BLUe's Bibliography—a generic approach

Kees van der Laan

Hunzeweg 57,
9893 PB Garnwerd, The Netherlands, 05941–1525
cgl@risc1.rug.nl

Abstract

A new and flexible approach to maintaining and using a bibliography database within (All)TEX is provided.

Keywords: Literature database, bibliography, plain TEX, macro writing, education.

Introduction

For typesetting a bibliography or a list of references many TEX-oriented tools are available[1]

- LATEX's bibliography environment, with its database facility BIBTEX (also used in $\mathcal{AMS}$-LATEX)
- $\mathcal{AMS}$-TEX's `\ref` ... `\endref`
- (L)TUGboat's, casu quo (L)TUGproc's conventions
- Alexander's TiB package
- Durst's Old Shell Game.

From the above it is concluded that there is no generally agreed upon (All)TEX tool to handle effectively and context-independently a list of references.

The incorporation of bibliographic information into a publication is complex because of the how-to-s

- set up and maintain a database of references
- incorporate the needed entries in a publication
- cite to the references from within the paper.

The static approach of selecting with an editor, and format these selected entries for the occasion, means a repetition of the clerical work for each publication. As long as one publishes occasionally that is no problem. Just go ahead, as simple as that.

A regular publishing author must adapt the mark-up of the references each time he submits an article. This is labour-intensive and error-prone. The more so when the local bulletin takes a different format from TUGboat, when we concentrate on publishing within the TEX-community.

This work emerged from the wish to maintain a database independently from any TEX publication series, parameterized such that it is easy to select entries from the database and to format these by customizing parameter macros.

In this paper an independent way to maintain a bibliography database, to select entries from the database, and to the format these, *all within* TEX, is worked out. For each publication the total database is used where the selection of the required entries is driven by a list of names pointing to the entries. To format the list appropriately a few formatting macros have been provided, which can be tailored to the conventions required by the publication series.

> For each publication the list of names `lit.sel` has to be extracted from the (total) list `lit.lab`, and `\ls` customized.

The problem of symbolic referencing to the list of references is solved en-passant, without the need for a multi-pass job.

BLUe stands for Mr BlUe—my innocent user and relative of Ben Lee User of the TEXbook fame.

Generic means that the proposed method can be used with (All)TEX, provided one adheres to a little discipline, and adjusts `\ls` for that particular situation.

Why yet another tool? Basically because BLUe's Bib

- is simpler
- is generic and cooperates with (All)TEX
- needs no (external) sorting
- needs no multi-pass job
- does not create new auxiliary files
- does not need external tools, except for the editor to handle the `lit.dat`, `lit.lab` files, in altering, or extending, the data(base).

Disclaimer.
I don't claim to have solved once and for all the formatting of the references, and the extracting of the database entries within TEX. All those user groups which publish TEX-oriented bulletins, have their peculiar formatting wishes. It is near to impossible to account for all of

[1] Wondering about how Knuth handled his references makes you realize that some persons are a class apart, in doing basic research. I would not dream of referencing via the index.

those conventions. Hopefully, BLUe's Bib will find its niche.

1 The basic approach

One has to decide upon the database tool, the tagging conventions, and the syntax of the entries.

The tool. There are many database tools available, for instance dBASE, next to more advanced relational database packages. What I need is a flexible, and general available tool

which serves a lifetime.

TEX meets these requirements. So, I decided to use (All)TEX.[2]

Tagging what? The next problem is to decide about the amount of tagging. Of course one can tag all the items, and that is the best and most general approach.[3] However, I don't need that complete generality and certainly don't like its overhead, nor its hindering effects induced by the full generality. For me names and years are important issues related to any bibliographic entry. I know by heart the kind of publication once I know the author's name and the year of publication, in the area I'm working. When I can handle these fields, and also have some parameter macros to typeset the title and the name of the journal that would be enough. Moreover, I require that the entries will be supplied in a systematic and (near) natural way. Because of this modest approach I can always incorporate more tagging, casu quo formatting, when needed.[4] The hardest thing is to *maintain consistency.* I hope I will succeed by this simple and natural approach, although consistency is not imposed and I have no check for adhering to it.

Basically, I like to supply each entry in a (near) natural way, as suggested by the following example

```
Knuth, D.E (1984):
The \TeX book.
Addison-Wesley.
ISBN 0-201-13447-0 (hard cover)
ISBN 0-201-13448-9 (soft cover).
(For the right printing look for
 \cs{language}, or \cs{emergystretch}
 in the index.)
```

which obeys the syntax

```
<name part> (<date part>):
   <title part>
   <rest>
```

This syntax leaves freedom and flexibility, although the essentials—name, date, title[5]—have prescribed positions. For example the number of authors does not matter, nor is it prescribed how the authors should be specified. That is up to you. The representation of the date is free too, leaving room for things like 'in progress' or 'priv. comm.' Nor do I like to prescribe structurally the title in more detail. The rest part is not further prescribed either, because of the great variety of sources—publishers, books, journal series, proceedings, theses, technical reports, and the like.

I like to call the above the ASCII entry of the database.[6]

Database entries. In view of using the entries with TEX as selector and formatter, I decided to supply the entries as replacement text of a def, that is the entries have been supplied with a *name*.[7] For the name I chose to use lowercase letters only: the (first) name followed by the initials and extra letter(s), the latter in case of more than one publication per author. For example

```
\def\knuthded{<ASCII entry>}
```

I allow modest TEX mark-up in the ASCII entry. For example diacritical marks needed for the names, in the title part and so on. In the rest part I don't mind that TUGboat is indicated by TUGboat's convention `\tubissue` with its parameter specs. Furthermore, it is just handy to allow `\TeX`—TEX—and `\PS`—PostScript—and the like for TEX-related names, as supplied in `tugboat.cmn`. I'm not worried much about the amount of TEX mark-up because I expect TEX to be the formatter for the rest of my life!

1.1 Conventions.

My conventions for tagging each entry are as follows, mostly to enhance consistency.
Name part.
Start the name part with the family name of the (first) author followed by his initials with a period separating each next initial and followed eventually by interjections like `van` and the like. This followed by the next author names, if any, separated by a comma from each preceding author. Each name, except for the first, starts with its initials, separated by periods, and each last initial separated by a space from the family name.[8]

[2] Note that the file of entries can be maintained by an editor.

[3] Practised by BIBTEX, AMS, and Tib.

[4] In fact I doubted about to include a macro for tagging the titles. For the moment I just required that the title follows the colon and ends by its end-of-line. I guess that will be sufficient.

[5] I consider it important to delineate the main fields, however. That is the separation of the various parts. For example the title part is enclosed between a colon and its end-of-line.

[6] There is usually too much formatting in a bibliography. Why discriminate between a title of a book and an article? Given the context that is clear enough. Especially when for a book the ISBN (or ISSN) number is provided. And why superfluous punctuation? For me there should be as little interpunction as is functionally needed to separate the elements of the entry.

[7] In database jargon these names are the primary keys.

[8] Quite a mouthful already!

Date part.
I use the year, and when it concerns proceedings `editor, <year>`. I also adhere to `in progress` and `priv. comm.`
Title part.
Just provide the title, as straight as possible.
Rest part.
My usual hierarchy is to start with the source followed by secondary sources, the publisher data, ISBN etc., and annotations to end up with. That is: Name of journal, issue number, page numbers and a terminating period. Then some `(Also ...)` parts within parentheses. Publisher again terminated by a period.[9] ISBN or similar number again terminated by a period. And at the end annotations or comments within parentheses.

Selecting entries. For selecting I made use of the active list separator `\ls`, with context dependent meaning. When the database file consists of the entries

```
\def\knuthded
   {Knuth, D.E (1984):
   The \TeX book. \aw.
   ISBN 0-201-13447-0 (hard cover),
        0-201-13448-9 (soft cover).
   (For the right printing look for
    \cs{language}, or \cs{emergystretch}
    in the index.)}
\def\knuthdeg
   {Knuth, D.E (1986):
   The \mf book. \aw.
   ISBN 0-201-13445-4 (hard cover).}
\def\salomond
   {Salomon, D (1992):
    NTG's advanced \TeX\ course:
    Insights and Hindsights.
    MAPS Special, $\approx$500p.}
```

then for selecting them all

```
\ls\knuthded \ls\knuthdeg \ls\salomond
```

I like to call this list the file `lit.sel`.[10] Note that the list is alphabetically ordered.[11] This way of selecting can also handle the formatting, by an appropriate definition of `\ls`. For example

```
\def\ls#1{\advance\bcnt1
          \item{[\the\bcnt]} #1}
```

which with `\newcount\bcnt` will yield the entries as an 'item list,' similar to Knuth's example in TEXbook p.341.

In summary. While marking up a publication the following scheme must be obeyed to handle the formatting of a list of references

```
%Front matter
%Copy proper
%Back matter
\head*References* %Or similar
\input tugboat.cmn%Abbreviations
\input lit.dat    %the data(base)
\input lit.tex    %\def\ls{...} etc.
\input lit.sel    %list of selected names
\bye
```

2 Cross-referencing

The last problem to deal with is the handling of symbolic cross-referencing.

We know already the 'names' and the order of occurrences of the references via `lit.sel`. Because of this we can at the beginning of the publication associate numbers—or whatever one wishes to typeset as citations—to the entries in the list of publications.

The idea is to redefine temporarily the 'names' with as replacement texts what has to be typeset.[12] Let us keep it simple and work out the example of citation by numbers.

With the above structure of the `lit.sel` file one can simply provide the definition

```
\def\ls#1{\advance\bcnt1\edef#1{\the\bcnt}}
```

Let us call the file with the above definitions `lit.ass`.[13] By this approach the control sequence `\knuthded` will yield in the copy proper 1,[14] and so on.

In summary. Handle references, with the possibility to cite them by their names, and format them according to the definition of `\ls`, as follows in publication

```
%Front matter (title, abstract etc.)
\begingroup
\input lit.ass%association macros
\input lit.sel%names are linked
             %to citations
%begin copy proper
\head*Introduction*%Or similar
...%\knuthded etc. yields its citation
   %number or whatever you chose
\head*Conclusions* %Or similar
...
%end copy proper
\endgroup\noindent
%Back matter
\head*References*  %Or similar
\input tugboat.cmn %a.o. the abbreviations
\input lit.dat     %the data
\input lit.tex     %the formatting macros
\input lit.sel     %the list of names
\bye
```

[9] City is superfluous in the context of an ISBN number.
[10] Mnemonics: literature selection.
[11] It can be ordered if needed by my Sorting in BLUe macros within TEX.
[12] Earlier I considered typesetting the bibliography at the beginning of a document with appropriate page numbers, and associate citation texts. When printed the pages can be palced wher we want, or `dvitodvi` can be used.
[13] Mnemonics: literature association.
[14] Brackets can be added if wanted. I refrained from introducing them as part of `\ls` to allow for a sequence of citations.

Note the data integrity: the file `lit.dat` is only *used* at the end, and the file `lit.sel` is used at the beginning and at the end. No rewriting! The names have been redefined within a scope, however.

Examples

As examples I have formatted

- the list of references as printed at the end of this paper (LaTeX context), and
- my total database of TeX-related references (TeX context).

For cross-referencing input `lit.ass` and `lit.sel` at the beginning, and don't forget to provide for the grouping.

Within a LaTeX context. The list of references at the end of this article has been obtained, within the context of `ltugproc.sty`, via

```
\documentstyle{ltugproc}
%Front matter, with lis.ass, lis.selbb
%Copy proper of this article
%Back matter
\section*{References}
\input{lit.dat} %data
%\input{lit.tex}%just the next 2 lines!
 \frenchspacing
 \def\ls#1{\ea\bibentry#1\endgraf}
%\input{lit.selbb}%selected for BLUe's
                % Bib the following list
 \ls\alexanderjc
 \ls\amsd
 \ls\amsf
 \ls\amsj
 \ls\arsenaud
 \ls\beebenhfc
 \ls\durstlka
 \ls\knuthded
 \ls\laancgk
 \ls\laancgm
 \ls\laancgo
 \ls\lamportlb
 \ls\rahtzspq
 \ls\whitneyrf
\end{document}
```

Within a (plain) TeX context. The printing of my complete (All)TeX-related database, can be done within plain as follows

```
\beginsection Bibliography\par%TB 340
\input tugboat.cmn%abbreviations
\input lit.dat    %complete database
\input lit.tex    %formatting macros
\input lit.lab    %list of all names
\bye
```

with, when context free, `\ls` as supplied earlier

```
\def\ls{\advance\bcnt1
           \item{[\the\bcnt]} #1}
```

and, within the AMS spirit[15]

```
%The file: lit.tex
%macros to format lit.dat,
%driven by lit.lab
\frenchspacing
\newcount\bcnt \newcount\suffixcnt
\let\lstnme\relax \let\lstyear\relax
%
\def\ls#1{\ea\bibitem#1}
%
\def\bibitem#1(#2){\global\advance\bcnt1
 \def\authornme{#1}\def\authoryear{#2}%
 \ifx\lstnme\authornme
  \def\authornme{------}%
  \ifx\lstyear\authoryear
   \global\advance\suffixcnt1
   \def\authoryear{--}%
  \else\let\lstyear\authoryear\suffixcnt0
  \fi
 \else\let\lstnme\authornme
      \let\lstyear\authoryear\suffixcnt0
 \fi
 \item{[\the\bcnt]}\authornme\,(\authoryear
  \suffix)}%end \bibitem
%
\def\suffix{\ifcase\suffixcnt\or a\or b\or
 c\or d\or e\or f\or g\or h\or i\or j\or
 k\or l\or m\or n\or o\or p\or q\or r\or
 s\or t\or u\or v\or w\or x\or y\or z\fi}
\endinput
```

and with data

```
%The file: lit.dat
\def\alexanderjc
  {Alexander, J.C (1986):
   Tib, a reference setting package.
   \tubissue{7}(3), 138\dash139.
   (An update note is in
                    \tubissue{8}(2), 102)}
%...and so on
```

3 Maintenance?

The *maintenance of the data* comes down to extending and maintaining the file `lit.dat`. This has nothing to do with the formatting proper of a publication at hand. Because of the systematics used, especially that `\def\<name>` is on one line, it is easy with a programmable editor, an AWK script or a similar tool, to extract these lines and replace `\def` by `\ls`. In doing so the file `lit.lab` has been obtained without retyping, and therefore without retyping errors.

For each publication an author has to create the list `lit.sel` as a subset of the list `lit.lab`, again by an editor or a similar tool. A discipline is needed when references have to be inserted, while proofing. Update `.dat`, `.lab`, `.sel`, by hand I guess, or start first by preparing the contents and do finally the bibliography. It's all a matter of discipline.

[15] It is always cumbersome to obey the wishes of others.

4 Looking back

When developing a tool the most difficult thing is to know when to stop. A work is never perfect nor finished, there is always the polishing phase. My experience is that when I think I'm finished it turns out to be true for only 80% or so! The energy needed for the last 20% is as much as for the first 80%, if not more! In the spirit of Rahtz' UKTUG presentation of 1990 on the issue, I pondered about the advantages, disadvantages and what else?[16]

Advantages.

- simple (approach, TEX encodings, and use)
- generic (can be used with any TEX flavour)
- one tool—TEX—for selecting (via TEX's hashing of accessing def-s) and for formatting
- near natural way of specifying the entries, limited tagging
- one-pass job
- open-ended, extensible.

Disadvantages.

- (unique) names have to be looked up, no pattern-matching search
- limited tagging of the elements of the entries
- no support for adhering to consistency, otherwise than using control sequences.[17]

What else? With respect to data-integrity and alleviating the clerical work—or avoiding the intelligent AWK-like scripts—it would be worthwhile to generate `lit.lab` automatically from the natural specified data, via pattern matching techniques. Ipso facto for `lit.sel` driven by specification of free keywords. For the moment I stopped, however, and will look over BLUe's shoulder how things go in practice.

Acknowledgements

The idea of suppressing repeated names in the example of printing the complete database, is borrowed from $\mathcal{AMS}$-TEX.

Erik Frambach is kindly acknowledged for proofing the article, and for stressing to add the reason why, as well as for suggesting that it would be helpful to include summaries of earlier works on the issue.

Conclusions

A simple, flexible and generic approach for handling a database of bibliographic entries is proposed, within the context of TEX, independent from a special TEX flavour.

The approach allows for selecting, formatting and cross-referencing.

I use this method for a database with hundreds of entries, with a few percent to be selected each time.

As examples it is shown how to process the included list of references within a LATEX context, and how to process the complete database within a (plain) TEX context as such and adhering to AMS' style.

Sorting on the fly, see [5, 11]—whoops this citation has been done via `[\arsenausd, \laancgo]`—is no longer needed, nor does one need a multi-pass job.

As suggested in my AMS BLUes paper, the above can be worthwhile for a publisher, relieving authors from the details of formatting a bibliography by letting them just supply the names to the database of (pre)formatted entries, available already at the publisher's computing environment.

References

Alexander, J.C (1986): Tib, a reference setting package. *TUGboat* 7, no. (3), 138–139. (An update note is in *TUGboat* 8, no. (2), 102. A C program inspired upon troff's refer (and its successor bib). It is a preprocessor tailored for use with (All)TEX. It allows for a few citation styles, and a publisher can extend these. It provides also for a powerful word-definition citation search. Flexible in general. Tagging of the entries a la refer has to be obeyed. From the UKTuG meeting of 1990 the following characteristics. It consists of: tibdex, a hashing program to speed up access to large collections of citations, by creating an index to the bibliography; tiblist, to format and print the database; looktib, a program to query the database; tib, main program goes through the source document and looks for keywords between [and]. Advantages: 1-pass, pattern-matching citation no need to invent unique key, compatible with UNIX refer databases, good toolbox of style elements, single index of bibliographies, formatting more-closely related to TEX. Disadvantages: pre-processor system, possible wrong matches, possible errors in repeated citations, non-algorithmic style language, left up to Tib to work out citation type, lack of extensibility in field, not very robust (crashes).)

AMS (1993): $\mathcal{AMS}$-TEX User's Guide 2.1.

AMS (1993): $\mathcal{AMS}$-LATEX User's Guide 1.1.

AMS (1993): Guidelines for preparing electronic manuscripts. $\mathcal{AMS}$-TEX (booklet, 52p), and the mirrored one $\mathcal{AMS}$-LATEX (booklet, 58p). (The first is very well-done. I have not seen a guideline of similar quality of yet! Simply the best available. Much experience is embodied to learn from. The second is verbose, incomplete, and deals at length with issues an author should not be bothered with.)

[16] Keep in mind however, that a disadvantage can also be interpreted as an advantage and vice versa! For example simpleness can be seen as an advantage, but is experienced as a disadvantage when bells-and-whistles are needed.

[17] An author can be relieved alltogether of the concistency aspects by allowing him to specify the lables, *not the contents!*

Arsenau, D (1992): overcite.sty, drftcite.sty, citesty. (From the file server. No longer needed when BLUe's Bib is used.)

Beebe, N.H.F (1991): The TUGlib server. MAPS 91.2, 117–123. (Also TEXline 11.)

Durst, L.K (1989): Bibliographic citations, or variations on the old shell game. *TUGboat* 10, no. (3), 390–394. (It discusses how to cope with TEXing a bibliography. BLUe's Bib has undoubtedly been influenced by this work, because I read it at least a year before creating BLUe's Bib. Durst does not use the list separator TEXnique, and therefore the encoding is a little more complex. Also the writing to a file and the external sorting makes it more cumbersome. In the paper it is not worked out how to use it within one of TEX's flavours. The basic approach is very similar to mine, however.)

Knuth, D.E (1984): Computers and Typesetting. The TEXbook. Addison-Wesley. ISBN 0-201-13447-0 (hard cover) ISBN 0-201-13448-9 (soft cover). (For the correct printing look in the index for `\language` or `\emergystretch`.)

Laan, C.G van der (1993): Manmac BLUes—or how to typeset a book via TEX. MAPS 93.1, 171-191.

Laan, C.G van der (1993): Sorting in BLUe. MAPS 93.1, 149-170. (Abridged TUG '93. *TUGboat* 14, no. (3), ?—?)

Laan, C.G van der (1993): Typesetting number sequences. MAPS 93.1, 145–148. (Submitted TUGboat. No longer relevant when BLUe's Bib is used.)

Lamport, L (1985): LATEX User's Guide & Reference Manual. Addison-Wesley. ISBN-0-201-15790-X. (With respect to BIBTEX the following characteristics from UKTUG meeting of 1990. Advantages: clear layout of database, unique identifyer for elements, compatibility with Scribe databases, explicit statement of citation type, extensible style language, uses LATEX's cross-referencing, easy to edit output, can be mixed with non-automatic generated bibliographies, cross-referencing and abbreviations. Disadvantages: multiple passes (LATEX, BIBTEX, LATEX, LATEX); have to remember unique references, style language is opaque, database is very wordy and boring to enter.)

Rahtz, S.P.Q (1987): Bibliographic tools. Literary and linguistic computing, 2, 4, 231–241.

Whitney, R.F, B.N Beeton (1989): TUGboat authors' guide. *TUGboat* 10, no. (3), 378–385. (Updated versions via the file server.)