Search
(courtesy of Google)

THE PracTEX Journal

The online journal of the TeX Users Group
ISSN 1556-6994

About *The PracTeX Journal*

General information
Submit an item
Download style files
Copyright
Contact us
About RSS feeds XML HTML

Archives of *The PracTeX Journal*

Back issues
Author index
Title index
BibTeX bibliography

Next issue
Approx. August 20, 2007

Editorial board

Lance Carnes, editor
Kaveh Bazargan
Kaja Christiansen
Peter Flom
Hans Hagen
Robin Laakso
Tristan Miller
Tim Null
Arthur Ogawa
Steve Peter
Yuri Robbers
Will Robertson

Other key people

More key people wanted

Current Issue

2007, Number 2
[Published 2007-06-10]

Notices

Articles

Columns

Journal home page
General information

Submit an item
Download style files
Copyright
Contact us

THE PracTEX Journal

From the Editor: In this issue; Next issue: Tools for LaTeX and TeX Users; TeX for Editors

Paul Blaga

- Comment on this paper
- Send submission idea to editor

In this issue

The 2007-2 issue is not a very big one (is it because it's the busiest period of the year?). We did manage, however to get some interesting and, hopefully, useful material.

The theme of the issue was "TeX for Editors" and I'm happy to see that all the papers are quite closely related to the topic. Two of the articles are related to the bibliographies in LaTeX. One is about a very useful package (for editors and publishers, mainly), `pdfpages`, and the other describes how an electronic journal, *The PracTeX Journal*, works.

Next issue

Theme: **Tools for LaTeX and TeX Users**
Editor: Francisco Reinaldo

In recent years there has been increasing development of text/table/graphic editing tools for LaTeX and TeX. Consequently, easy-to-use tools that can be accessed by a graphical user interface (GUI) are becoming more popular and many are currently available on the web. Tools such as *JabRef* (Java GUI front end for managing BibTeX and other bibliographies), *LyX* (a GUI word processor that uses LaTeX for printing and supports structural editing of LaTeX markup), and others are emerging as major players in the GUI tools area.

The PracTeX Journal Issue 2007-3 has the theme, "Tools for LaTeX and TeX Users". The intention is to present ideas on the development of, and experiences on the use of, these tools.

The scope of the issue includes, but is not limited to :
- tools that assist the author in preparing graphics, indexes, bibliographies, and other parts of documents;
- previews, and PostScript/PDF manipulation tools;
- free or almost free tools;
- cross-platform tools.

We encourage you to submit original papers describing your experiences using LaTeX and TeX tools, and also papers on tool development work in progress or completed.

--Submission Guidelines:

If you would like to submit an article or technical note for publication please contact the editors pracjourn@tug.org. We will work with you to prepare the article. Also see http://tug.org/pracjourn/submit.html for the Journal's guidelines.

-- Important Dates:
Paper submission deadline: July 15, 2007
Publication date: August 15, 2007

Best Regards,

Francisco Reinaldo
2007-3 Issue Editor

Thanks

Of course, first of all I would like to thank the authors for the time and effort they spent to write such interesting articles.

As for any issue, an essential part of the work was done by the production editors, which, in this particular case, were **Lance Carnes**, **Francisco Reinaldo**, **Yuri Robbers** and **myself**. Being myself a beginner in all the editing business, I couldn't make it without the constant help and encouragements of Lance, Rei and Yuri. Many thanks for that.

Many thanks also to the reviewers and proofreaders who checked the articles and sent comments and corrections.

Special thanks, also, to Dave Walden who keeps entertaining us with his interesting Travels in Tex Land (this time he's visiting the ConTeXt region).

Editorial: Editing Journals and Proceedings with LaTeX

A common misconception is that if you want to edit a journal or some proceedings with TeX you should absolutely have to be at least a TeXnician, if not a TeX wizard. I don't really think this assumption is correct. Definitely, you do need some expertise, but not much more than that. And you must have a methodical approach.

If the journal is already set to be used with LaTeX, and you have all the class and style files you need, your task is not very difficult. If you have to start from scratch, then you will have to take a number of decisions:

> *format of articles:* Sometimes, the paper size is imposed from typographical reasons, however, the exact dimensions of the text, of the header and footer, margins, separations of header and footer are for you to decide. A very useful package to help you do all that is *geometry,* available from CTAN.
> *fonts:* By default, the fonts used by LaTeX are Computer Modern. These are bitmap fonts and do not produce a very good result when printed, especially when some sort of scaling is involved. It is preferable, therefore, to use some Postscript fonts. If the journal doesn't have many mathematical formulae, there are several very good free Postscript fonts you can use (Times, Palatino, Helvetica, ...). Unfortunately, usually, when these fonts are loaded, only the text font of the document is changed, while the mathematical font remains the same, namely Computer Modern and, in most situations, the two don't match very well. There are, however, some compromise solutions, for instance using Times for text and Mathptm for math. There are, also, some commercial mathematical fonts to be used with Times, for instance Mathtime, or the fonts designed to accompany the Mathematica software. Finally, there exist complete sets of fonts for TeX and LaTeX, containing both text and math fonts (such as Lucida Bright).

Once these two important global choices have been made, several other choices have to be made as well:

- the layout of the title page of an article, as well as the layout of section titles;
- the layout of tables and figures;
- the layout of the reference list;
- the contents of the header and footer (you can use, for instance, the package *fancyhdr*, available from CTAN);
- what graphics file formats are acceptable and how to include them.

Once these decisions have been made, is time to prepare the class or style file. First of all, you have to decide which version of TeX is acceptable for the papers submitted. You will, most probably, only accept LaTeX. However, depending on your expertise and available time, you may as well decide to accept other TeX versions, such as Plain TeX or ConTeXt. In any case, LaTeX cannot be excluded. I will assume that the natural choice is to work (at least) with LaTeX.

There are, basically, two roads you can take from here. Either you build your own class file, starting from scratch, or you take a class file which is close to your needs and you modify it. The first way, making your own class file in my opinion, is only for experts; if you don't know the inner mechanism of LaTeX it is quite difficult to get by. I, therefore, strongly recommend the second way, adapting an existing class file.

There are some issues that you will have to deal with before starting to work on the class file. First of all, there is the problem of copyright. It is part of the LaTeX policy to allow the modifications of class and style files, provided they are not distributed under the original name, and that the original source is acknowledged. So, from this point of view, you are on the safe side if you play by the rules. Secondly, it is a good strategy to choose as a starting point a class file that was originally designed for a journal (or several journals), such as *amsart.* The reason is that you will need an entire series of commands that are not available in the standard article *class* , for example, commands for volume, issue, year, etc.

And now comes the hard work. You will have to deal with some characteristics of TeX and LaTeX which are not familiar to the casual user, or even to more advanced users, if they only use LaTeX to write articles or books. There is a chapter in *LaTeX Companion* (by Mittelbach, et al.; see http://www.tug.org/books) explaining the structure of a class file. You may want to read it before you start. Anyway, all you have to do is figure out where in the document is the thing you want to change and how to change it, to adapt it to your needs. An excellent guide on document class internals is the article *Rolling your own Document Class: Using LaTeX to keep away from the Dark Side,* by Peter Flynn, published in the 2006-4 issue of The PracTeX Journal.

The last step you have to perform is to prepare the interaction with your authors. It is preferable, if you have enough time, to prepare a list of exigencies for the submitters. You have to be as specific as possible. In my opinion, at least the following points should be touched:

- the format of the source file: don't say only "LaTeX", but also mention the version and, preferably, forbid the use of the obsolete LaTeX 2.09;
- general style aspects: avoid the use of non-standard package files, avoid redefinition of commands (even shortcuts: the modern word processors have shortcut keys, so there is no need redefine the commands to save key strokes);
- the use of the sectioning commands should be compulsory; in particular, you should absolutely forbid the redefinition of sectioning commands;
- strongly recommend the use of the *theorem* and similar environments; try to provide a *proof* environment and insist that the authors use it;
- forbid the use of vertical spacing commands (\vskip, \bskip, \vfill), as well as commands for line or page breaking;
- when it comes to graphics, as I said, try to be specific: specify exactly what graphic formats are acceptable (depending on the format you are using: PDF or PS); if the authors provide bitmap graphics, be sure they are aware that the resolution should be at least 300dpi to ensure good quality printouts; also, specify what packages should be used to include graphic files; discourage the use of obsolete packages, such as epsbox or others of the same kind;
- don't allow changing the fonts of the document; this is something *you* do, through the class file you defined.

Many journals have article templates. Personally, I don't think they are necessarily a good idea. I would rather prefer that authors use the standard *article* document class, which, definitely, they know better, than to try to use a class which is new to

them. I have always considered that it is a lot easier to make the (usually small) modifications to adapt a standard article to the rules of a new class, than to try to correct the inherent errors committed by those who attempt to use the template.

Paul Blaga
2007-2 Issue Editor

THE PracTEX Journal TEX USERS GROUP

Feedback

From the Readers

- Comment on this paper
- Send submission idea to editor

TpJ

Hi, I'm glad that I discovered your wonderful PracTeX Journal; I've really enjoyed reading a handful of articles from your archives. But I've noticed that some of them seem not to have been run through LaTeX enough times, because some of the references are rendered as "??". The last one I noticed was http://tug.org/pracjourn/2007-1/mertz, where none of the figure references are resolved (not convenient, because this paper is all about figures).

Regards,
Lee Phillips

[Dear Lee, We are glad that our journal is useful to you. Thank you for letting us know about the little problem with the article of Mertz and Slough. We corrected it already, as you can see for yourself. We're sorry about that and please tell us anytime you notice something odd. -- The Editors]

TpJ

An excellent article about how to use PSTricks to make book covers http://tug.org/pracjourn/2007-1/robbers/. Here is something I found while trying the examples in the article, and my LaTeX version (TeXnicCenter) was complaining about the missing package pst-char. Since November 2006 the package pst-char is included in pst-text. I went to the PSTricks homepage, and in the news section there was a comment about this update.

Cheers guys. Keep on.
Nuno Loureiro Ferreira
Portugal

TpJ

Greetings Lapo Filippo Mori and Editors,

In the PracTeX Journal I would appreciate articles on progress in making TeX and LaTeX use Universal code so that owners of the new Mac/Intel computers feel more welcome. I expect that Apple will drop Rossetta within a short time.

I and others need a great simplification of the route to LaTeX. For example, ready to run packages for innocent users who have a Mac Mini and a printer. Later, after some use, we may seek more complex systems.

Thanks.
Murray A Thompson (retired professor)
New Zealand

[Dear Murray, Thanks for the suggested articles. We will try to find authors. If you would like to contribute a note or article on your experiences please let us know. - The Editors]

Journal home page
General information
Submit an item
Download style files
Copyright
Contact us

THE PracTEX Journal

News from Around: Upcoming conferences in Pisa and Cluj (Romania); LaTeX tutorial days in London, San Francisco; New release of LaTeXDraw

The Editors

- Comment on this paper
- Send submission idea to editor

Upcoming conferences

The Italian TeX Users (GuIT) will be holding their annual **GuIT Meeting** on October 13, 2007 in Pisa. See the conference site for details. For a description of last year's well-attended meeting see the report by Onofrio de Bari.

(Recently, GuIT published the third annual issue of their journal, *Ars TeXnica*. Contact GuIT to order a copy of this nicely typeset journal (in Italian).)

The **Transylvania TeX Conference**, and the founding meeting of the **Romanian TeX Users (GROTeX)**, will be held September 1-2, 2007. The organizers are Paul Blaga, Zoltan Kasa and Horia Pop of the BABEŞ-BOLYAI University in Cluj, Romania. More details available at the conference site. This is a great opportunity to meet with knowledgable TeX and LaTeX users, and to visit this exotic and beautiful part of Europe.

Rub shoulders with TeX power-users from around the world at **TUG 2007** in San Diego, July 17-20, 2007. The keynote speaker is Peter Wilson, author of the memoir documentclass and other packages. For more information see the conference site.

LaTeX tutorial days

LaTeX tutorials will be held this fall in London and San Francisco. Each tutorial day will start with a LaTeX introduction, followed by a session where users of all levels can improve their skills. The presenters are experienced LaTeX teachers and users. The tutorials will be streamed for real-time viewing and also available for viewing later. Those interested in attending the tutorials or viewing the videos can contact PCTeX for more details.

Article update

Juergen Fenn's article Managing Citations and Your Bibliography with BibTeX in issue 2006-4 has been updated to correct some urls.

New Release

A new version (1.92) of the LaTeXDraw program, a Java GUI interface for PSTricks has been released very recently (30 May 2007). Details can be found on the web page of the project.

Journal home page
General information

Submit an item
Download style files
Copyright
Contact us

THE PracTEX Journal

PracTeX Journal: Making an Electronic Journal with web tools, Wiki, and version control

Paul A. Blaga

Abstract

This paper describes how an issue of TPJ is made up, and focuses on the software tools used. The tools used are the *wiki*, the *subversion* version control system, and Perl scripts. One of the goals of TPJ, and the reason for using these tools, is to create an environment where a small team of volunteers can put together an online journal with a minimum of time and work.

Paul Blaga is an Assistant Professor of Geometry at the Faculty of Mathematics and Computer Sciences of the "Babes-Bolyai" University in Cluj-Napoca, Romania. He started using LaTeX in the early 1990's. Initially, his interest was just for writing scientific papers, but afterwards he became more and more involved and gradually turned into a great LaTeX fan.

He has served several years as an editor of the mathematical journal *Studia Universitatis Napocensis, Mathematica* and was responsible for switching it to LaTeX in the late 1990's. He is also the co-author of a book on LaTeX.

His main interests in LaTeX are related, especially, to the graphical capabilities, but not only.

You can reach Paul at `pablaga@cs.ubbcluj.ro`

- PDF version of paper
- Comment on this paper
- Send submission idea to editor

The PracTEX Journal, 2007, No. 2
Article revision 2007/05/25

The PracTEX Journal: Making an Electronic Journal with web tools, Wiki and version control

Paul A. Blaga*

Address "Babeş-Bolyai" University of Cluj Napoca,
Faculty of Mathematics and Computer Science
1, Kogălniceanu Street,
400609 Cluj Napoca,
Romania

Abstract This paper describes how an issue of The PracTEX Journal is made up, and focuses on the software tools used. The tools used are the wiki, the *subversion* version control system, and Perl scripts. One of the goals of The PracTEX Journal, and the reason for using these tools, is to create an environment where a small team of volunteers can put together an online journal with a minimum of time and work.

1 Introduction

The PracTEX Journal is an international online journal aiming to help the spread and the development of different variants of TEX, from the "classical" Plain TEX and LATEX to "newcomers" such as ConTEXt. The articles published by The PracTEX Journal range from expository papers to technical ones, but all share a practical character.

There are some interesting aspects of the journal we would like to mention. First of all, it is published exclusively in electronic format. Second, the journal is compiled by editors from all around the world (Europe, United States, Australia, India) working together and simultaneously with the manuscripts. This is made possible by using two modern tools of electronic publications: the *wiki* and a version control system, *subversion*. The wiki is used by the editors to coordinate the

*The paper was actually written with the cooperation of Lance Carnes and Dave Walden. Thanks, also, to Francesco Reinaldo and Yuri Robbers for suggestions.

assembly of each issue. It can also be added to by readers and others who want to suggest an article idea. For us, it is an invaluable opportunity to exchange ideas, and a way of tracking the progress of producing an issue. The second tool, the *subversion* version control system, allows the editors to work simultaneously on the same manuscripts. Editors can update their local files to include all changes by other editors, so that at any point in time each editor has a current set of files for the current issue. It also lets the editors add or change files and has safeguards to prevent conflicting updates.

We shall explain, first, how these two tools work and then we will provide a description of how the journal itself is made up.

2 The Editorial process

The machinery of The PracTEX Journal has three main components: an automatically generated web site (designed by Dave Walden), a wiki (hosted by pctex.com), and the version control system (originally hosted at pctex.com and now at tug.org.)

Each issue of The PracTEX Journal has an issue editor who chooses a theme for the issue and solicits "theme" and other articles.

Now the wiki enters the stage. This is the place where prospective authors announce their projects. They can either edit the wiki page directly and add the title of their articles, or announce them to the editorial staff in order to have them take care of the process. Once an article is chosen for publication, a production editor is assigned to make it ready for publication. The *production editor* works with the author during the review and proofreading process. The progress of all articles is tracked in the wiki http://wiki.pctex.com/index.php?title=PracTeXJournal. This way the entire editorial staff is being kept up to date on the production process.

3 The PracTeX Journal wiki

The PracTEX Journal wiki is a web site that can be edited by the production editors in order to track the progress of the current issue. See the wiki at http://wiki.pctex.com/index.php?title=PracTeXJournal. A wiki is like an electronic white

board, but instead of using felt-tip pens it is edited online (much cleaner, and no smelly pens!). We note articles that have been received, what state they are in (proposed, received, in review, final), who is responsible for them, and other information.

For example, here is the wiki entry (in wiki syntax) for the article you are now reading.

```
|PracTeX Journal: Making an Electronic Journal with Wiki
   and Subversion || Paul Blaga et al || Paul || Proposed  || ~~~~~
```

This shows the working title, author, editor in charge, status (Proposed), and the date this was entered in the wiki. (The wiki changes `~~~~~` into the current date and time.)

As the issue progresses the schedule is updated. You are welcome to look at the wiki and see the progress of future issues. While you are there feel free to add an article idea http://wiki.pctex.com/index.php/PracTeXJournal#Article_ideas.

3.1 More about Wiki's

The name "wiki" is, surely, familiar to any web navigator, if only because of the free web encyclopedia Wikipedia http://wikipedia.org, probably one of the richest online encyclopedias. It is a good example of the wiki philosophy, because it was written by its readers and is constantly being updated.

To quote the beginning of the chapter on wikis from one of the latest books on virtual team management ([2]), "Wikis are collaborative websites that allow users to add and edit content." Behind these friendly websites there is software supporting the functions of the wiki. Usually, the word "wiki" is used both for the web page itself and for the software. The wiki was invented in 1994 by Ward Cunningham, and he also used this name for the first time. He didn't actually *invent* the name, it's just a Hawaiian word meaning "fast" or "quick". You can find the whole story in many places on the web (and, of course in the Wikipedia, as expected).

There are several wiki engines; some are free, some are commercial. We have no intention whatsoever to describe them. The interested reader can have a look

at the very nice book [4], one of the very few (if not the only!) entirely dedicated to wikis.

The way a wiki works is not difficult to understand. First of all, it is not entirely "democratic", as one might deduce from what we said so far. Any wiki page is created and put on the web by an *administrator*, which always has more rights and can restrict some of the rights of the users, if he thinks it's appropriate to do so.

No matter what software is used, on most wiki pages there are available at the least the following operations (beside the editing itself):

registration – usually, the reader is asked to register and then login before making any modification to the web page. Sometimes, the registration requires the approval of the administrator.

history – this is one of the most important features of a wiki website. It simply lets you track the previous versions and, if necessary, you can revert to any of these versions. This is essential both because you can correct errors made by changing involuntarily the contents of the site.

locking pages – the administrator can lock some of the page and only him will be able to modify these pages.

search – which has the obvious meaning, although not all the wikis would allow performing the entire range of searches.

recent changes – indicates what pages from the site were modified most recently.

As concerns the editing process, we have to say that so far there is no general consensus on the editing commands; each wiki software comes equipped with its own set. Most wikis allow only some elementary text modifier elements, some of the newer ones, however, are a lot more complex.

4 Version control for editing

While making up the first issue of The PracTEX Journal there was one person who coordinated all the article files. When an editor updated an article he would email it to the person in charge of files who would then update it into the issue archive. This person had to keep track of which files needed to be added or

updated, and hopefully put it them the right place. With ten or twelve articles, each with several revisions, this was fast becoming a nightmare. By the time the first issue was published the person in charge of files let it be known that this was a horrible, miserable job, and there had to be a better way.

We hit on a better way. It's called a version control system (VCS), a software system for maintaining versions of a large set of files. Since some of the editors are also software engineers, this was a natural choice; a VCS is used on most all software projects, including the TEX User Group's TEX Live project.

The VCS is the electronic form of the person in charge of files. It is a database of files and resides on a computer that can be accessed via the internet. Using some fairly intuitive commands the user can tell the VCS to "update this file into the TPJ 2007-2 issue" or "give me the latest version of the file `team.tex`", and other tasks. All files are sent and received automatically via the internet.

For those editors who had not used a VCS before it was a bit of a challenge to learn, but after a short while it paid off. We chose a VCS called *subversion*, a popular free software system. The usual way of using *subversion* is with a command line (see the next section for some examples). There are also some nice graphical user interfaces (GUI) that can be used with Windows, Linux and other systems. The editors who are not software developers (and those who are too lazy to memorize all the command line options) use a GUI interface.

Briefly, here is how a production editor would use *subversion* to work on the file for this article `team.tex`. (`svn` is the command to run *subversion*.)

- Starting up. Before editing, the local file copies must be brought up to date. This step updates any work was done by other editors into the local copy. Just type

  ```
  svn update
  ```

- Editing. Once the local files are current, the editor may use his/her favorite text editor to make changes to the files. Just open `team.tex` in a text editor and work as usual.

- Winding up. Once the edits are finished it is time to put the updated file in the archive, so that the other editors will be able to benefit from (or complain about) the changes made. To put the changes into the master archive, type

  ```
  svn commit -m "corrected a bunch of errors - can't you guys spel?"
  ```

The text after `-m` is a comment that will appear in the archive log.

That's a typical *subversion* session. It can do dozens of other useful things as well — see the next section for more details.

4.1 *Subversion* and version control systems

Wikis are, first all, great ways of *exchanging information*. Version control systems ([3, 1]), are a powerful tool for online *collaboration*. These systems are used mostly by programmers and others to coordinate a project composed of a set of interrelated files. The version control system used by the PracTEX Journal is *subversion*.

Subversion, and other version control systems, are at the core a database consisting of all the files used in a project. To access and update this database there are several user interfaces available for most all operating systems. Some of them have a graphical user interface (GUI), while other are operated by command line. We shall describe the command line version, which is used in Unix and similar systems.

Subversion allows several people to work simultaneously on the same project. The project's subversion database is located on a system which can be accessed from the internet. Collaborators are given an account which allows them to access it and modify it.

Thus, the first step in working with subversion is the creation of a space for the project. This step is performed by an *administrator* and the corresponding database (and the contents of the project which is stored there) is called a *repository*. Then the administrator creates accounts for the users who are authorized to work on the project.

At this point the users enter the stage. We assume that the interface software is already installed. The first step is to create copies of the project files on their own computers. This is very easy to do (once they are logged on). First, they have to create a directory where everything is supposed to lie (for instance, a "practex" directory, on the drive D). Then change directory to the chosen one (this is the windows version, of course, the modifications for Unix should be obvious):

```
cd D:\practex
```

Now, the user can download the repository, to create a local copy. For instance, for PracTEX the command is:

```
svn co svn://tug.org/pracjourn/trunk
```

This may take a while depending on the dimension of the project and the speed of the connection. The good news is that the steps so far have to be done only once.

Now, that everything is on your computer, you can modify everything you want and then propagate the changes to the central repository. Typically, a work session would look like that:

First of all, copy from the central repository the latest changes. This is done with the command:

```
svn update
```

All the changes will be downloaded. You will also get a list of the changed files/directories and the number of the version of the project.

Now you can do your everyday work. Normally, you will perform three kind of operations: editing files, creating files and creating directories. Beware that even if you create by the commands specific to your operating system, files and directory, or you copy them on your local repository from somewhere, they don't exist for subversion and will not be uploaded to the central repository unless you let subversion know what you did. This is done by the subversion command `add`:

```
svn add file1 file2 directory1
```

If you're not sure whether you "added" all the files and directory you intended to do, or did not modify all the necessary files, you can run a check:

```
svn status
```

and you will get the list of the modifications you made.

And now, the final step, the commitment (which is important in subversion, as well):

```
svn commit -m "Describe the changes you made."
```

the option `-m` stands, of course, for "message".

Now all your changes are propagated to the central repository. Notice that no changes is propagated until you give the commit command.

An important pair of commands allow the locking and unlocking of a file/directory. If you want to modify a file and you want to be sure that nobody will perform another modification at the same time with you, you can use the command:

```
svn lock file
```

and now nobody will be able to modify the file until you commit your changes or until you unlock the file with the command

```
svn unlock file
```

5 Putting everything together

After the review process, and an article is being worked on for the current issue, a directory is created and all files related to the article are added. This makes the paper available to the entire staff, via the VCS, and allows the production editor to make simple changes or fix problems in conjunction with the author, or even to replace the entire paper by an updated version submitted by the author. In the final stages the production editor changes the article header to reflect that this is the accepted version rather than a submitted paper, and updates the date of acceptance. Then an abstract and a short author biography for the journal website is entered. This will be used during the automated website generation.

When all papers are ready, the editorial has been written, etc., the editor in charge will take care of the final touches, which includes producing a single pdf file of the entire journal, and the current issue will be placed on a testing web site. The editorial staff as well as a group of carefully selected proofreaders then check for any remaining problems. After a short testing period, the "final" web site for the issue is made public.

6 Web site generation and updating

The *TPJ* website is generated by a Perl script. There are at least three advantages in having a website generation program such as we use for *TPJ*: (1) the HTML pages for the tables of contents for all issues and for each piece in each issue have consistent formats; (2) if we decide on a change in format, we can change every piece and every issue with (hopefully) a small change to the program or (better yet) an HTML template for a particular format; (3) as the program is processing all of the pieces in every issue, it also makes title, author, and `bibtex` indexes across all issues and pieces.

6.1 More about site generation

For readers who are interested, the program works as follows: The svn repository for the journal contains a directory for each issue, with directory names that give the issue number, e.g., `pracjourn/2007-2/`. Each such directory includes a "driver file" that tells the generation program the titles and authors of each piece in the issue as well as grouping the pieces into the categories of Notices, Articles, and Columns. The HTML table of contents for each issue is generated from the data in this driver file. There is also a directory for each piece in the issue. Various of the following are included in this directory: an abstract of the piece, a PDF of the piece, HTML for the piece in the instance where there is no PDF for the piece, etc. The website generation program processes the information in the driver file for each issue, creating the HTML page for the table of contents for the issue and the HTML pages for each piece in the issue, as well as the various indexes. When we are producing a new issue, we typically run the program in a mode where only the HTML pages for the latest issue are generated along with the indexes for all issues. The program also uses a couple of other small driver files to create the left column of the home page and some HTML pages linked to from the left column.

7 Application to other journals

It's possible that some of the techniques described here could be applied to other print or electronic journals. The wiki is essentially an electronic white board and can be learned easily. Whenever someone involved with a project has a new idea or a new method, he or she can post it to the wiki. It's a great way to keep everyone on the same page (publishing pun).

If a journal, either print, electronic or both, has a number of editors who are updating articles regularly, the Version Control System (VCS) is a useful tool. With a graphical interface it's easy to use. It avoids collisions (editor A accidentally overwriting editor B's work), it provides a log of everything that has been done, it allows an earlier version of file to be restored, and many more functions. It will even show the differences between versions of a file — if editor A takes the day off and her colleague editor B fills in for her, when she returns the VCS will quickly show all changes made.

There are a number of other little tricks we have employed to make The PracTEX Journal a low-cost way for a volunteer group to publish a quarterly journal. If you would like more information on any of the techniques discussed feel free to get in touch mailto:pracjourn@tug.org. You may also register at the wiki site http://wiki.pctex.com/index.php?title=PracTeXJournal and post your comments.

References

[1] Berlin, D., Rooney, G.: *Practical Subversion*, 2nd edition, Apress, 2006

[2] Brown, M.K., Huettner, B., James-Tanny, C.: *Managing Virtual Teams: Getting the Most from Wikis, Blogs and Other Collaborative Tools*, Wordware Publishing, Inc., 2007

[3] Collins-Sussman, B., Fitzpatrick, B.W., Pilato, C.M.: *Version Control with Subversion*, O'Reilly, 2004

[4] Ebersbach, A., Glaser, M., Heigl, R.: *Wiki Web Collaboration*, Springer, 2006

THE PracTEX Journal

PDFpages for Editors and Publishers

Yuri Robbers

Abstract

There are many ways in which the *PDFpages* package by Andreas Matthias06 can be helpful to editors. An obvious one is collating several papers into one. This paper will describe a few of the many ways in which pdfpages can make life easy for the editor and publisher.

Yuri Robbers holds a degree in Animal Behaviour and is a teacher, a researcher and a published author. He's always had a keen interest in typography, possibly because his father is a professional typographer. Ever since he discovered LaTeX in 1995, he's always done his best to avoid using word processors, and he has embarked on a quest to learn as much as possible about TeX and its derivatives and to apply this knowledge whenever possible. Contact him at yuri.robbers@gmail.com

- PDF version of paper
- Comment on this paper
- Send submission idea to editor

The PracTEX Journal, 2007, No. 2
Article revision 2007/05/16

PDFpages for Editors and Publishers

Yuri Robbers

Email yuri.robbers@gmail.com

Abstract There are many ways in which the PDFpages package (Matthias, 2006) can be helpful to editors. An obvious one is collating several papers into one. This paper will describe a few of the many ways in which pdfpages can make life easy for the editor and publisher.

1 Introduction

Editors often need to manipulate existing PDF documents in various ways that would be difficult, time consuming, or in extreme cases downright impossible in basic LaTeX2ε. One could think of concatenating various papers into one large PDF (possibly even with hyperlinks from a table of contents to each included paper), shrinking or enlarging pages, rearranging the pages of a document (for example into signatures, for printing as a booklet), creating a new document that is a subset of the original document or including pages of a document as illustrations in another document (perhaps the front covers of books in a catalog). There are, of course, many more possible applications.

When faced with such tasks, the life of an editor suddenly becomes a lot easier with the PDFpages package (Matthias, 2006) at their disposal.

In this paper I will show a few of the many possible applications of PDFpages.

2 Concatenating papers

It happens a lot when editing journals, such as the PracTEX Journal. One has a whole bunch of papers, and they need to be combined in a single "whole issue" PDF. Since the PracTEX Journal consists of materials written in TEX, LaTeX2ε, ConTEXt and HTML this is not necessarily a simple task. We cannot just create one big TEX source file and compile it. Even when using just LaTeX2ε it is possible to run

into trouble when one paper contains a package or style file that is incompatible with a package or style file in another paper.

In order to avoid these problems it is often easiest to create a PDF of each paper, and concatenate those. The traditional way of doing this uses Adobe® Acrobat Professional. This program is still one of the best (if not the best) at converting HTML into PDF, and also allows concatenating PDFs into one large PDF by mouseclick.

Apart from being free software the PDFpages package, however, has a very important advantage to offer, even in this simple case: programmability. The PDFpages package is not able to turn HTML into PDF, so that task will have to be performed using a different program, but the next step — concatenation — profits from programmability. Suppose one has forgotten to include a certain paper. Or one of the proofreaders (or worse: readers) has discovered a glaring error in one of the papers. With the mouse click approach to concatenation one will have to do the browsing of directories and clicking of files all over again. With another opportunity for mistakes. With the PDFpages approach, one simple types

```
pdflatex wholeissue.tex
```

and the whole issue is regenerated. Quick and easy, with no chance of mistakes. The only thing one needs to do is create the `wholeissue.tex` file once, and then concatenation is a simple matter of running the command above.

An example of such a `wholeissue.tex` file is given in figure 1. As one can see, apart from the usual preamble, the document consists of a single command `\includepdfmerge`. This command has an argument with a whole list of PDF filenames, followed by a comma, a minus sign, and another comma. The minus sign is the notation the PDFpages package uses to signify "all pages". It is possible to give a single number here in order to include just one page, or a page range, or even a complex list of pages, such as `1,3,4,8-12,16,15,14,19-22,35,1,1,1`. As one can see, it is even possible to include pages in a different order, or to include the same page multiple times.

Please note that the last paper that is being included does *not* have a comma after the minus sign: the comma signifies that either a further reference to a page, or another file to included will follow.

Since this example is part of the concatenation file for the 2007-1 issue of the PracTEX Journal, it contains two PDF files for most papers, since those papers come as a PDF file with an additional HTML file containing an abstract of

```
\documentclass{amsart}
\usepackage{pdfpages}
\begin{document}
\includepdfmerge[fitpaper=true]{%
contents.pdf,-,%
editorial.pdf,-,%
news.pdf,-,%
readers.pdf,-,%
beccari-html.pdf,-,%
beccari.pdf,-,%
gregorio-html.pdf,-,%
gregorio.pdf,-,%
robbers-html.pdf,-,%
robbers.pdf,-,%
madsen-html.pdf,-,%
madsen.pdf,-,%
cho-html.pdf,-,%
cho.pdf,-,%
blaga-html.pdf,-,%
blaga.pdf,-,%
baechle-html.pdf,-,%
baechle.pdf,-,%
neveln-html.pdf,-,%
neveln.pdf,-,%
asknelly.pdf,-,%
distract.pdf,-%
}
\end{document}
```

Figure 1: An example `wholeissue.tex` file that will concatenate a series of PDF files into one whole issue PDF for an online journal.

the paper and a short biography of the author. Both files are included in the `wholeissue.pdf` file.

Should one discover that an article has been missed in the whole issue PDF, one can easily add that file to the `wholeissue.tex` and regenerate the whole issue PDF. Similarly, it is easy to change the order of the papers in the whole issue PDF, or to regenerate the whole issue PDF when the PDF file containing one of the papers has been changed.

3 Producing screen and print versions of eBooks

There is a small independent publisher[1] in the Netherlands that produces eBooks, which I typeset for them in LaTeX2ε, using Peter Wilson's memoir document class (Wilson, 2004).

As an extra service to customers each eBook is presented in a so-called *screen version* as well as a *print version*. The screen version contains the front cover, the copyright page, the story itself and the back cover. It is intended to be read on a computer screen or a dedicated eBook reader such as the iLiad[2] or Sony Reader[3].

The print version contains the front cover, title page, copyright page, story and back page. Blank pages have been added wherever they are appropriate, as well as at the end of the story, if necessary, to make sure the total number of pages is a multiple of four. The pages have then been ordered into so-called signatures: an ordering of the pages different form the usual numerical one with two pages being printed on each side of a sheet of paper. This PDF file can be printed double sided on a printer, folded in half and — if desired — stapled through the back and/or cut to size. This way one has a small booklet which can easily be taken along in a pocket.

Both the screen version and the print version are being generated by means of the appropriate PDFpages file. Given a PDF file containing the title page, copyright page and story (generated by pdfLaTeX based on my memoir file), a PDF file containing the front cover, a PDF file containing the back cover, and a PDF file containing a blank page, I can generate the screen version with one command, and the print version with two.

1. Saga-Whyte Press, http://www.sagawhytepress.eu
2. http://www.irextechnologies.com and http://www.sagawhytepress.eu
3. http://www.learningcenter.sony.us/assets/itpd/reader/

3.1 Screen version

Suppose there is a directory for each eBook — "*Sheena and the Gun*" (Skjold, 2007b) shall be our example story — containing all the relevant files. Among the files in this directory is the PDF file of the story itself called `sheena.pdf`. The story file starts with a title page, then a copyright page, and then the story itself. In this directory there are also PDF files for the cover, aptly named `frontcover.pdf` and `backcover.pdf`.

In order to create the screen version of this eBook, all we need to do is run this PDFpages file through pdfLaTeX:

```
\documentclass{amsart}
\usepackage{pdfpages}
\begin{document}
\includepdfmerge[fitpaper=true]{%
frontcover.pdf,%
sheena.pdf,2-,%
backcover.pdf%
}
\end{document}
```

Since we already have the front cover, a title page is not truly necessary in the screen version of the eBook. That is why the story file itself is included from page 2 onwards. Also note that for PDF files that contain only one page, it is possible (though not at all necessary) to leave out the page range designation.

3.2 Print version

Creating the print version of the same eBook unfortunately requires two runs of pdfpages. The reason is that if one combines the concatenation of the files and the re-ordering of the pages into a signature in one run of PDFpages, one gets extraneous white space in various places. There might be a more elegant solution to this than mine, but I have not been able to find it.

My solution entails two runs of pdfTeX. The first one takes care of concatenation:

```
\documentclass{amsart}
\usepackage{pdfpages}
```

```
\begin{document}
\includepdfmerge[fitpaper=true]{%
frontcover.pdf,%
blankpage,-,%
sheena.pdf,-,%
blankpage,1,1,1,%
backcover.pdf%
}
\end{document}
```

where I include the blank page between the story and the back cover three times by including its first (and only) page three times. The first two are to make sure that the total number of pages is divisible by four, and the third one is the inside back cover (just as the very first blankpage between the front cover and the story is the inside front cover).

Then I run the resulting PDF file through PDFpages again, in order to take care of the re-ordering:

```
\documentclass{amsart}
\usepackage{pdfpages}
\usepackage[paperheight=250mm,paperwidth=168mm]{geometry}
\begin{document}
\includepdf[pages=-,landscape,signature=28]{sheena-concatenated.pdf}
\end{document}
```

There are a few changes to be seen here. First of all in the preamble. I use the geometry package in order to set the paperheight to equal twice the pagewidth and the paperwidth to equal the pageheight. This way I can turn the pages and fit two of them together onto one sheet.

Next I use `\includepdf` rather than `\includepdfmerge` because I am not merging files, just using the one file. When using `\includepdf` one includes the page specification in the optional argument. The other optional argument, landscape, switches width and height of the paper. For reasons unknown to me one cannot just refrain from exchanging paperwidth and -height and from using the landscape option, because then the final result will have extraneous white again. The option `signature=28` tells PDFpages that one wants a bunch of 28 pages to be generated in such a way that the resulting printout can be folded in half and have

a booklet with all the pages in the right order.

One could of course specify a larger number than the number of pages present. In that case PDFpageswould add blank pages at the end of the document. In this example that could not be used, since the back cover needed to be included as the very last page.

Specifying a smaller number than the number of pages is a very useful function too in printing. When looking at a printed book, a novel for example, one sees that the book actually consists of a number of small booklets stacked on top of each other. Each booklet has 32, or sometimes 16 pages. This effect is achieved by setting `signature=32` or `signature=16` respectively. The printer would churn out stacks of booklets of 32 or 16 pages each, that can be sown, then stacked on top of each other and finally glued together into the cover, which must have an actual spine in this case rather than just being the front and back cover printed together on one sheet.

4 Previews of books

Something else that can useful for a publisher is the creation of a preview. Creating a preview of a short story works much like creating the screen or print version of a booklet as shown above. The main difference is that one includes only part of the main book file. For a screen preview of the short story used above, one would include the front cover, the copyright page, the first two or three pages of the story, perhaps a page with information about the author and publisher, and where to obtain the full book, and then the back cover.

For a print preview one would do the same, but include the correct number of blank pages wherever appropriate, and then run a second pass in order to create a signature PDF file.

Take, for example, the novel "*Scent of Summer Magnolia*" (Skjold, 2007a). This is a novel consisting of 560 pages, which includes the frontmatter, mainmatter and backmatter. In addition, there is the front cover, back cover and spine. For the preview we will not be using the spine. We want to include both the covers, the front matter (title page, copyright page, table of contents, etc.), the first two chapters of the novel itself, and the back matter ("*About the author*"). The concatenation file would look as follows:

```
\documentclass{amsart}
\usepackage{pdfpages}
\begin{document}
\includepdfmerge[fitpaper=true]{%
frontcover.pdf,%
blankpage.pdf,%
magnolia.pdf,1-18,559,560,%
blankpage.pdf,%
backcover.pdf%
}
\end{document}
```

Note that here we do not need to have any additional blank pages, since our selection of pages already comes to a total number divisible by four.

The creation of the signature file requires just changing the number of pages to be included in the signature (24), the papersize and the filename in the signature file given at the end of subsection 3.2:

```
\documentclass{amsart}
\usepackage{pdfpages}
\usepackage[paperheight=12in,paperwidth=9in]{geometry}
\begin{document}
\includepdf[pages=-,landscape,signature=24]{magnolia-concatenated.pdf}
\end{document}
```

And this way one has a booklet that can be printed, folded, stapled and given away to potential customers in order to entice them into buying the book.

5 Including (parts of) documents as illustrations

The final useful example I will give here is the inclusion of existing documents or parts of existing documents as illustrations in a new document. This can be useful when discussing, for example, an existing text or figure from a publication available in PDF format, or in order to display covers of books (in a catalog perhaps) or journals (in an index for a complete volume of that journal perhaps.

In this section I will show how to do such an inclusion using PDFpages. As an example I will use part of the PDFpages documentation (Matthias, 2006). Of

course I will not be including all pages, but I will include four, in order to show one page of thumbnails.

This code

```
\includepdf[nup=2x2,pages={1-4},frame=true,%
noautoscale,scale=.35]{pdfpages.pdf}
```

produces the last page of this document. Please note that PDFpages package creates *pages* rather than graphics objects, so they cannot be put inside floats, or otherwise be made just part of a page rather than an entire page.

6 Conclusion

The PDFpages package has many useful applications, especially for editors. This paper shows a few of them. Do read the PDFpages documentation (Matthias, 2006) for further information, and experiment, for it is practice that makes perfect.

7 Acknowledgments

I am grateful to Dave Walden for his comments on a draft of this paper.

References

Matthias, A. (2006). The pdfpages package. CTAN://macros/latex/contrib/pdfpages.

Skjold, A. (2007a). *Scent of Summer Magnolia*. Leiden, the Netherlands: Saga-Whyte Press.

Skjold, A. (2007b). *Sheena and the Gun*. Leiden, the Netherlands: Saga-Whyte Press.

Wilson, P. (2004). The memoir class for configurable typesetting. CTAN://macros/latex/contrib/memoir/.

The pdfpages Package*

Andreas MATTHIAS
amat@kabsi.at

2006/08/12

Abstract

This package simplifies the insertion of external multi-page PDF or PS documents. It supports pdfLaTeX and VTeX.

Contents

1 Introduction

When creating PDF documents, it is sometimes useful to insert pages of other, external PDF documents. This can be done with the `\includegraphics` command from the `graphics` package. But a simple `\includegraphics{doc.pdf}` normally produces 'Overfull \hbox' and 'Overfull \vbox' warnings, because the size of the inserted pages does not match the print space.

The `pdfpages` package makes it easy to insert pages of external PDF documents without worrying about the print space. Here are some features of the `pdfpages` package: Several logical pages can be arranged onto each sheet of paper and the layout can be changed individually. A lot of hypertext operations are supported, like links to the inserted pages, links to the original PDF document, threads, etc. When working with VTeX the same is possible with PostScript documents, too. Note that PostScript documents are only supported by VTeX and *not* by pdfLaTeX.

*This file has version number v0.4a, last revised 2006/08/12.

When producing DVI output `pdfpages` cannot insert pages of a PDF documents. But instead of interrupting execution `pdfpages` will insert empty pages. This feature is important when using packages like `pst-pdf`, which need to produce DVI output at the first run.

2 Usage

2.1 Package Options

`\usepackage[`⟨*options*⟩`]{pdfpages}`

⟨*option*⟩ – **`final`:** Inserts pages. This is the default.

`draft`: Does not insert pages, but prints a box and the filename instead.

`enable-survey`: Activates survey functionalities. (*experimental, subject to change*)

2.2 Commands

`\includepdf` Inserts pages of an external PDF document.

`\includepdf[`⟨*key=val*⟩`]{`⟨*filename*⟩`}`

⟨*key=val*⟩ – A comma separated list of options using the ⟨*key*⟩=⟨*value*⟩ syntax.

⟨*filename*⟩ – Filename of the PDF document. (The filename *must not* contain any blanks!)

The following list describes all possible options of `\includepdf`. All options are using the ⟨*key=value*⟩ syntax.

- Main options:

pages Selects pages to insert. The argument is a comma separated list, containing page numbers (`pages={3,5,6,8}`), ranges of page numbers (`pages={4-9}`) or any combination. To insert empty pages use {}.
E.g.: `pages={3,{},8-11,15}` will insert page 3, an empty page, and pages 8, 9, 10, 11, and 15.
Page ranges are specified by the following syntax: ⟨*m*⟩`-`⟨*n*⟩. This selects all pages from ⟨*m*⟩ to ⟨*n*⟩. Omitting ⟨*m*⟩ defaults to the first page; omitting ⟨*n*⟩ defaults to the last page of the document. Another way to select the last page of the document, is to use the keyword `last`. (This is only permitted in a page range.)
E.g.: `pages=-` will insert *all* pages of the document, and `pages=last-1` will insert all pages in reverse order.
(Default: `pages=1`)

nup Puts multiple logical pages onto each sheet of paper. The syntax of this option is: `nup=`⟨*xnup*⟩`x`⟨*ynup*⟩. Where ⟨*xnup*⟩ and ⟨*ynup*⟩ specify the number of logical pages in horizontal and vertical direction, which are arranged on each sheet of paper. (Default: `nup=1x1`)

landscape Specifies the format of the sheet of paper, which is rotated by 90 degrees. This does *not* affect the logical pages, which will *not* be rotated by the 'landscape' option. To rotate the logical pages use the 'angle' option (e.g. 'angle=90'). Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `landscape=false`)

- Layout options:

delta Puts some horizontal and vertical space between the logical pages. The argument should be two dimensions, separated by space. See Chapter 2.3 and Figure 1. (Default: `delta=0 0`).

offset Displaces the origin of the inserted pages. The argument should be two dimensions, separated by space. In 'oneside' documents positive values shift the pages to the *right* and to the *top* margin, respectively, whereas in 'twoside' documents positive values shift the pages to the *outer* and to the *top* margin, respectively. See Chapter 2.3 and Figure 1. (Default: `offset=0 0`)

frame Puts a frame around each logical page. The frame is made of lines of thickness `\fboxrule`. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `frame=false`)

column Pdfpages normally uses '*row-major*' layout, where successive pages are placed in rows along the paper. The `column` option changes the output into a 'column-major' layout, where successive pages are arranged in columns down the paper. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `column=false`)

columnstrict By default the last page is not set in a strict 'column-major' layout, if the logical pages do not fill up the whole page. The `columnstrict` option forces a strict 'column-major' layout for the last page. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `columnstrict=false`)

1	4	
2	5	
3		

`columnstrict=true`

1	3	5
2	4	

`columnstrict=false`

openright This option puts an empty page before the first logical page. In combination with `nup=2x1`, `nup=2x2`, etc., this means that the first page is on the right side. The same effect can be achieved with the `pages` option, if an empty page is inserted in front of the first page. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `openright=false`)

pagecommand Declares LaTeX commands, which are executed on each sheet of paper. (Default: `pagecommand={\thispagestyle{empty}}`)

turn By default pages in landscape format are displayed in landscape orientation (if the PDF viewer supports this). With `turn=false` this can be prohibited. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `turn=true`)

noautoscale By default pages are scaled automatically. This can be suppressed with the `noautoscale` option. In combination with the `scale` option (from graphicx) the user has full control over the scaling process. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `noautoscale=false`)

fitpaper Adjusts the paper size to the one of the inserted document. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `fitpaper=false`)

reflect Reflects included pages.

signature Creates booklets by rearranging pages into signatures and setting `nup=1x2` or `nup=2x1`, respectively. This option takes one argument specifying the size of the signature, which should be a multiple of 4.
An example for documents in portrait orientation:
```
\includepdf[pages=-, signature=8,
            landscape]{portrait-doc.pdf}
```
An example for documents in landscape orientation:
`\includepdf[pages=-, signature=8]{landscape-doc.pdf}`

signature* Similar to `signature`, but now for right-edge binding.

picturecommand Declares picture commands which are executed on every page within a picture environment with the base point at the lower left corner of the page. (The base point does not change if the page is rotated, e.g. by the `landscape` option.) (Default: `picturecommand={}`)

picturecommand* Like `picturecommand`, but with the restriction that `picturecommand*` executes its picture commands only on the very first page. (Default: `picturecommand*={}`)

pagetemplate By default the first inserted page will be used as a template. This means that all further pages are scaled to match within the contour of this first page. This option allows to declare another page to be used as a template; which is only useful if a PDF document contains different page sizes or page orientations. The argument should be a page number. (Default: `pagetemplate=`⟨*first inserted page*⟩)

rotateoversize This option allows to rotate oversized pages. E.g. pages in landscape orientation are oversized relatively to their portrait counterpart, because they do not match within the contour of a portrait page without rotating them. By default oversized pages are scale and are *not* rotated. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `rotateoversize=false`)

doublepages Inserts every page twice. This is useful for 2-up printing, if one wants to cut the stack of paper afterwards to get two copies. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `doublepages=false`)

doublepagestwist Whereas with `doublepages` the cutting edge is once on the inner side and ones on the outer side, `doublepagestwist` turns the pages such, that the cutting edge is always on the inner side. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `doublepagestwist=false`)

doublepagestwistodd Turns the pages such, that the cutting edge is always on the outer side. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `doublepagestwistodd=false`)

Journal home page
General information

Submit an item
Download style files
Copyright
Contact us

THE PracTEX Journal

LaTeX and the different bibliography styles

Federico Garcia

Abstract

Although I chose `style' for the title of this article, it is perhaps best to specify right away that the article is not devoted to styles of formatting entries in a final bibliography list (i.e., a `style' in BibTeX terms, as defined by the.bst file). Rather, we will be looking at the different types of in-text citation: citation by footnote, by parenthesis labels, or by brackets. The citation style admittedly determines certain aspects of the entry-formatting style, but the two things are pretty much independent, and an article such as this one can focus on only one of them.

Federico Garcia is a composer, chess player, and lover of TeX. He has collaborated with his uncle Rodrigo De Castro in several Spanish-language books on TeX and LaTeX. Since moving to Pittsburgh in the US in 2001, he has written a number of packages, including opcit, TeXmate, and subfiles. Visit Federico at www.fedegarcia.net.

- PDF version of paper
- Comment on this paper
- Send submission idea to editor

For submission to The PracTEX Journal
Draft of May 13, 2007

LaTeX and the different bibliography styles

Federico Garcia

Email federook@gmail.com
Website www.fedegarcia.net

Contents

1 Introduction

Although I chose 'style' for the title of this article, it is perhaps best to specify right away that the article is not devoted to styles of formatting entries in a final bibliography list (i.e., a 'style' in BibTeX terms, as defined by the `.bst` file). Rather, we will be looking at the different types of in-text citation: citation by footnote, by parenthesis labels, or by brackets. The citation style admittedly determines certain aspects of the entry-formatting style, but the two things are pretty much independent, and an article such as this one can focus on only one of them.

The twofold thesis of this article is that there are three main citation styles (the ones mentioned: footnote, brackets, and parentheses), and that LaTeX in 2007 provides virtually complete support for all of them. Today (but not five years ago) it is the case that the choice of citation style is not subject to what the software allows, but is really up to the user (within certain limits at least, since institutions — journals, etc. — influence the decision by enforcing one style or another).

In this circumstance, it seems like a good idea to carry out a survey of the three families of bibliographical citation and their support in LaTeX, and that is my purpose here.

The present article stems from a talk I gave at the 2006 Practical TeX conference.

2 The three main style families

It is curious how proponents of each of the three styles — usually — 'don't like' the other styles. They (we) tend to have strong ideas about why one of the styles (our own, usually) is best, and seldom stop to reflect how it is that whole groups of intelligent people have a directly opposite opinion. One thing is for sure: styles are roughly chosen according to discipline. As a result, in our upbringing we are usually exposed to one of the styles far more than the others. We get used to it, and then the others, when we encounter them, do feel a little odd. We conclude, naturally enough, that don't like them.

But the truth is that there are good reasons why each of the styles exists and is used. In this section I try to make some of these reasons explicit. I won't hide the fact that I lean strongly toward the footnote-style, but I will try to do the others some kind of 'objective' justice.

3 Bracket styles

LaTeX's native support for bibliographical referencing is directed exclusively toward the family of bibliography styles where the citation is done through brackets: something like [1], or [Cas44].

This family of styles, most familiar for LaTeX users, has one immediate advantage: the expression in brackets makes sense *both* as a parenthetical comment *and* within the sentence proper. In other words, one can equally make direct reference to a publication (as in 'see [2]' or '[3] is a good reference for...') or simply add the reference as a clarification (as in 'this has already been proven [2]').

Another advantage is that the brackets can be freely used in conjunction with parentheses, so that the form of the actual reference does not depend on the context. It is equally admissible to say 'I once read a book [2] where...' and 'I saw once (in a book that I read [2]) that...'

This efficiency of the brackets is the main reason why this family of bibliography styles needs only one command name: `\cite`. This, and the fact that it is the natural behavior of LaTeX, means that I need to say little more about this style. (In fact, what I have already said was basically for purposes of comparison.)

4 Author-year styles

4.1 Introduction

When the proof of a theorem makes reference to a previously proven theorem, the author and the publication date of the previous paper are not crucial to the argument. Whoever might be interested in that proof in itself can consult the final list of references, and start the search. But for the purposes of the original argument, things like the author, the title, and year of the references are, generally, of no consequence in this kind of discourse.

On the other hand, if an author is referring to previous essays on — say — ethical perception of environmental issues, *then* information on who wrote those previous essays, when they were written, and even what they are called, can be absolutely crucial to the argument. After all, in this context it is not the same to quote a French postmodern philosopher as to quote a study by the Department of Defense...

It is in these contexts that the bracket citation style is truly insufficient. This

kind of discipline — let's keep calling them 'the humanities' — has come to adopt widely an alternative kind of citation, generally known as author-year citation. It consists basically in embedding some of the crucial information (the author and the year) into the label of the citation: instead of '[1]', one would have '(Cassirer, 1944)'.

Note that one of the native LaTeX bracket styles, `alpha`, is a compromise between the two things — in 'Cas44', Cas is the first three letters of the last name, and 44 is the year. However, even in this case, `alpha` is oriented more towards the sciences than the humanities: what if one cites Nietzsche, who wrote in the 1800s?

In any case, the most relevant difference between this style and the LaTeX default is that — for unknown reasons, I might add — author-year styles use regular parentheses instead of brackets.

This has a wealth of interesting consequences. Parentheses, unlike brackets, have a meaning other than bibliography, and, alas, the two meanings collide. I can say 'this has already been argued (Cassirer, 1944)'. But things like 'for this issue see (Cassirer, 1944)' or '(Cassirer, 1944) is a good reference for…' are funny. Even funnier results are produced by citations within parentheses: "I saw once (in a book I read (Cassirer, 1944)) that…"

Thus, these styles tend to feature a number of variations to the way sources are actually cited, designed to solve the dilemmas of grammar and aesthetics illustrated above. So:

- This has already been argued (Cassirer, 1944).
- For this issue see Cassirer, 1944.
- Cassirer (1944) is a good reference for…
- I read once a book (Cassirer, 1944) where…
- I saw once (in a book I read [Cassirer, 1944]) that…

The choice of the right kind of citation is probably beyond complete automation. That means that it is the user that has to choose. And, in turn, this means that many different commands have to be available. In fact, LaTeX packages that support this family of styles have an unusually large number of citation commands.

	Some philosophers of science have claimed that science progresses as life (notably Kuhn 1996, Koestler 1959). Kuhn started his conceptual trip with his exploration of the Copernican Revolution (1957). At virtually the same time, at the other side of the Atlantic, the same trip was documented in Koestler (1959). In musicology, Leo Treitler expresses very similar views (1984, 1989, 1999). Kuhn's 'paradigms' were directly addressed by musicians since 1991 (McClary).
`\citeaffixed`	
`\citeyear`	
`\citeasnoun`	
`(\citename...)`	
	There was and is, of course, opposition to Kuhn, whose ideas were always shunned by mainstream philosophy of science. Most virulent of all was the criticism by Imre Lakatos (1970). In musicology, this criticism has no direct offspring, but the other extreme, the 'anarchy of knowledge' (Feyerabend 1978), finds parallels in the diverse manifestos of postmodernist musicology, for example Tomlinson's (1984).
`\cite`	
`\possessivecite`	

Figure 1: harvard sample

4.2 Samples

I will refer to three particular packages, all very successful, that support author-year citation: harvard, achicago, and natbib. The following three pages feature samples of how they work. They are built in such a way that PgDn and PgUp allows immediate comparison. The citations are in red so as to make them more prominent. (The reader might want to take a moment to scroll through these samples and get the feeling of the differences, and the similarities, between the three packages.)

As can be readily seen, translation between the three is pretty straightforward. But it is interesting to see the different command names that the three authors chose for the several citation variants. In harvard (the first, seminal one, by Peter Williams and Thorsten Schnier, final version 1994), the naming follows a 'logical' or 'grammar-oriented' model: citations are qualified by the grammatical function of the label in the sentence. When the citation is a noun, you use `\citenoun`; when something has to be affixed to the parenthesis *before* the citation proper, you type `\citeaffixed` (for 'suffixes', additions *after* the citation, the optional argument of `\cite` is used).

	Some philosophers of science have claimed that science progresses as life (notably Kuhn 1996; Koestler 1959). Kuhn started his conceptual trip with his exploration of the Copernican Revolution (1957). At virtually the same time, at the other side of the Atlantic, the same trip was documented in Koestler (1959). In musicology, Leo Treitler expresses very similar views (1984, 1989, 1999). Kuhn's 'paradigms' were directly addressed by musicians since 1991 (McClary).
(\citeNP)	
\citeyear	
\citeN	
\citeA	
	There was and is, of course, opposition to Kuhn, whose ideas were always shunned by mainstream philosophy of science. Most virulent of all was the criticism by Imre Lakatos (1970). In musicology, this criticism has no direct offspring, but the other extreme, the 'anarchy of knowledge' (Feyerabend 1978), finds parallels in the diverse manifestos of postmodernist musicology, for example Tomlinson's (1984).
\cite	
Tomlinson's \citeyear	

Figure 2: achicago sample

Matt Swift, who wrote achicago (last version 2001), chose a 'form' criterion: the names of his commands follow what it is that the citation needs (the author? the year?), whether or not parentheses should be added (all commands have a `\...NP` version for 'No-Parenthesis'), and so on. The package does not handle pre-citation notes (like the expression 'notably' in the sample) directly, but using these no-parenthesis commands the user can achieve similar effects.

achicago is a full-fledged package with, quite intriguingly, several extra-bibliography elements. Quotations are no longer typeset `\small`, and `\emph` translates not to `\textit` but to `\textsl`. These things can be a little annoying when one is following uses set by someone else (journals, professors, etc.). On the other hand, the BibTeX that accompanies the package (the file `achicago.bst`) is amazingly comprehensive, providing fields for such notions as translator, original title, etc. In his introduction to the package the author enters the discussion of the pros and cons of each family of styles. More about this later.

The wonderful natbib package (by Patrick Daly, last version 2006) is the definitive word on author-year bibliography styles with LaTeX. It builds on the harvard experience and offers a most complete set of customization possibilities. Extra features

(\citep[notably][]) \citeyearpar \citet (\citeauthor...) \citep Tomlinson's \citeyearpar

Some philosophers of science have claimed that science progresses as life (notably Kuhn 1996; Koestler 1959). Kuhn started his conceptual trip with his exploration of the Copernican Revolution (1957). At virtually the same time, at the other side of the Atlantic, the same trip was documented in Koestler (1959). In musicology, Leo Treitler expresses very similar views (1984, 1989, 1999). Kuhn's 'paradigms' were directly addressed by musicians since 1991 (McClary).

There was and is, of course, opposition to Kuhn, whose ideas were always shunned by mainstream philosophy of science. Most virulent of all was the criticism by Imre Lakatos (1970). In musicology, this criticism has no direct offspring, but the other extreme, the 'anarchy of knowledge' (Feyerabend 1978), finds parallels in the diverse manifestos of postmodernist musicology, for example Tomlinson's (1984).

Figure 3: natbib sample

include an easy conversion to bracket labels, a useful system of 'aliases', control over punctuation and capitalization, and continued two-way support with packages like hyperref. The older packages harvard and achicago are very dear to me personally, but for users new to this family of styles I see no reason to recommend any package other than natbib.

The commands in natbib are named somewhat more capriciously than in its predecessors. There is no plain `\cite` (!). Instead of this, `\citep` is intended for parenthetical citations and `\citet` for citations within the text (the ones that would be 'noun' citations). Both commands support two optional arguments, for notes within the parentheses to either side of the citation itself.

4.3 Advantages and disadvantages of author-year

Oren Patashnik (creator of BibTeX, and one who clearly doesn't like author-year labels) has even argued that this citation style "encourages the passive voice and vague writing". With Matt Swift (in his introduction to achicago), I have to say I'm not sure. But there is no denying that the parenthetical labels interrupt the flow of reading. The same reasons that in certain contexts make this style better

than bracket labels — i.e., that the author and the year *are* crucial information in some kinds of argument — can be held actually *against* it. In these contexts, the title is also crucial: suppose you quote someone like Foucault; is this an interview, a popularizing essay, or a rigorous book? What the reader is to do with the citation certainly depends on this. And in that case, the reader is forced to put his finger in the book, and go search the entry in the final reference list.

Or what about different editions of books, or reprints of articles? What is one to do with a citation like (Descartes, 1949)? If information about the publication is important, citations like '[1]' are insufficient, but '(Adorno, 1976)' is insufficient too, and sometimes even misleading.

One might then ask why it is that these styles are so widely standardized today. Well, there is a clear reason for their early appeal: unlike numeric references, and unlike footnotes, a late change to a manuscript does not require going over the whole thing to update numbers and cross references. This was extremely relevant in typewriter, WordStar or WordPerfect times (I imagine — I'm just too young to have experienced it myself!). It may even be relevant today, taking on the risks of stereotyping, with Word users, whose vast majority is not aware that this can be automated… In any case, this 'advantage' is of course rendered meaningless by the computers of today, and in particular by TEX.

In fact, it is a little ironic that further development of computerized document preparation is even turning this advantage of author-year styles into a hindrance: more and more, citations are expected to be interactive hyperlinks. This, today, implies an *enormous* difference between typing (say, with the natbib package)

```
...(notably Tomlinson's [1984]).
```

and typing

```
...(notably Tomlinson's [\citeyear{tomlinson1}]).
```

The first, easy to remember and type, *won't produce a link*. If you want the link, and today you certainly do, you have to use the second — and then the effort of taking care of the punctuation, command sequence, and key, seems a little like… like using a tank to kill a fly.

So, beyond the often unsurmountable institutional pressure — journals, professors, etc. — I really see no reason to use author-year styles. Above all today, that — my main point in this article — software has advanced to a point where all alternatives are equally well supported.

5 Footnote citations

Maybe not for pure mathematics, but in other contexts (certainly including the *history* of mathematics) I would say there is no better option than footnote citation. LaTeX has supported this since 2002, with the appearance of opcit. This package will translate `\cite` into `\footnote` (unless it occurs inside one), and append the information of the reference into the footnote.

The first time a publication is cited, the information will be full: author, title, journal/publisher, address, year, etc. Further citations of the same work, however, will abbreviate the reference into the last name, followed by the traditional '*op. cit.*' (Latin for 'cited work'). Moreover, if the same citation occurs in two successive footnotes, it will simply say '*Idem*' ('same'). The optional argument to `\cite` will be appended after the information (either full or abbreviated), separated by a comma.

opcit provides a starred version `\cite*` that omits the author's name (often redundant in footnotes). On the other hand, if there are several works by the same author, in which case '*op. cit.*' can be ambiguous, a mechanism to assign 'aliases' to the works (the 'hereafter' mechanism) is provided.

6 opcit 2

opcit was written by this author, and its first version dates from 2002. In 2006 I uploaded the second version of the package, with a complete BibTeX style (the first one was very limited). This second version, that owes a lot to comments and suggestions by several users, and in particular those of John Scott, fixes minor problems of the first version, and adds some extra features, notably:

- The ability to omit certain information in the footnotes but not in the final reference list. This can be used to omit an article's page numbers when a '[p. 12]' optional argument follows, or to omit the second part of the title, information on series, original edition dates, etc. — information that is not really needed in the footnotes.
- '*op. cit.*' expressions and other 'aliases' can be hyperlinks to the footnote where the work was first cited.

– Citations can be reset (for example, at the beginning of chapters) so that a post-citation will again cite the information in full.
– Support for cross referencing between entries through BibTeX's special field `crossref`.

6.1 Additions to opcit 2

The second version of opcit has been generally well received and, as far as I can judge, widely used. Some users have already made comments and suggestions, and in two cases they have contributed some pieces of code that fix or improve a couple of opcit's current features. These additions, mentioned in this section, will be included in a third release I'm working on (hopefully for the Summer of 2007), but for the moment they are in beta testing.

6.1.1 Hereafter improved

Eric Rauchway, a devoted "fan" of opcit, wrote to me some months ago about getting the "hereafter" of articles *not* italicized. ("Hereafter" is the user-defined reference to a previously cited source, that replaces the default *op. cit.*. It is useful when there are citations of several works by the same author. It is desirable that articles' hereafters are not italicized, while those of books are.) He and his friend Kevin Bryant have found a solution to this, and I will include their find in a following release. The solution involves modified versions of both `opcit.sty` and `opcit.bst`. If interested, please write to me (`federook@gmail.com`) to get the modified files.

6.1.2 Name-swapping

The second release of opcit swaps the first and last name of authors for the final reference list (so that the footnote says "Ernest Gellner", but the final list says "Gellner, Ernest"). However, in some cases this it is desirable to keep the regular order: for example for Dante Alighieri. (Also, there is a problem when the author is Aristotle, since opcit doesn't really know what to swap, and puts a spurious floating comma.) Patrick Gardner contributed the following solution "which might be of use to others who are using opcit for ancient and medieval authors" (like he is himself):

```
"\newBibCommand{\SwapNames[2]}{#1 #2}%
{#2\ifx\@empty#1\else, #1\fi}"
write$ newline$
```

This should replace line 946 (the `begin.bib` function) of `opcit.bst`. With this, opcit will handle "Aristotle" correctly, and then putting between braces the *full* name "Dante Alighieri" in the `.bib` file will prevent name swapping.

6.2 The future of opcit

The main problem still facing opcit is a very hard-to-understand (for me, anyway) conflict with endnote, the package that collects the notes to be printed at the end of the document/chapter. It really would be nice to be able to turn endnotes on and off without further changes. (The fascinating discussion on footnotes-or-endnotes resembles that of the bibliography styles in that the opposing sides really hate each other; again, both have good arguments to their cause, but 'the truth' probably lies in a context-dependent approach.) I succeeded once in creating a list of endnotes from opcit footnotes, but the solution was far from robust, and did not really throw light on how to address the problem.

On the other hand, there are ideas and work going on regarding other compatibility issues of opcit. With the release of the second version, the package secures LaTeX support for footnote-style bibliography… *in English*. But use with other languages is not directly implemented. This not only requires the modification of the BibTeX style (so that particles like 'in', 'chapter', etc. are translated), but also might bring about problems with babel. For example, José Luis Rivera from Mexico has identified conflicts with the latter's spanish option, and has started working on complementing opcit with a Spanish BibTeX style, which possibly involves some tweaking to opcit itself.

The implementation of opcit in languages other than English will hopefully involve other users as well, and is, as I see it, the most important future extension of the package.

7 Other important things to mention

7.1 Some hybrid approaches

For the sake of completeness, a couple of packages should be mentioned that provide a kind of 'bridge' between the three main families of styles:

alpha was already mentioned to be a compromise between labels like '[1]' and labels like '(Cassirer, 1944)': it gives '[Cas44]'. See page 4. In the same vein, natbib has the option of typesetting labels in either of the two forms (and also as superscripts).

cite makes bracket labels appear as superscripts, almost as footnote marks (although between [and] and without an actual footnote). The package (which also has other nice features) is extremely sophisticated, but has almost no documentation (it dates from before the doc package for LaTeX documentation). As a result, it has come to be, in effect, obsolete. Even Sebastian Rahtz, when trying to provide support for it in hyperref, had to give up trying to understand it.

footbib goes one step further than cite: the superscripted labels do actually point to a footnote. However, it is not a footnote in the full sense: it follows its own numbering, and in case there are also 'regular' footnotes in the page, the two sets are separated from each other.

7.2 custom-bib

This topic is not directly related to the thesis of this article, but it does seem odd to omit it from a general discussion of the possibilities of bibliography in LaTeX. The fact is that, for some reason or other, I myself have not actually used custom-bib, and in fact learned about it relatively recently. Thus I have not been able to incorporate it into my musings about bibliography. But that it is an important thing to mention there can be no doubt.

custom-bib — latest version dated April 27th of the current year — is another wonderfully ingenious TeX program Patrick W. Daly (the author of natbib) that helps the user create a totally customized BibTeX style (i.e., a `.bst` file). Here we

are back to the normal meaning of 'style': the set of rules that govern the appearance of the entries in the final reference list — whether the title is italicized, the journal number typeset in boldface, etc.

The package works in a straightforward way. Once there is a `makebst.tex` file in the system — you might have to create it first by running TEX on the file `makebst.ins` — the user runs TEX or LATEX on it:

```
latex makebst.tex
```

Then the program will simply *ask* (!) how you want to format your entries, and from it create a BIBTEX style. It is a truly amazing use of TEX's interactive capabilities, which are usually overlooked (since interactivity is not exactly what document preparation is about, after all).

The package is tailored towards the first two style-families described above: brackets (called 'numerical' in custom-bib) and author-year. Use for opcit, I anticipate, would require some extra hacking on the `.djb` file (an intermediate step between `makebst.tex` and the final `.bst` file), but in principle the main difficulties here would arise from the lack of documentation in opcit about custom-bib, and maybe the other way around as well. That is, opcit includes some directions on how to customize the `.bst` file, but these directions assume familiarity with BIBTEX's programming language — and this familiarity is precisely what custom-bib is supposed to spare the user.

In any case, as mentioned, I am not the most qualified to enter into custom-bib matters, and the package is mentioned here as the wonderful tool it is for deeper-than-surface bibliography handling in LATEX.

8 Conclusions

I have to finish by pointing out some facts that have come to my attention since I presented a version of this article at PracTEX 2006. For example, José Luis Rivera told me that the MLA (*Modern Languages Association*, which is in effect the main legislator (and champion) of author-year styles, has indeed addressed the issue of some funny things like citing (Aristotle, ca. -340). They allow a variation of style that can be called 'author-title': (Aristotle, *Nicomachaean Ethics*). Even for modern authors, like in (Derrida, Postcards), this has been accepted and even encouraged. This certainly is a response to the problem that often the title of a work is more crucial than the year.

The discussion can go on and on (is it the title itself, or the fact that the work is so well-known? how does this depend, once again, on context?), but one thing that has to be said is that, as far as I know, LaTeX has not seen any direct efforts in this direction. The MLA, certainly, has adopted this relatively recently (mid-90s is José Luis's recollection).

I cannot claim that the discussion in the previous pages is comprehensive or complete. However, I hope that the points raised above are not obvious or trivial (they weren't to me when I started thinking of all this), and feel that the topic is interesting, if nothing else because it shows how, here too, one little change or decision leads to more and more. In fact, people who have seen drafts of this article tend to respond rather quickly and very 'personally' (as in "personally, I hate footnotes", or "I *totally* agree with this or that…"). The topic, touching on uses that habit has ingrained to the point that they enter the realm of taste, seems to reach everybody and raise very deep opinions in them. So, besides the arguably 'useful' fact that this article might make readers aware of possibilities that were previously unknown to them, I hope it has also provided some enjoyment.

THE PracTEX Journal TEX USERS GROUP

Demystifying LaTeX bibliographies

S. Parthasarathy

- PDF version of paper
- Comment on this paper
- Send submission idea to editor

In this essay, we will try to explore and explain the vexed problem of including bibliographic references in LaTeX documents (reports, papers, theses etc.). There is a huge plethora of literature on this subject. Infortunately, these materials are focused on LaTeX's experts, which is driven to a developer's point of view level. This current paper will examine bibliographies for common user's point of view, trying to pass by only the essentials of this very vast and involved bibliographic topic. The author hopes that this paper will make LaTeX, enjoyable for more people.

Parthasarathy teaches discrete mathematics to undergraduate Computer Science students, at Hyderabad, India. He has branched into full-time teaching after a 25-year stint in the software industry. His association with LaTeX started in 1993, at the United Nations University, International Institute of Software Technology, Macau, where he worked with Prof. Dines Bjorner. His website http://algolog.tripod.com/nupartha.htm will give more specific details about him.

For submission to The PracTEX Journal
Draft of June 6, 2007

Demystifying LaTeX bibliographies. *

S. Parthasarathy†
drpartha@gmail.com

Abstract In this essay, we will try to explore and explain the vexed problem of including bibliographic references in LaTeX documents (reports, papers, theses etc.). There is a huge plethora of literature on this subject. Infortunately, these materials are focused on LaTeX's experts, which is driven to a developer's point of view level. This current paper will examine bibliographies for common user's point of view, trying to pass by only the essentials of this very vast and involved bibliographic topic. The author hopes that this paper will make LaTeX, enjoyable for more people.

1 Preamble

It is a common practice to refer to various publications when writing a paper, or a report, on a complex subject. LaTeX provides very elegant tools for including details of external publications which you have cited in your main paper. Unfortunately, there is not much material which makes it easy to understand how all this works. This current paper is based on the experiences of a person, who has gone through the pains of understanding how Bibliography works under LaTeX. To make it easier for reader to learn the concepts, this paper has made some simplifications. For a more rigorous and deep research, you must refer to the LaTeX book [lam-1] by Lamport or the Companion book [goo-1] by Goossens. There are some people that say "Lamport wrote the Bible, and Goossens *et al* wrote the Gospel". There is also an excellent tutorial available on the w-w-web [nic-1] with a very detailed review of the concepts behind LaTeX bibliographies. It also gives some very good and practical tips and lots of examples.

*Ask the author for a copy of the LaTeX source of this document (demystify.tex) . You can try out all the examples yourself, by hacking the source.

†Algologic Research and Solutions, 78 Sancharpuri Colony, Bowenpally, Secunderabad 500 011, India WWW-URL – http://algolog.tripod.com/nupartha.htm

2 Terminology and basics

In this section, we introduce some basic terms with a special meaning. For example:

Main document : the document in which you are going to include (cite) external documents;

External document : the material cited in the main document. We are primarily interested in publication details about the external document, and not the contents itself;

Bibliographic list : a consolidated list of external documents. A list gives you essential details about each external document cited in the main document. Also called by "shopping list", it is usually generated and included at the end of the main document, and printed (or rendered) along with the main document, as a separate section. Sometimes, the section is titled as **References**, or **Bibliography**, depending on your document style. If you wish, it is possible to replace the title **References** or **Bibliography** by any title of your choice, using the `\renewcommand` command, but this is other topic to be written;

You will also need a "bibliographic style" file (.bst), which is responsible to format the bibliographic list displayed in the document. The style specifies the sequence/ordering in which details will be given for each external document cited by you,such as font style and size and so one. The bibliography style may be either a standard one defined by editors, or a specific style was defined by you;

Bibliographic database: the bibliographic list, rich of details, to be cited in your main document. It can be grouped and located as:

1. **Built-in :** in the case of `\begin{thebibliography}...\end{thebibliography}` environment, the database is part of the main document itself.
2. **External :** a external database file has the publication details of various external documents (e.g. BibTeX file (.bib)).

Each external document (identified by a `\bibitem` command) occupies one

record of the database. Each record can be composed of several fields, such as:

1. Name(s) of the author(s);
2. Title;
3. Journal, book, conference, or other location where the document was published;
4. Volume number, issue number etc.;
5. Place of publication;
6. Name of publisher;
7. Date of publication.

This current paper shows you a simple way to switch between these two options.

3 Choose the right tool

The first thing that we must remember is that, it is not always necessary (or justified) to use an elaborate tool in order to create a Reference/Bibliography list. You can always make-do with :

1. Footnotes;
2. Enumerated lists, combined with `\ref & \label`.

Maybe, these methods can be clumsy, and cumbersome at times, but they are down-to-earth, and need no special skills. However, there are situations when a more sophisticated method is necessary. To this case, two alternatives are available:

1. **Built-in bibliography**, using the `\begin{thebibliography}...\end{thebibliography}` environment.
   The `\begin{thebibliography}...\end{thebibliography}` environment is simpler. It is part of the main document itself. In this approach, you will have to retype, for each main document, all details of all the publications, cited by you in the main document.

2. **External bibliographic database** (usually, BibTeX).
   This is an external database (given usually as a .bib file). It is much more elaborated than the `\begin{thebibliography}...\end{thebibliography}` option. The benefits are that you do not need to retype details of all the publications, whenever you use the data from this database and share it. In addition, you can use more than one external database (in the same main document), if you wish. The bibliographic list includes details of all the documents cited by you (even if they are stored in different databases). This feature helps you to avoid huge monolithic databases. You can create smaller (and more manageable) databases, grouped by theme, or grouped in some other convenient way.

   Sometimes, you would like to list out a document in the bibliographic list, even if you have not cited it explicitly in the main document. You can fake a `\cite` using the `\nocite` command (please visit Section 6 for more details).

This paper uses the built-in bibliography `\begin{thebibliography}...\end{thebibliography}`, to illustrate the actual use of bibliography tools. However, this paper will examine both the above tools. For the case of special bibliographic database (BibTeX), you will have to use a different set of files (horse.tex, zebra.bib) as example. You can get these two files by sending an email to the author.

4 Built-in bibliography

Use the `\begin{thebibliography}....\end{thebibliography}` environment if you have a (.bib) file but wish to transport a final but compilable version or to develop a particular and small "dataset" inside the main document. This flexibility permits you to pack only the necessary references from the whole database (.bib). This is the easiest way to include bibliography in your main document. In fact, as D Venugopal[1], a remarkably mature user of LaTeX usually says:

> If you are not a serious researcher, and write only one or two research papers in your entire life, then use the

1. Duvvuri Venugopal, Banares Hindu University, Benares, India, (venugopal_duvvuri@rediffmail.com)

`\begin{thebibliography}....\end{thebibliography}` environment. Otherwise it is best to create a .bib file.

Your main document will look like this :

```
\documentclass{someclassname}
Preamble stuff goes here
\begin{document}
The contents of your main document go here.
You can cite  an external document like this:
In \cite{paper1} Partha has given a long lecture
about LaTeX. And in \cite{paper2} he gives examples
about how to use LaTeX. In \cite{shake3},
Mark Anthony said ‘‘Friends, Romans,
Countrymen...lend me your ears’’.
```

If you use, the `\begin{thebibliography}...\end{thebibliography}` environment (like this paper), details about the above three external documents must be listed in the `\begin{thebibliography}...\end{thebibliography}` environment (as part of your main document itself), like this :

```
\begin{thebibliography}{abc}
\bibitem{paper1}Authornameofpaper1, and other details
\bibitem{paper2}Authornameofpaper2, and other details
\bibitem{shake3} William Shakespeare, Julius Caesar,
Pub.: Shakespeare Press,
Stratford-upon-Avon (UK), July 1623.
\end{thebibliography}
\end{document}
```

Run LaTeX twice on this main document, and you will get your final documentincluding the bibliography, listed at the end of the document (bibliographic list). In the current paper, the bibliographic list is shown as **References**, at the end of the paper. You can also take a look at the source code to understand how it `\thebibliography` works.

5 External bibliography

LaTeX provides a standardised way of building an external bibliographic database. Unfortunately, this method is not frequently used. Do not get discouraged if you do not understand BibTeX in your first run. BibTeX is a complex subject, i.e., a rich approach. Even the name is confusing! For instance, some authros refer BibTeX as a database file, a database format and also the name of the program that is used for processing the datasets included in a BibTeX database. It has many variants and options. Fortunately for us, BibTeX has been explained in this paper by simple concepts to be used for.

When you use BibTeX, you will need :

1. A method of specifying where the bibliographic data is stored (a path).
2. A method of specifying how the data will be displayed in the bibliographic list (a style stored in a .bst file).

In our example for the use of BibTeX, the main document is called horse.tex (horse.tex is similar to the current demystify.tex, except that the built-in `\thebibliography` environment was replaced by the external bibliographic database zebra.bib). Let us create and use a BibTeX database called zebra.bib (BibTeX files always have the extension .bib)[2]. Each record of the database will have an identifier key (used by the `\cite`), followed by the fields. Each field of each record has a name + value pair. The name and the value are separated by an = mark. The name/value pair is used by LaTeX to decide the style in which the bibliographic list will be composed (sequence/ordering of fields, fonts style/size etc.). Below, an example of a record from the horse.bib file is shown:

```
@BOOK{goossens1,
   author = "M Goossens and F Mittelbach and A Samarin",
   title = "The \LaTeX\ Companion",
   publisher = "Addison-Wesley",
   year = 1984
   }
```

2. You can get horse.tex and zebra.bib from the author

Notice that the author names are connected by "and". You may get a lot of warnings when you run LaTeX on horse.tex for the first time. So, LaTeX must be run two more (three times). The warnings will disappear. Notice also, that the .bib file does not have any `\bibitem`. Actually, LaTeX generates a new file with the extension .bbl where it puts the data in a structured list-like form, using the `\bibitem` (similar to `\item` in a list).

You will also notice that BibTeX files have a very cumbersome syntax. There are special tools available for creating BibTeX files e.g. Bibdb[3], tkbib[4] or JabRef[5]. But, the price you pay, is that you have to spend time, to learn using these tools. Creating all your bibliography data as a BibTeX file would be necessarily painful. So, why bother at all ? There are many reasons :

1. BibTeX is a standardised way of collecting bibliographic data. Many organisations and publishers use this standard. It makes sense to follow such a widely accepted standard;
2. You can cite references from more than one BibTeX file inside your main document. Imagine, you have two BibTeX files : zebra.bib and tiger.bib. You can decide to use the references from both these databases, into the same document, like this : `\bibliography{zebra,tiger}` Of course, the citekeys you use in `\cite` must uniquely belong to one of the many BibTeX files you have selected in the `\bibliography` command;
3. You can reuse the bibliographic data, in more than one document. Imagine, you have created two main documents : horse.tex and mule.tex. In both these documents, you can use the data from the same database using e.g. `\bibliography{zebra}`.

Imagine a research team working on a special project. All the references and bibliographies used by the team, can be assembled in one BibTeX file. In a school, all scholar researchers can share the same BibTeX file and cite references from the common bibliography, in their individual papers. The students will not have to collect and type out the bibliographic details each time.

3. Bibdbit is downlodable at http://www.mackichan.com
4. tkbib is downlodable at http://vlsicad.cs.binghamton.edu/~pmadden/tkbib
5. JabRef is downlodable at http://jabref.sourceforge.net/

Now, imagine all research teams in the same institution, but working on different subjects, they have their own BibTeX files, for their own subject. The Director of the institution can draw references from all these bibliographies, when he writes a paper giving an overview of the institution and all the research activities of the institution. He will not have to collect, and compile, a bibliography of his own. BibTeX comes in as a handy database tool in such cases.

Usually, a researcher publishes his work in different journals and conferences. Each journal/conference has its own standards for presenting the bibliography list. The sequence in which the records are sorted, the sequence in which the fields of each record are presented, the formatting style for presenting each field, is specified by the journal/conference. Using a standard base like BibTeX, you can easily switch between various presentation styles, just by changing the `\bibliographystyle` command.
In all these case, the effort of creating a database using the clumsy-looking syntax of BibTeX becomes worthwhile.

6 Uncited citations

Now, what does that mean ? Maybe you want to refer a work from the .bib file but you do not actually `\cite` in the text. So, you use the `\nocite{MOZART}` command in the .tex source. As a consequence, the reference labeled by MOZART will appear at the bibliography list, although there is no place in the rendered main text where you have cited it. The `\nocite{MOZART}` document is a kind of orphan, who exists in the bibliography list (orphanage), but has no home (in the body of the main text). To give you an practical example, there is a book of Don Knuth in the bibliography list of the horse.tex file, but it was not mentioned anywhere in the body of the main document. You will need this artefact, if you want to give a list of useful reference documents, but have no specific reason to cite them in the body of the main document.

7 Closing remarks

LaTeX produces very elegant documents. It gives you tools for develop your documents. In addition, several facilities to cite bibliographies inside your main text. These facilities are useful for storing and presenting your bibliographic data in a standardised way (and reusing them, if necessary).
Note : The following portion gives you the thebibliography database in two forms. You can block any of the these forms by using a `verbatim` environment as an envelope. You will see the citations shown as plain numbers or as small texts (in the rendered version) , depending on which portion you decide to leave unblocked. The part which is blocked using `verbatim` will show up as unordered code, in the rendered version, like this :

```
\begin{thebibliography}{ww}
\bibitem{lampo1}L. Lamport, LaTeX : A document preparation system,
Pub.: Addison-Wesley, 1986 (The LaTeX bible)
\bibitem{goossens1} M Goossens, F Mittelbach, A Samarin,
The \LaTeX\ Companion, Pub.: Addison Wesley, 1994. (The LaTeX gospel)
\bibitem{beast1}Nicolas Markey, Tame the beast,
URL http://
http://tug.ctan.org/tex-archive/info/BibTeX/tamethebeast/ttb_en.pdf
\end{thebibliography}
```

Bibliography management is a very crucial part of the work of people who desire to publish technical papers and reports regularly. LaTeX provides convenient tools for bibliography management. This paper has given an overview of the concepts involved. In particular, we have seen some fundamental details about BibTeX. BibTeX is a very rich and sophisticated tool. The reader must study the references given at the end of this paper, to fully master all the concepts involved.

8 Acknowledgements

The author is grateful to his **invisible gurus** Mr. D Venugopal, and Mr. Francisco Reinaldo (Production Editor, PracTeX Journal) who helped the author to substantially improve the clarity and correctness of this paper.

References

[lam-1] L. Lamport, LaTeX : A document preparation system, Pub.: Addison Wesly, 1986 (The LaTeX bible)

[goo-1] M Goossens, F Mittelbach, A Samarin, The LaTeX Companion, Pub.: Addison Wesley, 1994. (The LaTeX gospel)

[nic-1] Nicolas Markey, Tame the beast, URL
`http://tug.ctan.org/tex-archive/info/BibTeX/tamethebeast/ttb_en.pdf`

* * * * *

Journal home page
General information

Submit an item
Download style files
Copyright
Contact us

THE PracTEX Journal

Travels in TeX Land: Trying ConTeXt

David Walden

Abstract

In this column in each issue I muse on my wanderings around the TeX world. In this particular column I focus on my initial efforts to learn and use ConTeXt.

Dave Walden is retired after a career as an engineer, engineering manager, and general manager involved with research and development of computer and other high tech systems. More history is at www.walden-family.com/dave. He may be contacted at dave@walden-family.com

- PDF version of paper
- eic.tex
- eic.pdf
- walden.tex
- Comment on this paper
- Send submission idea to editor

The PracTEX Journal, 2007, No. 2
Article revision of 2007–06–05

Travels in TEX Land: Trying ConTEXt

David Walden

ABSTRACT: *In this column in each issue I muse on my wanderings around the TEX world. In this particular column I mainly focus on my initial efforts to learn and use ConTEXt.*

1 The theme of this issue of *TPJ*

In three columns over the past 15 months, I described methods and tools I use to make it easier to draft large documentation projects (i.e., books) and my experience with self-publishing the book *Breakthrough Management*. I believe these columns are essentially on the theme of this issue (TEX for editors), and I refer you to those columns:[1]

- **LATEX for Productivity in Book Writing** (issue 2006-2)
- **Experiences Refining Page Layout for a Book** (issue 2006-3)
- **The post-typesetting phase of producing a book** (issue 2007-1)

By the way, since the last of these three columns, I corrected dozens of typos and other errors that were found in the first two printings of the book. All of these changes were only a few letters or a few words that dealt only with the errors; no change was made to the substance of the book. Given how much I have heard about how a small change to a TEX file can change the typesetting of many pages, I was pleasantly surprised to discover that almost all of the pagination remained unchanged. In one place in this 280 page book I had to pull an extra line forward a page to maintain the line balance on facing pages, and in one other place I had to change a word to avoid a line overflow.

Thus, in this column I am going to report on something other than the theme of this issue, namely, on my first attempt to use ConTEXt.

2 Trying ConTEXt

I have known about ConTEXt for the past several years, but never tried it out. I got more interested in ConTEXt a couple of years ago doing **my interview of Hans Hagen**, the creator of ConTEXt but still did not try it. After typesetting my book with LATEX last year (see my columns mentioned in section 1), I got more interested in ConTEXt, particularly its ability to typeset on a grid that Hans had mentioned.[2] Consequently, after putting my column for

[1] All of the blue (on a display or color printer) words in the following are hot links to various URLs.

TPJ issue 2007-1 to bed, I decided to try ConTEXt, and I describe that experience in this and following sections.

But first, I must apologize for my hit-and-miss approach to finding out things about ConTEXt (as described below) rather than systematically reading a manual. However, this tends to be my approach to learning something new (albeit perhaps quite inefficient). I suspect that many other people approach a new system in a similarly unsystematic way; perhaps by reading my experience they will be encouraged to try their own ad hoc path to facility with ConTEXt.

I started by printing Steve Peter's two columns on ConTEXt in the 2005 issues of *TPJ*:

- `starttext % Practical ConTeXt`
- `starttext % Practical ConTeXt:` **ConTeXt Text Editors**

Then I executed Steve's "hello world" example.

Next, I looked at Joe Hogg's paper in *TPJ* issue 2006-3:

- **ConTEXt Starters**

I printed out his example letter and looked at his ConTEXt file for the letter.[3] I created a skeleton letter from those examples and cut-and-pasted the text of a letter I had drafted in Word into the body of my skeleton ConTEXt letter. I looked up a few needed commands in Han Hagen's 370-page **ConTEXt manual** (which I had printed out a year ago but not looked at since). For instance, I found the following pair of commands in the manual:

```
\startitemize ... \stopitemize
```

Thus, I was able to do basic formatting of a four page letter. (An expurgated version of the source text file for the letter is available as `eic.tex` on the HTML page for this paper, along with the resulting output (`eic.pdf`). I make no claims that the formatting of this letter is good quality ConTEXt markup; it is only what I was quickly able to pull together.) That part seemed easy enough, taking a couple of hours not counting refining the content of the letter. Of course, already knowing TEX cut out lots of learning someone would have to do who started from scratch with ConTEXt.

This first try of mine of ConTEXt had defaulted to producing dvi output which I converted to a PDF by clicking the dvi-to-pdf icon in WinEdt, the editor I use. So I tried looking at the ConTEXt Garden, the **wiki for ConTEXt**, to which Steve Peter's paper had directed me, but I didn't find anything there immediately (although it looked like there was a lot of good stuff there such that I should keep this website in mind for answers to future questions). I next opened the **TeXEXEC explained** manual I had printed out at the same time as the ConTEXt manual a year ago, and there on the second page it told me about including the PDF switch in my command line to compile a ConTEXt file:

```
texexec --pdf file-name
```

[2] Also, the **LuaTEX** efforts, which seem to be an important center of future TEX evolution, involve Hans Hagen and other people deeply involved with ConTEXt. Thus, it seems likely that eventually I will be drawn into these guys' orbit, e.g., to obtain the benefits of LuaTEX when it is available, an my curiosity would probably also lead to me trying their ConTEXt system; so I might as well start getting familiar with ConTEX now rather than later.

[3] Also included with Joe's paper is an example of the ConTEXt for a four-page brochure.

I decided I wanted some hot links for URLs in my letter, so I looked up a ConTEXt source file of Joe Hogg's example letter that appeared as part of his contribution to issue 2006-3 of *TPJ* (`http://www.tug.org/pracjourn/2006-3/hogg/`) and saw how he defined hot links

```
% Enable colors and activate hyperlinks
\setupcolors [state=start] \setupinteraction [state=start, color=blue]
\definecolor [darkRed][r=.5,g=.0,b=.0]
\useURL[JCQM][{http://www.cqm.org/journal}][]
             [{\it Journal of the Center for Quality of Management}]
```

and later referred to them with

```
\from[JCQM]
```

Then I tried to change the font and could not figure out how to do it. I subscribed to the ConTEXt discussion group (`ntg-context@ntg.nl`) but (through no fault of the discussion group) didn't get any farther very quickly. So I gave up. I decided to come back to ConTEXt sometime later when I had a substantial project that required ConTEXt and thus it would be worth spending the time to figure out how to configure ConTEXt properly.

An aside: I wish I could focus on the complexity of the typesetting system (e.g., ConTEXt, as in this column) or the new programming language or whatever else I am trying to learn rather than on configuring the application software package to run on my computer system. This is a frequent bother with using open source or free software. Of course, commercial software also often has similar problems, e.g., having to learn a development environment when all I want to do is use the compiler, editor, or word processor. In all cases I long for software packages that just work without complicated downloading, installation, or configuration; I don't really mind complexity, but I wish I only had to deal with it in the area I am actually trying to work in.

3 Back to ConTEXt and fonts

A month later, I fumbled around for a few minutes at a time on and off for a couple of more days, until I found Bill McClain's ConTEXt beginners page and its section on **Selecting Fonts**. I used his simplest method

```
\font\myfirstfont=bchb8r
\myfirstfont
```

to switch my earlier letter into CharterBT-Bold which for the first time gave me some comfort that my ConTEXt configuration was not broken. (When one tries something for the first time and it doesn't work, it's hard to know whether it is one's own error and lack of understanding or if the never-tried-before system doesn't work right.)

I next tried the following by McClain which he calls "A simple example." He says he has the following fonts on his system:

```
bchr8r  roman
bchb8r  bold
```

```
bchri8r italic
bchbi8r bold italic
```

So, he created the following typescript file, named `type-charter.tex`:

```
\usetypescriptfile[type-buy]

\starttypescript [serif] [charter] [8r]
  \definefontsynonym [Charter-Roman]       [bchr8r]  [encoding=8r]
  \definefontsynonym [Charter-Bold]        [bchb8r]  [encoding=8r]
  \definefontsynonym [Charter-Italic]      [bchri8r] [encoding=8r]
  \definefontsynonym [Charter-Bold-Italic] [bchbi8r] [encoding=8r]
\stoptypescript

\starttypescript [serif] [charter] [name]
  \usetypescript[serif][fallback]
  \definefontsynonym [Serif]           [Charter-Roman]
  \definefontsynonym [SerifBold]       [Charter-Bold]
  \definefontsynonym [SerifItalic]     [Charter-Italic]
  \definefontsynonym [SerifBoldItalic] [Charter-Bold-Italic]
\stoptypescript

\starttypescript [Charter]
  \definetypeface [MyCharter] [rm] [serif] [charter] [default] [encoding=8r]
\stoptypescript
```

And he called it like this:

```
\usetypescriptfile[type-charter]
\usetypescript[Charter]
\setupbodyfont[MyCharter]
```

I did that and it worked for me too. Apparently I also have the bchr8r, bchb8r, bchri8r, and bchbi8r fonts on my system.[4]

Nonetheless, I put aside figuring out more deeply how to use fonts in ConTEXt. I'll come back to that after I know a little more about ConTEXt in general. In the meantime, I have printed out copies of two papers from issue 2005-2 of this journal that relate to using various fonts with ConTEXt, for when I am sure I will need them later:

- **Integrating TrueType Fonts into ConTEXt** by Thomas Schmitz
- **OpenType installation basics for ConTEXt** by Adam T. Lindsay

4 Writing a non-trivial document in ConTEXt

To get more experience with ConTEXt, I decided to try to write this column in ConTEXt. The

[4] Yuri Robbers says that these fonts come with almost any TEX installation—I live and learn.

final source file is listed on the HTML page for this paper with the title `walden.tex`; this PDF file you are reading is the output. I don't claim to use good ConTEXt style (yet).

I had some notes for this column (although nothing yet for this section) in a plain text file that had a few LaTeX commands mixed in with a lot of text without required markup. I changed the extension of this file from `.txt` to `.tex` and compiled the file over and over with ConTEXt, looking up stuff on ConTEXt as needed until the file completely compiled. For instance, I found `\starttyping` and `\stoptyping` for displaying verbatim commands by Googling on "Context verbatim" which led me to the **Verbatim text** page of the ConTEXt Garden wiki.

However, after an hour of such hit and miss looking up of needed commands, I decided I needed to actually read (or at least skim over) one full document on ConTEXt; and I did thumb through and glance at every page of **ConTEXt an excursion**.

In the "excursion" I found:

- `\subject` as an alternative for `\section` when no section number is wanted, as for the Acknowledgments and Biography sections of this paper

- that by commenting out the command `\setupwhitespace[medium]`, which I had used in my letter from which I started formatting this column, I eliminated the between paragraph vertical spaces the letter used by that were inappropriate for this column

- the commands

  ```
  \startnarrower ... \stopnarrower
  ```

  for having the paper's abstract be narrower than the full text block size

- that the `\footnote` command looked the same as in LaTeX (when used without ConTEXt's optional label field)

At this point, I decided to try to match somewhat the standard LaTeX formatting for *TPJ* papers. I put the journal issue number at the top left of the first page with the revision date below it. I switched the title and author lines from `\midaligned` to `\leftaligned` and made them bigger using ConTEXt's `\tfb` and `\tfc` type-sizing commands. I was also excited to find that the command `\currentdate[year,--,mm,--,dd]` results in a date in the format 2007–06–05. (Just changing the argument of the command results in a different format, e.g., `\currentdate[day,month,year]` results in 5 June 2007.) In general lots of ConTEXt basic commands seem to have more useful optional formats than LaTeX's basic commands have, options that one often has to load a separate package to obtain in LaTeX. For instance, `\section{Title words}` has an optional argument, e.g., `\section[secfoo]{Title words}`, such that the section can be cross-referenced with as in `Section~\in{secfoo}`.

Then it occurred to me that perhaps there were documents somewhere that helped LaTeX users (like me) switch to ConTEXt. I Googled around and found two resources:

- A web page at the ConTEXt Garden titled **From LaTeX to ConTeXt**
- A document by Berend de Boer titled **LaTeX in proper ConTEXt**[5]

[5] A version of this document was also published in *MAPS* 24, 2000, pp. 65–92—an issue of the journal of the Dutch TEX Users Group (NTG).

In the second of these, I found the version of `\startitemize` that uses numbers rather than bullets, such as for the list in Section 7 of this column. I should have searched earlier for these documents.

Although my column doesn't usually involve images, I wanted to try one in ConTEXt. In the ConTEXt manual I found the following commands, which I used for the bio section at the end of this column:

```
\placefigure[left]{none}{\externalfigure[dave-head.pdf]}
```

So now I have this column in its current state (without what follows this section). It wasn't so hard—only a few hours once I settled into working on it.

5 Trying a bit of math

My writing typically includes almost no math. However, I wanted to try math in ConTEXt, and on pages 1–2 of my **unpublished note on the history of dynamic routing algorithms** (written using Peter Wilson's Memoir class) is the following LATEX code for a bit of math:

```
\begin{eqnarray}
&x_{i} + l_{ij} \ge x_{j}&\\
&x_{0} = 0&\\\
&\mbox{maximize}\qd{}x_{n}&
\end{eqnarray}

\noindent{}where $x_{i}$ is the current approximation of the distance
from the source node 0 to destination $i$ and $l_{ij}$ is the length
from node $i$ to node $j$.  He next turns inequality 1 above into the
inequality,

\begin{equation}
x_{i} - x_{j} \ge l_{ji}
\end{equation}
```

To typeset that in ConTEXt, I changed it as follows, using some commands for formulas that I found in **ConTEXt an excursion**:

```
\def\MYformula#1{\placeformula \startformula #1 \stopformula}
\setupformulae[location=left]
\placeformula \startformula x_{i} + l_{ij} \ge x_{j} \stopformula
\vskip -.3in
\placeformula \startformula x_{0} = 0 \stopformula
\vskip -.3in
\placeformula \startformula \mbox{maximize}\ x_{n} \stopformula

\noindent{}where $x_{i}$ is the current approximation of the distance
from the source node 0 to destination $i$ and $l_{ij}$ is the length
```

```
from node $i$ to node $j$.  He next turns inequality 1 above into the
inequality,

\placeformula
\startformula
x_{i} - x_{j} \ge l_{ji}
\stopformula
```

That is pretty brute force, and there is undoubtedly a better way to handle three equations in a row and make them be single spaced. Or maybe math typesetting is a little harder in ConTEXt than in LaTeX; I'll have to do some more studying to find out. In any case, my little bit of ConTEXt for math produced the following display relating to the Bellman-Ford algorithm:

$$x_i + l_{ij} \geq x_j \tag{1}$$

$$x_0 = 0 \tag{2}$$

$$\text{maximize } x_n \tag{3}$$

where x_i is the current approximation of the distance from the source node 0 to destination i and l_{ij} is the length from node i to node j. He next turns inequality 1 above into the inequality,

$$x_i - x_j \geq l_{ji} \tag{4}$$

6 ConTEXt and MetaPost

I had tried writing and calling a trivial MetaPost program a number of years ago. While I managed eventually to draw a circle or whatever, I found the process of naming the MetaPost output file, calling the file from LaTeX, etc., quite confusing, and never tried again. . . until now.

Somewhere I had read that MetaPost was the native graphics capability of ConTEXt and was highly integrated with ConTEXt. So I decided to give MetaPost another trivial try to see if it was more natural to use MetaPost from within ConTEXt. I Googled on "MetaPost Context" and found the **MetaFun** page of the ConTEXt Garden wiki. I clicked there on the **MetaFun Manual**.[6]

Somehow I quickly stumbled onto page 323 of the Hans Hagen's 364-page MetaFun Manual where I saw the following:

```
\startbuffer[dummy]
draw fullcircle
xscaled 3cm yscaled 2cm
rotatedaround(origin,30)
withcolor .625red ;
\stopbuffer
```

[6] A manual of MetaPost macros that also describes MetaPost in considerable detail.

```
\startlinecorrection[blank]
\processMPbuffer[dummy]
\stoplinecorrection
```

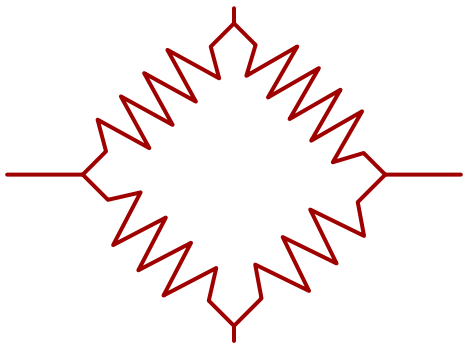
That seemed pretty straight forward, so I tried it and it worked. Wonderful! So much easier already than my previous attempt to use MetaPost.

I looked for a slightly fancier bit of MetaPost code and found some on pages 257–258 of the manual:

```
def spring (expr a, b, w, h, n) =
( ( (0,0) -- (0,h) --
for i=1 upto n-1: (if odd(i) : - fi w/2,i+h) -- endfor
(0,n+h) -- (0,n+2h) )
yscaled ((xpart (b-a) ++ ypart (b-a))/(n+2h))
rotatedaround(origin,-90+angle(b-a))
shifted a )
enddef ;

z1 = (+2cm,0) ; z2 = (0,+2cm) ;
z3 = (-2cm,0) ; z4 = (0,-2cm) ;
pickup pencircle scaled 1.5pt ;
drawoptions (withcolor .625red) ;
draw spring (z1, z2, .75cm, 2, 10) ; draw z1 -- 1.5 z1 ;
draw spring (z2, z3, .75cm, 2, 9) ; draw z2 -- 1.1 z2 ;
draw spring (z3, z4, .75cm, 2, 8) ; draw z3 -- 1.5 z3 ;
draw spring (z4, z1, .75cm, 2, 7) ; draw z4 -- 1.1 z4 ;
```

I put that code between the `\startbuffer` and `\stopbuffer` commands shown above, and embedded the `\processMPbuffer` command in a `\midaligned` command and it produced the following:



Pretty easy.

7 Current thoughts on ConTEXt

Having tried ConTEXt to the limited extent described above, I have a few opinions on it versus

LaTeX. (The following is in two columns only because I wanted to try the `\startcolumns[n=2]` and `\stopcolumns` commands.)

1. It will take a while to get used to not having as much default document formatting as exists in the LaTeX classes such as `book` and `article`. I assume many people have implemented their own versions of such classes for ConTeXt but so far I have not found an archive of them that new users can use as is or adapt to their own needs. (One can see lots of example source code via the **Sample documents** page of the ConTeXt Garden wiki. I find this to be is a particularly useful way to learn things about ConTeXt. For instance, I noticed in one source file that `\CONTEXT` appears to be built into ConTeXt as an alternative to `Con\TeX{}t`.)

2. It will take a while to get used to the large ConTeXt command set that is new to me.

3. There is a massive amount of **ConTeXt documentation** available. For instance, there appear to be 41 separate manuals related to ConTeXt at `Pragma-ADE.com` (with interactive screen-based versions available for at least a few of them). There is the extensive ConTeXt Garden wiki. And there are other ConTeXt websites.

   However, I find it a little hard to find my way around all this document. Patrick Gundlach (of the ConTeXt Garden) apparently somewhat agrees with me and has written a sort of **mission statement** explaining his documentation augmentation project. There also appears to be no single tutorial book such as Kopka and Daly's *A Guide to LaTeX* and no single reference work parallel to the *The LaTeX Companion* by Mittelbach et al.

   Making things still harder for me is that the documentation often seems a little cryptic. For instance, the ConTeXt Garden wiki has a substantial page on page layout including setting up headers and footers, and the ConTeXt manual has a substantial section on headers and footers in its layout chapter; however, neither or these made it clear to me how to have a different footer on the first page of a document. In fact, I couldn't even find a description of the command description syntax that the manual uses. Through trial and error I eventually found something that worked, but even having working code did not make the header and footer documentation clear to me.

   Clearly it will also take a while for me to get used to ConTeXt's documentation.

   One of the purported reasons for there being less than definitive documentation for ConTeXt is that the system changes fast—for instance, in response to user requests. Of course, rapid accommodation of user needs is desirable in many ways.

4. Some things seem easier or better in ConTeXt than in LaTeX; for instance, adding an image with the text flowing around it (as shown in the Biographical note below) was trivial in ConTeXt, and I could never be bothered with figuring out how to do this in LaTeX (no doubt there is a package that provides this capability). Another example: I noticed while creating a footnote that ConTeXt apparently can handle verbatim text in footnotes simply using `\type`. (As I remember, a `\verb` in a LaTeX footnote requires special handling different than when it is used in the main text.)

5. While I have not dug really deeply into ConTeXt yet, it looks like ConTeXt will automatically do a better job of some things that are important to me, such as having the bottom lines of facing pages match. There also appear to be a bunch of non-TeX utilities that are integrated with ConTeXt that are going to be very useful as I learn to use them.

6. Being able to just include lines of MetaPost in a ConTEXt file seems quite wonderful. (More generally, the `texexec` program apparently does a lot of stuff for the ConTEXt user that I don't fully grasp yet.)

7. ConTEXt is not as well integrated out of the box with my editor, WinEdt, as LaTeX is. It is a slight bother having to run ConTEXt in command line mode in parallel with editing with WinEdt rather than just clicking a button in WinEdt to compile with ConTEXt. (I should be able to configure WinEdt to have such a ConTEXt button, but can't be bothered to do so.)

With this much learning and use of ConTEXt under my belt, I feel like I am over some hump and well on my way with ConTEXt and will surely use it again for a bigger project in the near future.

All in all, I am impressed with ConTEXt. Also, its approach of being more like Plain TEX than LaTeX in terms of not assuming much logical document structure fits well with my personal inclinations. However, the next time I must definitely figure out how to use ConTEXt's various capabilities for structuring and reusing the components of a document. That—and how to use different fonts with ConTEXt—will be my top learning priorities.

[*This column continues on the next page*]

Appendix: Grid experiment

Just for fun, I included a `\showgrid` command, just before the title line for this appendix. I also included a `\page` command so the grid wouldn't obscure my concluding remarks to the previous section. This capability would be useful if I wanted to work harder at improving the vertical spacing of the photo and text in the biographical note.

Acknowledgements

I greatly appreciate the efforts of the authors of the various pieces of ConTEXt documentation I mentioned in the body of this column. I drew particularly heavily from Bill McClain in Section 3.

Karl Berry reviewed a draft of some parts of this column and caught a number of typos. Yuri Robbers reviewed the full draft and provided many useful suggestions for improvement. Issue editor Paul Blaga and editor-in-chief Lance Carnes spotted typos.

Biographical note

David Walden is retired after a career as an engineer, engineering manager, and general manager involved with research and development of computer and other high tech systems. The photograph at left is from when he was considerably younger and not as in focus as he is now. More history is at **www.walden-family.com**.

As mentioned near the end of Section 4. the photo was placed here using `\placefigure[left]{none}{\externalfigure[dave-head.pdf]}`.

Journal home page
General information

Submit an item
Download style files
Copyright
Contact us

THE PracTEX Journal TEX USERS GROUP

Ask Nelly: How do I create footnotes to tables without developing an ulcer in the process? What is a good way to create subfigures within one float? How do I typeset a critical edition?

The Editors

Abstract

Ask Nelly is a question and answer column. Nelly is the quiet person who sits at the back corner desk, who knows a lot, and when asked any question is always ready with a patient answer. If Nelly doesn't know the answer, Nelly will know an expert who has the answer. Feel free to Ask Nelly about any aspect of LaTeX, TeX, Context, etc.

- Comment on this paper
- Send submission idea to editor

TPJ

Q: Dear Nelly: When I use one of LaTeX's standard tabular environments, I find it exceedingly messy to add footnotes in that table. Since I want the footnotes directly under the table (of course), I have to deal with redefining footnote-related commands, setting up minipages, experimenting to figure out the correct widths for each of these minipages, and when everything finally looks ok, and I decide to add or delete a column, I need to redesign the minipage! Do you know a better way to achieve what I want?

A: I completely understand your frustration. Having to deal with these issues is tedious and ought to be unnecessary. Luckily TeX guru *Donald Arseneau* thought the same, and he create the `threeparttable.sty` style file. If you do not have it installed yet, you can find it in CTAN directory

```
macros/latex/contrib/misc/
```

Just include it with a

```
\usepackage{threeparttable}
```

And then you can easily create tables consisting of three parts (the caption, the table itself, and the table notes) like this:

```
\begin{table}
  \begin{threeparttable}
    \caption{Wild animals of {\TeX}land\tnote{1}}
    \begin{tabular}{ll}
      \hline
      Diminutive & Gnat\tnote{2}\\
                 & Foobug\\
      \hline
      Fair-sized\tnote{3} & Armadillo\\
                          & Wumpus\tnote{4}\\
                          & Borogove\\
      \hline
      Huge & Gnu\tnote{5}\\
           & Jabberwock\\
      \hline
      Unknown & Grue\tnote{6}\\
      \hline
    \end{tabular}
    \begin{tablenotes}[para]
       \item[1] {\TeX}land is usually travelled by the famous
              explorer \emph{Dave Walden};
       \item[2] First discovered and described by the zoologist
          Leslie Lamport in 1985;
       \item[3] Fair-sized means not smaller than a mouse (the
                animal, not the pointing device) and not larger
               than a donkey;
       \item[4] Beware of the Wumpus: it is both elusive and
          dangerous;
       \item[5] First discovered and described by the zoologist
           Richard Stallman in 1984;
       \item[6] Grues are exceedingly dangerous, they only occur
           in the dark, and no one who encountered them lived
            to tell the tale.
    \end{tablenotes}
  \end{threeparttable}
\end{table}
```

As you can see, we can use a normal LaTeX table environment for the float, as well as a normal `tabular`, `tabular*`, `tabularx` or any of the `array` environments. The table notes are all marked with `\tnote{}`, so there is no need for awkward redefinitions. The notes are typeset using one or more `tablenotes` environments, each of which can take an optional argument (`para` in this case, other options are `flushleft`, `online` or the default `normal`) that determines how the notes are being typeset.

The one disadvantage that I can think of is that there is no automatic numbering of notes (yet).

The above question was answered by **Yuri Robbers**, a member of the editorial board of this journal. He can be reached at yuri.robbers@gmail.com

TPJ

Q: Dear Nelly: In my thesis I have many floats that contain two, three or even four graphs. I've used `minipage` environments for each graph within one float, and I have labelled those visually so that I can refer to each graph in the caption. My problem is, however, that every time I change one of the figures, I need to re-align all these minipages, and things get messy. Do you know how I can life easier for myself?

A: Yes, I do. Your life will be a lot easier with the `subfig` package by *Steven Cochran*. This package may already be installed on your system, but if need be it can be downloaded from CTAN in the

```
macros/latex/contrib/subfig/
```

directory. Include it with a

```
\usepackage{subfig}
```

Then you can use the `\subfloat{}` command within your floats to include each graph. They will automatically be numbered using alphanumeric labels within parentheses. You can use an optional argument to extend these labels with a subcaption. An example:

```
\begin{figure}
  \centering
  \subfloat[Annual rainfall in {\TeX}land, 1977--2007]{
    \includegraphics[width=0.4\textwidth]{rainfall}
  \qquad
  \subfloat[Number of sunny days \emph{per annum} in {\TeX}land,
    1977--2007]{
    \includegraphics[width=0.4\textwidth]{sunshine}
  \caption{The pleasant climate of {\TeX}land illustrated.}
\end{figure}
```

There are many options to this package, which are explained in detail in the accompanying documentation.

The above question was answered by **Yuri Robbers**, a member of the editorial board of this journal. He can be reached at yuri.robbers@gmail.com

TPJ

Q: Dear Nelly: I have been wanting to get around to this for a while but haven't had the time. I wish to use (or make) a class that can be used for bilingual texts, that is I wish

to have every second page print the translated language. Obviously the main concern is having the translated page in sync with the original language, so there is an issue about the grouping of the text into parts. Do you know of any classes or can you think of a way to do this?

A: There are several packages able to do typeset critical editions, including parallel texts in two languages, complicated line numbering and multiple sets of footnotes as well as multiple sets of endnotes available. The three that are — to my knowledge — most often used are Edmac, Ednotes and Ledmac, all for LaTeX:

Edmac http://tug.ctan.org/cgi-bin/ctanPackageInformation.py?id=edmac
Ednotes http://tug.ctan.org/cgi-bin/ctanPackageInformation.py?id=ednotes
Ledmac http://tug.ctan.org/cgi-bin/ctanPackageInformation.py?id=ledmacs

They differ a bit in capabilities, so it's best to skim the documentation of all three before deciding which one to use.
Good luck!

The above question was answered by **Yuri Robbers**, a member of the editorial board of this journal. He can be reached at yuri.robbers@gmail.com

TPJ

THE PracTEX Journal

Distractions — From Shakespeare, with Love

The Editors

- Comment on this paper
- Send submission idea to editor

From Shakespeare with Love

In recent issues of the journal, this column usually contained crosswords or chess problems. In this issue, we decided to draw the attention of our readers on a package which is not widely known (although it belongs to Donald Arseneau, the author of some of the finest LaTeX packages), namely *shapepar*. The package is available from CTAN and the original documentation can be found at the address http://www.ctan.org/tex-archive/macros/latex/contrib/shapepar/shapepar.pdf

The package can be used to produce paragraphs having a "non-standard" shape. Even more, you can define yourself new shapes, other than the one predefined by the package. Once the shape is defined, it is very easy to use it. We chose, for this column, to write one of the most beautiful of Shakespear's sonnets, Sonnet 66, in the form of a heart. As remarked earlier, it is very easy to do it. The command is just

\shapepar{\heartshape}{ ...}

You can see the result below:

Sonnet 66.

as well the source.

Enjoy!

Shakespeare – Sonnet 66

Tired for all these, for
restful death I cry; As, to behold desert
a beggar born, And needy nothing trimmed
in jollity, And purest faith unhappily forsworn,
And gilded honour shamefully misplaced, And
maiden virtue rudely strumpeted, And right perfec-
tion wrongfully disgraced, And strength by limp-
ing sway disable, And art made tong-tied by au-
thority, And folly, doctor-like, controlling skill,
And simple truth miscalled simplicity,
And captive good attending captain ill.
Tired with all these, from
these would I be gone,
Save that to die I
leave my love
alone.