setlength{\procoptfootvskip}{3mm}%
\else \setlength{\procoptfootvskip}{0mm} \fi
```

\lhead The left header is given as:

```
\lhead{\color{\confproc@colorheaders}\proclhead}
```

\chead the central header is:

```
\chead{\color{\confproc@colorheaders}\procchead}
```

\rhead and the right header is:

```
\rhead{\color{\confproc@colorheaders}\procrhead}
```

\lfoot The left footer is also set empty:

```
\lfoot{}
```

\rfoot as well as the right footer:

```
\rfoot{}
```

\proccfoot The center footer is the page number with option and mandatory vertical spaces:

```
\cfoot{\color{\confproc@colorheaders}\vskip\procfootvskip%
  \vskip\procoptfootvskip\proccfoot}%
```

\pagestyle Depending on the value of the headers option, we change the default page style:

```
\ifdefined \conf@FancyHeadersExceptPapers
  \pagestyle{fancy}
\else
  \pagestyle{empty}
\fi
```

8.7.5 Table of contents layouts

Using the `titletoc` commands, we define the various table of contents layout.

8.7.6 Default

For right numbering:

```
\if@proc@TocNumberingRight
```

we first set the left margin of papers inserted as sections:

```
  \titlecontents{section}[0em]% left margin
```

we then set the table of contents spacing between 2 papers:

```
    {\vspace*{0.5mm}}% space between two papers in the TOC
```

and the filler and page number:

```
    {}%
    {}%
      {\hfill \hspace*{-2.5em}\makebox[0pt][r]{\contentspage}\hspace*{2.5em}}% filler and page
      [\addvspace{0.5mm}]% space after
```

For left numbering:

```
\else%
%  \dottedcontents{section}[]{\fillright}{}{1pc}
  \titlecontents{section}[2.5em]%
    {\vspace*{0.5mm}}%
```

we set the left shift of page numbers:

```
    {\hspace*{-2.5em}\makebox[0pt][r]{\contentspage}\hspace*{2.5em}}% left shifting page num.
    {\hspace*{-2.5em}\makebox[0pt][r]{\contentspage}\hspace*{2.5em}}% idem
    {}% filler and page
    [\addvspace{0.5mm}]% space after
%  \titlecontents{subsection}[2.5em]%
%    {\vspace*{0.3em}}%
%    {}% left shifting page num.
%    {}% idem
%    {}% filler and page
%  \titlecontents{subsection*}[2.5em]%
%    {\vspace*{0.3em}}%
%    {}% left shifting page num.
%    {}% idem
%    {}% filler and page
\fi
```

`\tocmattertocstyle` **At document frontmatter with right numbering**

```
\if@proc@TocNumberingRight
  \newcommand{\frontmattertocstyle}{
```

Parts are used for the preamble:

```
    \titlecontents{part}[0em]%
      {\addvspace{3mm}}%
      {\Large\bfseries}%
      {\Large\bfseries}%
      {}%
      [\addvspace{0.5mm}]
```

and chapters for each page for the preamble:

```
    \titlecontents{chapter}[0em]%
      {\addvspace{2mm}}%
      {\large\bfseries\itshape}%
      {\large\bfseries\itshape}%
      {}%
```

```
      [\addvspace{0.5mm}]
  }
```

At document frontmatter with left numbering

```
\else
  \newcommand{\frontmattertocstyle}{
```

Parts are used for the preamble:

```
    \titlecontents{part}[0em]%
      {\addvspace{3mm}}%
      {\Large\bfseries}%
      {\Large\bfseries}%
      {}%
      [\addvspace{0.5mm}]
```

and chapters for each page for the preamble:

```
    \titlecontents{chapter}[0em]%
      {\addvspace{2mm}}%
      {\large\bfseries\itshape}%
      {\large\bfseries\itshape}%
      {}%
      [\addvspace{0.5mm}]
  }
\fi
```

At document mainmatte

\mainmattertocstyle

Sections are always used for papers. Chapters are used as sessions when days are used. Parts are used as days, or when sessions of no days are used. The corresponding TOC style or the main matter is then defined for right page numbers in TOC:

```
\if@proc@TocNumberingRight
  \newcommand{\mainmattertocstyle}{
    \titlecontents{part}[0pt]%
      {\addvspace{3mm}}%
      {\Large\bfseries}%
      {\Large\bfseries}%
      {}%
      [\addvspace{0.5mm}]
    \titlecontents{chapter}[0pt]%
      {\addvspace{2mm}}%
      {\large\bfseries\itshape}%
      {\large\bfseries\itshape}%
      {}%
      [\addvspace{0.5mm}]
  }
```

and for left page numbers in TOC:

```
\else % left TOC page numbers
  \newcommand{\mainmattertocstyle}{
    \titlecontents{part}[0pt]%
      {\addvspace{3mm}}%
      {\Large\bfseries}%
      {\Large\bfseries}%
      {}%
      [\addvspace{0.5mm}]
    \titlecontents{chapter}[0pt]%
      {\addvspace{2mm}}%
      {\large\bfseries\itshape}%
```

```
      {\large\bfseries\itshape}%
      {}%
      [\addvspace{0.5mm}]
  }
\fi
```

`npagespreamble` Before redefining the main matter, we define a counter that is used to count the number of pages in the preamble (especially for the `pdftk`[30] output data:

```
\newcounter{npagespreamble}
```

`\mainmatter` New [v0.7] Hence, we redefine the `\mainmatter` command that does not use anymore this style (it is left to the user to decide weither he wants to use it or not) but indicated the number of pages in the preamble:

```
\renewcommand\mainmatter{%
  \PackageInfo{confproc}{counted \arabic{npagespreamble} pages in the preamble}
  \cleardoublepage
  \@mainmattertrue
  \pagenumbering{arabic}}
```

8.7.7 At document backmatter

`\backmattertocstyle` Sections are used to format/display the general bibliography and index of authors, but they appear as parts in the table of contents, for both right page numbers:

```
\if@proc@TocNumberingRight
  \newcommand{\backmattertocstyle}{
    \titlecontents{part}%
      [0pt]%
      {\addvspace{3mm}}%
      {\Large\bfseries}%
      {\Large\bfseries}%
      {\hfill \hspace*{-2.5em}\contentspage\hspace*{2.5em}}%
      [\addvspace{0.5mm}]
    \titlecontents{chapter}%
      [0pt]%
      {\addvspace{2mm}}%
      {\large\bfseries\itshape}%
      {\large\bfseries\itshape}%
      {\hfill \hspace*{-2.5em}\contentspage\hspace*{2.5em}}%
      [\addvspace{0.5mm}]
  }%
```

and left page numbers:

```
\else
  \newcommand{\backmattertocstyle}{%
    \titlecontents{part}%
      [0pt]%
      {\addvspace{3mm}}%
      {\makebox[0pt][r]{\contentspage}\hspace*{2.5em}\Large\bfseries}%
      {\makebox[0pt][r]{\contentspage}\hspace*{2.5em}\Large\bfseries}%
      {}%
      [\addvspace{0.5mm}]
    \titlecontents{chapter}%
      [0pt]%
      {\addvspace{2mm}}%
      {\makebox[0pt][r]{\contentspage}\hspace*{2.5em}\large\itshape\bfseries}%
```

[30] Get `pdftk` at: `http://www.accesspdf.com/pdftk/`

```
      {\makebox[0pt][r]{\contentspage}\hspace*{2.5em}\large\itshape\bfseries}%
      {}%
      [\addvspace{0.5mm}]
  }%
\fi
```

\backmatter We then redefine the \backmatter command and ensure the footer is properly modified:

```
\renewcommand\backmatter{%
  \if@openright \cleardoublepage
  \else \clearpage \fi
  \@mainmatterfalse
  \cfoot{\color{\confproc@colorheaders}\vskip \procfootvskip %
    \vskip \procoptfootvskip \proccfoot}}
```

8.7.8 Headers/footers

\otherpagestyle The default page style (and corresponding headers and footers) is set for non PDF-inserted pages:

```
\newcommand{\otherpagestyle}{
  \if@proc@FancyHeadersExceptPapers\pagestyle{fancy}
  \else \pagestyle{empty} \fi}
```

\otherpagestyle and for a particular page:

```
\newcommand{\thisotherpagestyle}{
  \if@proc@FancyHeadersExceptPapers\thispagestyle{fancy}
  \else \thispagestyle{empty} \fi}
```

\PDFpagestyle as well as for PDF-inserted pages:

```
\newcommand{\PDFpagestyle}{
  \if@proc@FancyHeadersOnPapers\thispagestyle{fancy}
  \else\thispagestyle{empty} \fi}
```

\chapterfont Using the sectsty package, all chapters have the same font in the table of contents:

```
\chapterfont{\thisotherpagestyle}
```

\clearsingleordoublepage We define the \clearsingleordoublepage command depending if on the document format (one-side or two-side):

```
\newcommand{\clearsingleordoublepage}{
  \iffalse\@twoside \clearpage
  \else \cleardoublepage \fi}
```

8.7.9 *X* and *Y* shifts

\LaTeXxShift \LaTeXyShift \WordxShift \WordyShift We now define the *X* and *Y* shifts for LaTeX (\LaTeXxShift and \LaTeXyShift) and Word (-\WordxShift, \WordyShift) generated papers as lengths:

```
\newlength{\LaTeXxShift} \setlength{\LaTeXxShift}{0pt}
\newlength{\LaTeXyShift} \setlength{\LaTeXyShift}{0pt}
\newlength{\WordxShift} \setlength{\WordxShift}{0pt}
\newlength{\WordyShift} \setlength{\WordyShift}{0pt}
```

8.7.10 Paper insertion commands

\conf@paper@title We now define (as empty) the commands used to add TOC, bookmark and bib data to inserted PDF papers, *i.e.* the paper title:

```
\newcommand{\conf@paper@title}{}
```

\conf@paper@authors the paper authors:

```
\newcommand{\conf@paper@authors}{}
```

\conf@paper@index the commands for insertion in the index:

```
\newcommand{\conf@paper@index}{}
```

\conf@paper@ref the paper reference, *i.e.* a tag (*e.g.* the file name, or the submission number):

```
\newcommand{\conf@paper@ref}{}
```

\conf@paper@pagenum the number of pages:

```
\newcommand{\conf@paper@pagenum}{}
```

\conf@paper@cite the bibliographic references (for the general bibliography):

```
\newcommand{\conf@paper@cite}{}
```

\papertitlestyle the style for the title:

```
\newcommand{\papertitlestyle}{}
```

\paperauthorstyle and finally the style for both the list of authors and the text between the title and the list of authors:

```
\newcommand{\paperauthorstyle}{\texorpdfstring{\newline\itshape}{\break}}
```

npages A new counter npages is added, for the number of pages of a paper:

```
\newcounter{npages}
```

\proctoctitleauthor The \proctoctitleauthor command defines the style for title/author entry in the table of contents using the style \papertitlestyle for the paper with title \papertitle and the style \paperauthorstyle for the paper with authors \paperauthors :

```
\newcommand{\proctoctitleauthor}[2]{%
```

We chose to insert both the paper title and the list of authors in the table of contents,

```
    \texorpdfstring{{\papertitlestyle #1}{\paperauthorstyle #2}}%
```

whereas only the title is inserted as a section in the bookmark.

```
      {{\papertitlestyle #1}}}
```

Then, each author will be inserted as a subsection in the \procinsertpaper command.

8.7.11 Table of contents insertion

\tableofcontents We redefine the usual \tableofcontents command that switches to the corresponding section style for insertion in the table of contents:

```
\renewcommand\tableofcontents{%
```

clears a single or double page according to the one/two-sided page format:

```
  \clearsingleordoublepage
```

adds the conference program name to the PDF bookmark:

```
  \pdfbookmark[0]{\contentsname}{contents}
```

switches to one-column if needed:

```
  \if@twocolumn \@restonecoltrue\onecolumn
  \else \@restonecolfalse \fi
```

inserts the table of contents name as a starred section:

```
  \section*{\contentsname}%
```

inserts the table of contents itself:

```
  \@starttoc{toc}%
```

restores the two-column mode if any:

```
\if@restonecol\twocolumn\fi
```

and clears a single or double page according to the one/two-sided page format:

```
\clearsingleordoublepage}
```

8.7.12 Organize the program by days or sessions

\procday The \procday command inserts the day given as argument in the table of contents:

```
\newcommand{\procday}[1]{%
  \phantomsection \addcontentsline{toc}{part}{#1}}
```

\session The \session command adds a session to the table of contents:

```
\newcommand{\session}[1]{%
  \phantomsection \addcontentsline{toc}{chapter}{#1}}
```

8.7.13 Paper switch

\paperswitch The \paperswitch command will be redefined in the expapersswitch.tex file, containing information about all papers. It is therefore declared empty:

```
\newcommand{\paperswitch}{}
```

8.7.14 Modifying the bibliography style

\bibhang We first set the \bibhang length:

```
\setlength{\bibhang}{0.5em} %
```

\thebibliography We then redefine the \thebibliography environment, for proper use and insertion of the new section title in the table of contents:

```
\if@proc@BibNone
  \renewenvironment{thebibliography}[1]{%
    \PackageInfo{confproc}{ignoring #1 biblio file (‘bib=none’ option)}}
\else
  \renewenvironment{thebibliography}[1]{%
    \ifconfproc@twocolbib \twocolumn \fi
    \ifdefined\conf@BibMerge \nocite{*}
    \else \clearsingleordoublepage \fi%
    \section*{\bibname}%
    \addcontentsline{toc}{part}{\bibname}
    \@mkboth{\MakeUppercase\bibname}{\MakeUppercase\bibname}%
    \procbibintro
    \list{\@biblabel{\@arabic\c@enumiv}}%
      {\settowidth\labelwidth{\@biblabel{#1}}%
       \leftmargin\labelwidth
       \advance\leftmargin\labelsep
       \@openbib@code
       \usecounter{enumiv}%
       \let\p@enumiv\@empty
       \renewcommand\theenumiv{\@arabic\c@enumiv}}%
    \sloppy
    \clubpenalty4000
    \@clubpenalty \clubpenalty
    \widowpenalty4000%
    \sfcode‘\.\@m}
  {\def\@noitemerr
    {\@latex@warning{Empty ‘thebibliography’ environment}}%
```

```
    \endlist
    \setlength{\labelsep}{0em}
    \setlength{\itemindent}{-\bibhang}
    \setlength{\leftmargin}{\bibhang}}
\fi
```

\newblock We redefine the \newblock command to reduce the space between bib items:

```
\renewcommand\newblock{\hskip 0em plus 0.0em minus .07em}
```

8.7.15 General bibliography introduction

\procbibintro The \procbibintro cmd defaults the introductory paragraph of the full bibliography:

```
\newcommand{\procbibintro}{{\it ~~~This bibliography is a compilation
  of all bibliographic  references from each paper. Page numbers that
  appear at the  end of each entry link to the bibliography sections that
  include it. Please click on the URL or on the page number to access
  the linked item.}}
```

8.7.16 Index insertion

\insertindex The \insertindex cmd defines the index insertion (it may later be hidden in a proper redefinitin of the \theindex command):

```
\newcommand{\insertindex}{
```

We first clear the page, so that two-side documents start on a right (odd) page:

```
  \clearsingleordoublepage
```

We then back to the 1-column format, in case one adds text before the index:

```
  \onecolumn
```

We then include a phantom section and a link to bookmark (do not remove, as this dirty hack provides a valid pointer to the index):

```
%  \section*{\addcontentsline{toc}{part}{\bibname} \bibname}%
  \section*{~~}%
  \addcontentsline{toc}{part}{\indexname}%
```

The index of authors may have no header/footer, in the case it is a single (and last) page that is printed inside the cover (as we did for the paperback version of the DAFx-06 proceedings):

```
  \renewcommand{\procchead}{}%
  \renewcommand{\proccfoot}{}%
```

We then print the index:

```
  \printindex}
```

and we are done for the index of authors, as well as for the whole confproc class!

8.7.17 Layout design: show the margin lines

\procmarginlines We define a \procmarginlines command that defines the margin layout under the form of a tabular for showmarginlines=true:

```
\ifconfproc@showmarginlines
  \PackageInfo{confproc}{drawing margin lines' command (with a table)}%
  \pagestyle{fancyplain}
  \renewcommand{\headrulewidth}{0.0pt}
%  \renewcommand{\footrulewidth}{0.0pt}
  \newcommand{\procmarginlines}{
    \renewcommand{\footrulewidth}{0.4pt}
    \noindent
```

First adjust the initial vertical space:

```
    \vspace*{7mm} % adjusting vertical initial space
```

A table is the used to draw the vertical lines (blue color):

```
    \begin{table}[h!] % table for vertical lines
      \centering
      \color{blue}
```

The spacing between columns corresponds to the spacing between vertical lines:

```
      \begin{tabular}{|@{}p{3.3in}@{}|@{}p{0.3in}@{}|@{}p{3.3in}@{}|} % spacing between columns
```

The upper horizontal line is added:

```
        \hline % upper horizontal line
```

Add empty lines and a vertical space to fill in the table; this space is less than a page height:

```
        ~~~~~~~~~~~ & ~ &~~~~~~~~~~~~~\\
        \vspace*{7.5in} % less than a page height
        ~~~~~~~~~~~ & ~ &~~~~~~~~~~~~~\\
      \end{tabular}
    \end{table}
  }
```

the same command is otherwise defined as an empty command if showmarginlines=false:

```
\else
  \newcommand{\procmarginlines}{}
  \PackageInfo{confproc}{no margin lines}%
\fi
```

8.7.18 paper insertion

We first define a path to the papers folder:

\PAPERPATH

```
\newcommand{\PAPERPATH}{}
```

\confemptypapercite We then define the \confemptypapercite command to insert fake papers' last page (used by the papers=empty option) with their citations:

```
\newcommand{\confemptypapercite}[2]{%
  \vspace*{0.3\textheight}%
  \begin{flushleft}
    \begin{tabular}{lp{0.7\textwidth}}
    \Large [Title] & \Large \conf@papertitle\\
      & \vspace*{0.5cm}\\
    \Large [Author(s)] & \Large \conf@paperauthor\\
      & \vspace*{0.5cm}\\
    \Large [File name] & \Large \url{\PAPERPATH #2}\\
      & \vspace*{0.5cm}\\
    \if@proc@BibNone
      \Large [Citation(s)] & \Large [disabled by ``bib=none'' option]\\
        & \vspace*{2cm}\\
    \else
      \Large [Citation(s)] & \Large \confcite{\conf@cite}\\
        & \vspace*{2cm}\\
    \fi
      & \textbf{\Huge Page #1}\\
    \end{tabular}
  \end{flushleft}
}
```

conf@npages We now define a counter that contains this paper's last page number:

```
\newcounter{conf@npages}
```

\confemptypaper We then define the \confemptypaper command to insert fake papers (used by the papers=empty option) and that makes use of this counter to point to the citation page if any:

```
\newcommand{\confemptypaper}[2]{%
  \vspace*{0.3\textheight}%
  \begin{flushleft}
    \begin{tabular}{lp{0.7\textwidth}}
    \Large [Title] & \Large \conf@papertitle\\
      & \vspace*{0.5cm}\\
    \Large [Author(s)] & \Large \conf@paperauthor\\
      & \vspace*{0.5cm}\\
    \Large [File name] & \Large \url{\PAPERPATH #2}\\
      & \vspace*{0.5cm}\\
    \if@proc@BibNone
      \Large [Citation(s)] & \Large [disabled by ``bib=none'' option]\\
        & \vspace*{2cm}\\
    \else
      \Large [Citation(s)] & \Large [see page \theconf@npages{} of this paper]\\
        & \vspace*{2cm}\\
    \fi
      & \textbf{\Huge Page #1}\\
    \end{tabular}
  \end{flushleft}
}
```

\procpaper New [v0.7] We now come to the paper insertion \procpaper command, one of the most important command of the whole class. It has been redefined from \procinsertpaper but with key-values options, as suggested by Andreas Matthias. We first define the command parameters:

```
%%%%% begin key-value option management for \procpaper{} command %%%%%
\newlength{\conf@xshift}
\newlength{\conf@yshift}
\newcounter{conf@switch}
\newcounter{conf@firstpage}
\newcounter{conf@lastpage}
\newcommand{\conf@pagecmd}{}
\newcommand{\conf@tmpauthorlist}{}
\newcommand{\conf@tmptitle}{}
```

\confstylecheckauthor We also define the style of the author list when overlayed on the 1st page:

```
\newcommand{\confstylecheckauthor}{}
```

\confstylechecktitle as well as the style of the title when overlayed on the 1 st page:

```
\newcommand{\confstylechecktitle}{}
```

\confstylechecktitle and a temporary number of pages:

```
\newcounter{locnpages}%
```

we can now define the \procpaper command with keyval syntax:

```
\def\procpaper{\@ifnextchar[{\@procpaper}{\@procpaper[]}}
```

Its default values are set only if the current paper has to be inserted:

```
\def\@procpaper[#1]#2{{%
\ifthenelse{\equal{\confproc@paperselec}{all}\or\equal{\confproc@paperselec}{#2}}
  {\setlength{\conf@xshift}{0cm}
```

```
\setlength{\conf@yshift}{0cm}
\setcounter{conf@npages}{1}
\setcounter{conf@switch}{1}
\def\conf@papertitle{Default paper title}
\def\conf@paperauthor{Default paper author list}
\def\conf@index{}
\def\conf@cite{}
\def\conf@bookmark{} %\pdfbookmark[2]{Default paper author 1}{p_XXX.author1}}
```

and key-values are finally set:

```
\setkeys{ppaper}{#1}
```

The horizontal offset is set depending on the binding:

```
\iffalse\@twoside \addtolength{\conf@xshift}{0cm}
\else \addtolength{\conf@xshift}{\proc@binding} \fi

```

We can now insert the PDF paper, depending on the paper and bib options, as well as depending on its number of pages. First in the case paper=empty:

```
\if@proc@ReplacePDFs
  \immediate\write\npagesfile {file #2.pdf has \theconf@npages \space pages}
  \clearsingleordoublepage
  \setcounter{conf@firstpage}{\thenpagespreamble+\thepage}
  \setcounter{conf@lastpage}{\thenpagespreamble+\thepage+%
    \theconf@npages-1}

  \phantomsection
  \addcontentsline{toc}{section}{\proctoctitleauthor{\conf@papertitle}%
    {\conf@paperauthor}}
  \ifnum\theconf@npages=0
    \typeout{confproc: Error, you asked for an empty paper}
    \typeout{confproc: #2.pdf}
  \fi
  \ifnum\theconf@npages=1
    \confemptypapercite{1}{#2.pdf} \conf@bookmark  \conf@index{}%
  \fi
  \ifnum\theconf@npages=2
    \confemptypaper{1}{#2.pdf} \conf@bookmark \conf@index{}%
    \newpage \confemptypapercite{2}{#2.pdf}
  \fi
  \ifnum\theconf@npages>2
    \confemptypaper{1}{#2.pdf} \conf@bookmark \conf@index{}%
    \setcounter{locnpages}{2}
    \ifthenelse{\thelocnpages<\theconf@npages}%
      {\typeout{smaller}}%
      {\typeout{bigger}}
    \whiledo{\value{locnpages}<\value{conf@npages}}{%
      \newpage \confemptypaper{\thelocnpages}{#2.pdf}%
      \addtocounter{locnpages}{1}}
    \newpage \confemptypapercite{\theconf@npages}{#2.pdf}
  \fi
```

we finaly set the counter for the last page number of the current empty paper:

```
  \setcounter{conf@lastpage}{\thenpagespreamble+\thepage}
```

Second, in the case paper=draft | final | countpages:

```
\else
  \setcounter{conf@firstpage}{\thenpagespreamble+\thepage}
  \conf@index{}%
```

```
    \ifconfproc@showpapernumber
      \cfoot{\color{\confproc@colorheaders}\vskip \procfootvskip %
        \vskip \procoptfootvskip \proccfoot\\
        \color{\confproc@colorheaders}[paper \theconf@switch{}]}{}
%    \else
%    % TODO: DO WE DO ANYTHING OTHERWISE?
    \fi
    \ifconfproc@checktitle
      \renewcommand{\conf@tmptitle}{{\color{blue}%
        \confstylechecktitle\conf@papertitle}}
    \else
      \renewcommand{\conf@tmptitle}{}
    \fi
    \ifconfproc@checkauthor
      \renewcommand{\conf@tmpauthorlist}{{\color{blue}%
        \confstylecheckauthor\conf@paperauthor}}
    \else
      \renewcommand{\conf@tmpauthorlist}{}
    \fi
    \renewcommand{\conf@pagecmd}{\conf@tmptitle\\ \conf@tmpauthorlist}

```

In the particular case where paper=countpages, the user-indicated number of pages is ignored:

```
    \if@proc@IncludeFullPDFs    % include all pages in order to count!!!
      \includepdf[noautoscale,offset=\conf@xshift{} \conf@yshift{},pages=1,%
        linktodoc,linkname=\PAPERPATH #2.pdf,%
        addtotoc={1, section, 1, %
          \proctoctitleauthor{\conf@papertitle}{\conf@paperauthor},%
            \theconf@switch},%
        pagecommand = {\conf@pagecmd\procmarginlines %
          \conf@bookmark \PDFpagestyle}%
        ]{\PAPERPATH #2.pdf}%
      \includepdf[noautoscale,offset=\conf@xshift{} \conf@yshift{},pages=2-,%
        linktodoc,linkname=\PAPERPATH #2.pdf,%
        pagecommand = {\procmarginlines \PDFpagestyle}%
        ]{\PAPERPATH #2.pdf}%
```

Otherwise, we include either a 1-page paper:

```
    \else
      \ifnum\theconf@npages=1 % 1-page paper
        \if@proc@verbose
          \typeout{confproc: 1-page long paper}
        \fi
        \if@proc@BibNone
          \includepdf[noautoscale,offset=\conf@xshift{} \conf@yshift{},%
            pages=1,linktodoc,linkname=\PAPERPATH #2.pdf,%
            addtotoc={1, section, 1, %
              \proctoctitleauthor{\conf@papertitle}{\conf@paperauthor},%
                \theconf@switch},%
            pagecommand = {\conf@pagecmd\procmarginlines %
              \conf@bookmark \PDFpagestyle}%
            ]{\PAPERPATH #2.pdf}%
        \else
          \includepdf[noautoscale,offset=\conf@xshift{} \conf@yshift{},%
            pages=1,linktodoc,linkname=\PAPERPATH #2.pdf,%
            addtotoc={1, section, 1,%
              \proctoctitleauthor{\conf@papertitle}{\conf@paperauthor},%
                \theconf@switch},%
```

```
2345            pagecommand = {\conf@pagecmd\procmarginlines %
2346              \conf@bookmark \PDFpagestyle\vspace*{-1cm}\confcite{\conf@cite}}%
2347            ]{\PAPERPATH #2.pdf}%
2348        \fi
2349      \else
```

a 2-pages paper:

```
2350        \ifnum\theconf@npages=2 % 2-pages paper
2351          \if@proc@verbose\typeout{confproc: 2-page long paper}\fi
2352          \if@proc@BibNone
2353            \includepdf[noautoscale,offset=\conf@xshift{} \conf@yshift{},%
2354              pages=1,linktodoc,linkname=\PAPERPATH #2.pdf,%
2355              addtotoc={1, section, 1, %
2356                \proctoctitleauthor{\conf@papertitle}{\conf@paperauthor},%
2357                  \theconf@switch},%
2358              pagecommand = {\conf@pagecmd\procmarginlines \conf@bookmark %
2359                \PDFpagestyle}%
2360              ]{\PAPERPATH #2.pdf}%
2361            \includepdf[noautoscale,offset=\conf@xshift{} \conf@yshift{},%
2362              pages=2,linktodoc,linkname=\PAPERPATH #2.pdf,%
2363              pagecommand = {\procmarginlines \PDFpagestyle}%
2364              ]{\PAPERPATH #2.pdf}%
2365          \else
2366            \includepdf[noautoscale,offset=\conf@xshift{} \conf@yshift{},%
2367              pages=1,linktodoc,linkname=\PAPERPATH #2.pdf,%
2368              addtotoc={1, section, 1, %
2369                \proctoctitleauthor{\conf@papertitle}{\conf@paperauthor},%
2370                  \theconf@switch},%
2371              pagecommand = {\conf@pagecmd\procmarginlines \conf@bookmark %
2372                \PDFpagestyle\vspace*{-1cm}\confcite{\conf@cite}}%
2373              ]{\PAPERPATH #2.pdf}%
2374            \includepdf[noautoscale,offset=\conf@xshift{} \conf@yshift{},%
2375              pages=2,linktodoc,linkname=\PAPERPATH #2.pdf,%
2376              pagecommand = {\procmarginlines %
2377                \PDFpagestyle\vspace*{-2cm}\confcite{\conf@cite}}%
2378              ]{\PAPERPATH #2.pdf}%
2379          \fi
```

or a 3 (and more)-pages paper:

```
2380        \else % 3 pages and more
2381          \includepdf[noautoscale,offset=\conf@xshift{} \conf@yshift{},%
2382            pages=1,%
2383            linktodoc,linkname=\PAPERPATH #2.pdf,%
2384            addtotoc={1, section, 1, %
2385              \proctoctitleauthor{\conf@papertitle}{\conf@paperauthor},%
2386                \theconf@switch},%
2387            pagecommand = {\conf@pagecmd\procmarginlines %
2388              \conf@bookmark \PDFpagestyle}%
2389            ]{\PAPERPATH #2.pdf}%
2390          \ifdefined\conf@BibMerge%
2391            \includepdf[noautoscale,offset=\conf@xshift{} \conf@yshift{},%
2392              pages=\theconf@npages,linktodoc,linkname=\PAPERPATH #2.pdf,%
2393              pagecommand = {\procmarginlines %
2394                \PDFpagestyle\vspace*{-2cm}\confcite{\conf@cite}}%
2395              ]{\PAPERPATH #2.pdf}%
2396            \PDFpagestyle{}%
2397            \if@proc@verbose\typeout{confproc: bibliography insertion only}\fi
2398          \else
```

```
            \addtocounter{conf@npages}{-1}
            \includepdf[noautoscale,offset=\conf@xshift{} \conf@yshift{},%
              pages=2-\theconf@npages,linktodoc,linkname=\PAPERPATH #2.pdf,%
              pagecommand = {\procmarginlines \PDFpagestyle}%
              ]{\PAPERPATH #2.pdf}%
            \PDFpagestyle{}%
            \addtocounter{conf@npages}{1}
            \if@proc@BibNone
              \includepdf[noautoscale,offset=\conf@xshift{} \conf@yshift{},%
                pages=\theconf@npages,linktodoc,linkname=\PAPERPATH #2.pdf,%
                pagecommand = {\procmarginlines \PDFpagestyle}%
                ]{\PAPERPATH #2.pdf}%
            \else
              \includepdf[noautoscale,offset=\conf@xshift{} \conf@yshift{},%
                pages=\theconf@npages,linktodoc,linkname=\PAPERPATH #2.pdf,%
                pagecommand = {\procmarginlines %
                  \PDFpagestyle\vspace*{-2cm}\confcite{\conf@cite}}%
                ]{\PAPERPATH #2.pdf}%
            \fi
          \fi
        \fi
      \fi
      \if@proc@verbose
        \typeout{confproc: partial paper insertion %
          (last page=bib items)}
      \fi
    \fi
```

we compute the number of pages and write it to the file that counts the number of pages:

```
    \setcounter{conf@lastpage}{\thenpagespreamble+\thepage-1}
    \setcounter{conf@npages}{\theconf@lastpage}
    \addtocounter{conf@npages}{- \theconf@firstpage}
    \immediate\write\npagesfile{file #2.pdf has \theconf@npages \space pages}
```

and finally turn the page for next paper or section:

```
    \newpage
    \ifconfproc@twosidepapers \cleardoublepage
    \else \clearpage \fi
  \fi
```

Let us also give some feedback to the user when verbose=true:

```
  \if@proc@verbose
    \typeout{______ debug: insert paper ______}
    \typeout{confproc/file: #2.pdf (\theconf@npages \space pages)}
    \typeout{confproc/title: \conf@papertitle}
    \typeout{confproc/authors: \conf@paperauthor}
    \typeout{confproc/index: \conf@index}
    \typeout{confproc/shift: (\the\conf@xshift, \the\conf@yshift)}
    \typeout{confproc/citations: \conf@cite}
    \typeout{confproc/bookmarks: \conf@bookmark}
    \typeout{confproc/switch ID: \theconf@switch}
    \typeout{_________________________}
  \fi
%   \end{macrocode}
%    We then append the current paper's data to the \package{pdftk}\footnote{Get  \package{pdftk
%    \begin{macrocode}
  \ifconfproc@pdftk
%    \immediate\write\pdftkinfoall {_________________}
```

```
%%-- pdftk version: !!! does not work with PDF v > 1.3
%%    \immediate\write\pdftkinfoall{pdftk A=${PDFFILE} cat A\arabic{conf@firstpage}-\arabic{con
%%-- Ghostscript version: ok with PDF v = 1.4
    \immediate\write\pdftkinfoall{gs -dBATCH -dNOPAUSE -q -sDEVICE=pdfwrite -dFirstPage=\arabic
```

Such text is also written to individual files in the selected folder:

```
%%-- pdftk version: !!! does not work with PDF v > 1.3
%%    \immediate\write\pdftkinfoall{pdftk A=${PDFFILE} cat A\arabic{conf@firstpage}-\arabic{con
%%-- Ghostscript version: ok with PDF v = 1.4
    \immediate\openout\pdftkinfofile=\confproc@pdftkfolder/#2.pdftk
    \immediate\write\pdftkinfofile {InfoKey: Title}
    \immediate\write\pdftkinfofile {InfoValue: \conf@papertitle}
    \immediate\write\pdftkinfofile {InfoKey: Author}
    \immediate\write\pdftkinfofile {InfoValue: \conf@paperauthor}
    \immediate\write\pdftkinfofile {InfoKey: Subject}
    \immediate\write\pdftkinfofile {InfoValue: \confproc@pdftksubject}
    \immediate\write\pdftkinfofile {InfoKey: Producer}
    \immediate\write\pdftkinfofile {InfoValue: \confproc@pdftkproducer}
    \immediate\write\pdftkinfofile {InfoKey: Creator}
    \immediate\write\pdftkinfofile {InfoValue: \confproc@pdftkcreator}
    \immediate\closeout\pdftkinfofile
```

Finally, this text is also written to the log when verbose=true:

```
    \ifconfproc@verbose
      \typeout{_________________}
      \typeout{pdftk A=${PDFFILE} cat A\arabic{conf@firstpage}-\arabic{conf@lastpage} output ${
      \typeout{gs -dBATCH -dNOPAUSE -q -sDEVICE=pdfwrite -dFirstPage=\arabic{conf@firstpage} -d
%     \typeout{pdftk ${SPPATH}/#2.pdf update_info ${INPATH}/#2.info output ${INPATH}/#2.pdf}
      \typeout{InfoName: #2.info}
      \typeout{InfoKey: Title}
      \typeout{InfoValue: \conf@papertitle}
      \typeout{InfoKey: Author}
      \typeout{InfoValue: \conf@paperauthor}
      \typeout{InfoKey: Subject}
      \typeout{InfoValue: \confproc@pdftksubject}
      \typeout{InfoKey: Producer}
      \typeout{InfoValue: \confproc@pdftkproducer}
      \typeout{InfoKey: Creator}
      \typeout{InfoValue: \confproc@pdftkcreator}
      \typeout{InfoEnd}
    \fi
  \fi
  }{}
}}
```

We finally set the key-values using the pre-defined internal commands:

```
\define@key{ppaper}{xshift}{\setlength{\conf@xshift}{#1}}
\define@key{ppaper}{yshift}{\setlength{\conf@yshift}{#1}}
\define@key{ppaper}{npages}{\setcounter{conf@npages}{#1}}
\define@key{ppaper}{switch}{\setcounter{conf@switch}{#1}}
\define@key{ppaper}{title}{\def\conf@papertitle{#1}}
\define@key{ppaper}{author}{\def\conf@paperauthor{#1}}
\define@key{ppaper}{index}{\def\conf@index{#1}}
\define@key{ppaper}{cite}{\def\conf@cite{#1}}
\define@key{ppaper}{bookmark}{\def\conf@bookmark{#1}}
%%%%% end key-value option management for \procpaper{} command %%%%%
```

\procinsertpaper We also redefine its pre-version 0.6 using this latest function, for (partial) backward compatibility:

```
\newcommand{\procinsertpaper}[9]{%
```

It has the following 9 arguments: i) X (=horizontal) and Y (=vertical) shifts (with a space in between), ii) number of pages, iii) a reference, iv) the title, v) the list of authors, vi) the index entries, vii) the citations for the general bibliography, viii) the PDF file name and ix the bookmark entries for the authors. Note that the horizontal and vertical shifts are no more preserved, so please serioulsy consider using the lates \procpaper command.

```
  \PackageWarning{confproc}{!!! '\procinsertpaper' cmd is obsolete (v0.5) %
    and does not preserve PDFs' horizontal and vertical shifts, nor general %
    bib items. Please use the '\procpaper' command instead.}
  \procpaper[title={#4},author={#5},npages=#2,index={#6},cite={#7},%
    bookmark={#9}]{#8}}
```

8.8 Load configuration

Input a local configuration file (`confproc.cfg`), if it exists.

```
\InputIfFileExists{confproc.cfg}
  {\typeout{************^^J%
    * Local config file confproc.cfg used *^^J%
    ************}
  }{}%
⟨/package⟩
```

References

[1] Scott Pakin. How to Package Your LaTeX Package. CTAN: info/dtxtut/. November 2004. Cited page 4

[2] The LaTeX3 Project. LaTeX2e for class and package writers. CTAN: macros/latex/base/cls.dtx. March 1999. Cited page 4

[3] Frank Mittelbach, Denys Duchier, Johannes Braams, Marcin Woliński, and Mark Wooding. The DocStrip program. CTAN: macros/latex/base/docstrip.dtx. March 1999. Cited page 4

[4] The LaTeX3 Project. The LaTeX Project Public License. URL: www.latex-project.org/lppl.txt. September 1999. Cited page 91

[5] Eoin Brazil. Creating Conference Proceedings: Tips And Tricks From The Trenches. UL-IDC-02-03. URL: richie.idc.ul.ie/eoin/research/UL-IDC-02-03.pdf. Interactive Design Centre, University of Limerick, 2002. Cited page 4

[6] SLAC. eConf: Full-Text Proceeding Instructions and Templates. URL: www.slac.stanford.edu/econf/editors/fulltext-template/instructions.html. Retrieved on Sept. 2007. Cited page 5

[7] American Mathematical Society. Guide to AMS Editor's package. URL: www.ams.org/authors/editpkg.html. Retrieved on Sept. 2007. Cited page 5

[8] Adobe systems Inc. Acrobat Professional. URL: www.adobe.com/products/acrobatpro/ Retrieved on Sept. 2007. Cited page 4

[9] Peter Wilson. The `combine` class and the `combinet`, `combnat` and `combcite` packages CTAN: macros/latex/contrib/combine/. March 2004. Cited page 4

[10] Sebastian Rahtz and Heiko Oberdiek. The `hyperref` package. CTAN: macros/latex/contrib/hyperref/. September 2006. Cited page 4, 5, 6, 12, 25, 26

[11] Andreas Matthias. The `pdfpages` package. CTAN: macros/latex/contrib/pdfpages/. 2004. Cited page 4, 5

[12] Heiko Oberdiek. The kvoptions package. CTAN: macros/latex/contrib/oberdiek/kvoptions.pdf. July 2009. Cited page 92

[13] Joseph Wright and Christian Feuersänger. Implementing key–value input: An introduction. 1st draft: www.texdev.net/wp-content/uploads/2009/03/keyval.pdf. Vol. 30, Number 1, pp.110–122, 2009. Cited page 92

[14] Vincent Verfaille. How to make your own proceedings for another conference? URL: www.dafx.ca/dafx06_proceedings_diy.html. October 2006. Cited page 4

[15] Vincent Verfaille. Report on the making of the DAFx-06 proceedings. URL: www.dafx.ca/proceedings/report.pdf. MUMT-SPCL-07-01 report, McGill University, March 2007. Cited page 4

[16] Vincent Verfaille A new package for conference proceedings. The PracTeX Journal, Volume 4. URL: www.tug.org/pracjourn/2007-4/verfaille/verfaille.pdf. November 2007. Cited page 10

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

D

E

F

H

I

L

M

N

O

P

Change History