
A personal book catalogue: bookdb

Peter Wilson

1 Introduction

For many years I have been collecting books. They are located in five rooms, as well as two bookcases in my printing and binding workshop. But recently I found that I was buying books that I already had. I decided that the best way to save money was to catalogue all the books that I owned and where they were kept.

I searched on the web for free database programs that would be appropriate. There were only a few that would run under Linux, which is the operating system I am most comfortable with. Of those, some I could not install, and the one that I could I couldn't get to work for me.

I contacted several booksellers that I dealt with and asked them what they used for cataloguing their stock. They all replied, but most used proprietary and expensive software that included things like preparing invoices that were irrelevant as far as I was concerned. The two responses that grabbed my attention were 'use a card index' (but I needed space for books, not card indexes) and 'try bibtex', which immediately appealed as I had used it for many years; why I hadn't thought of it myself I'll never know.

2 Usage

The standard BIBTEX entries did not meet my needs. I looked at `biblatex` [3] but its entries also didn't match my requirements so I decided to tweak BIBTEX. To that end I used Patrick Daly's `makebst` program [2] for generating a `*.bst` file that went some way towards meeting my needs. This required some hand-coded additions later; I read the articles by Oren Patashnik [5], the creator of BIBTEX, and Nicolas Markey [4], which helped me on my way. In the end I had a BIBTEX file called `bookdb.bst` [6] that included all the regular entries but a greatly expanded `book` entry, as follows:

book	A book with a publisher <i>Required:</i> author or editor, title, publisher, year <i>Optional:</i> volume or number, series, address, edition, month, note, collator, foreword, preface, introduction, volumes, pages, illustrations, binding, size, condition, copy, location, category, value
-------------	---

Also, I created a new entry called `heading`, as follows:

heading	A heading in the bibliography <i>Required:</i> key <i>Optional:</i> note
----------------	--

The additional fields in the `book` entry are:

binding:	Information about the book's binding. Output as: Binding: 'binding.'
category:	The general theme of the book. Output as: Category: ' category .'
collator:	The name(s) of those who collated the book contents. Output as: 'collator' (collator(s))
condition:	The book's condition. Output as: Condition: 'condition.'
copy:	For a limited edition, the particular copy. Output as: Copy: 'copy.'
foreword:	The names(s) of the author(s) of the book's Foreword, if not written by the author(s) of the main text. Output as: foreword by 'foreword'
illustrations:	Information about the number and kind of any illustrations and possibly who created them. Output as: Illustrations: 'illustrations.'
introduction:	The name(s) of the author(s) of the book's Introduction, if not written by the author(s) of the main text. Output as: introduction by 'introduction'
location:	Where the book is located. Output as: Location: 'location.'
pages:	The total number of pages. Output as: 'pages' pp.,
preface:	The names(s) of the author(s) of the book's Preface, if not written by the author(s) of the main text. Output as: preface by 'preface'
size:	The book's physical dimensions. Output as: Size: 'size.'

value: The book's value. Output as:
Value: 'value.'
volumes: The number of volumes. Output as:
Volumes: 'volumes.'

I use the `heading` entry for putting a heading or division marker into a bibliography. The `key` is required so that the `heading` is sorted into the correct position in the bibliography (normally sorting is based on the author or editor). The contents of the `note` field form the printed heading. For instance if you wanted a heading before each alphabetical group of authors you could do something like:

```
@heading{A1,
  key = {A1},
  note = {\thead{AAAAAAA...}}
}
@heading{B1,
  key = {B1},
  note = {\thead{BBBBBBB...}}
}
etc
```

where `\thead` might be defined as:

```
\providecommand{\thead}[1]{%
  \textbf{\large #1}}
```

To help clarify matters, Figure 1 shows a possible entry in a `*.bib` file. The output after processing in a document using `bookdb.bst` is illustrated in Figure 2.

As an example this is a file that I use for printing a catalogue of my books, where the book details are in file `mybooks.bib`. Note that using `bookdb` requires the use of the `natbib` package [1].

```
% books.tex a catalogue of my books
\documentclass[11pt,a4paper]{memoir}
\usepackage[T1]{fontenc}
\usepackage{natbib}
\pagestyle{empty}
\begin{document}
\nocite{*}
\bibliographystyle{bookdb}
\bibliography{mybooks}
\end{document}
```

3 Implementation

As I said earlier I had to extend the `bookdb.bst` file produced by the `makebst` program. I didn't really know how it all worked but after much trial and many errors I got something that on the whole met my needs. My basic process was to copy elements of the original `bst`, change some names, and see what was produced.

First of all I added the new `book` fields to the `bookdb.bst` `ENTRY` command as:

```
@book{ABOOK,
  author = {A. N. Author and A. Nother},
  title = {A Book Entry},
  editor = {Smith and Jones},
  collator = {Jane and Tim},
  translator = {Jo and Mary},
  foreword = {Alpha},
  preface = {Zoe},
  introduction = {Bloggs and Friend},
  volume = 7,
  publisher = {Herries Press},
  year = 2020,
  pages = {xii + 278 + vi},
  edition = {Third},
  isbn = {0-201-36299-8},
  volumes = 9,
  illustrations = {11 wood engravings},
  binding = {full red leather},
  size = {11 by 17 inches},
  note = {This is a note},
  condition = {Hot off the press},
  copy = {23 of 125},
  location = {my study},
  category = {private press},
  value = {\$270}
}
```

Figure 1: An example entry for `bookdb` processing

A. N. Author and A. Nother. *A Book Entry*, Smith and Jones (eds.), Jane and Tim (collators), Jo and Mary (translators), foreword by Alpha, preface by Zoe, introduction by Bloggs and Friend, volume 7 (Herries Press, 2020), xii + 278 + vi pp., third edition. ISBN 0-201-36299-8. Volumes: 9. Illustrations: 11 wood engravings. Binding: full red leather. Size: 11 by 17 inches. This is a note. Condition: Hot off the press. Copy: 23 of 125. Location: my study. Category: **private press**. Value: **\$270**.

Figure 2: The example's output

```
ENTRY
{ ...
  binding
  category
  collator
  condition
  copy
  foreword
  illustrations
  introduction
  location
  pages
```

```

    preface
    size
    translator
    value
    volumes
}

```

The next thing was to add the new entries in the correct order to the function that output the book bibliography entries, together with how they should be formatted. This was the final result after many repetitions of try it, BIB_TE_X it, change it. The original code is in a typewriter font and my additions are in slanted typewriter.

```

FUNCTION {book}
{ output.bibitem
  author empty$
    { format.editors "author and editor"
      output.check
      editor format.key output
      add.blank
    }
  { format.authors output.nonnull
    crossref missing$
      { "author and editor" editor
        either.or.check }
      'skip$
      if$
    }
  if$
  new.block
  format.btitle "title" output.check
  format.editors output
  format.collator output
  format.translator output
  format.foreword output
  format.preface output
  format.introduction output
  crossref missing$
  { format.bvolume output
    new.block
    format.number.series output
    new.sentence
    format.publisher.address output
  }
  {
    new.block
    format.book.crossref output.nonnull
  }
  if$
  format.book.pages output
  format.edition output
  format.isbn output
  format.volumes output

```

```

    format.illustrations output
    format.binding output
    format.size output
    new.block
    format.note output
    format.condition output
    format.copy output
    format.location output
    format.category output
    format.value output
  fin.entry
}

```

Effectively ‘all’ I had left to do was specify the formatting of my new fields. I used three basic forms:

1. Some introductory text, like ‘introduction by’ or ‘Illustrations:’, followed by the field data.
2. Like the first form but with the field data in a bold font.
3. Name(s) forming the field data followed by what their contribution was in parentheses.

As an example of the first form here is the code for binding:

```

FUNCTION {format.binding}
{ binding "binding" bibinfo.check
  duplicate$ empty$ 'skip$
  {
    new.block
    "Binding: " swap$ *
  }
  if$
}

```

Life was a little more complicated for the second form. This is the code for the value field which requires two functions, the first dealing with the bolding and the second with the output.

```

FUNCTION {boldval}
{ duplicate$ empty$
  { pop$ "" }
  { "Value: \textbf{" swap$ * "}" * }
  if$
}
FUNCTION {format.value}
{ value "value" bibinfo.check
  duplicate$ empty$ 'skip$
  {
    new.block
    boldval
  }
  if$
}

```

The third form required several functions, as in the code for collator, where if there is a single

collator this is output as ‘Name (collator)’ but if there are multiple collators the output is ‘Name1 and Name2 ... (collators)’.

```
FUNCTION {bbl.collator}
{ "collator" }
FUNCTION {bbl.collators}
{ "collators" }
FUNCTION {get.bbl.collator}
{ collator num.names$ #1 >
  'bbl.collators 'bbl.collator
  if$
}
FUNCTION {format.collator}
{ collator "collator" format.names
  duplicate$ empty$ 'skip$
  { " " *
    get.bbl.collator
    "(" swap$ * ")" *
  }
  if$
}
```

4 My book database

The *.bib for my book catalogue looks somewhat like this:

```
%%% mybooks.bib 2015/04/22
%%% for formatting headings
@preamble{ "\providecommand{\ahead}[1]{%
  \textbf{\large #1}}" }

%%% publishers
@string{CUP =
  "Cambridge University Press"}
% etc
%%% categories
@string{science =
  "science, mathematics, computers"}
% etc

@heading{A1,
  key = {A},
  note = {\ahead{AAAAAAAAAAAAA...}}
}

@book{A1KEY,
author = {First A. Author},
% etc
}

@book{A2KEY,
author = {Second A. Author},
% etc
}
% etc
```

Peter Wilson

I used the BIBTEX `@preamble` command to provide a definition of the `\ahead` macro. This, if required, can be overridden by an existing definition in the document used to print the bibliography.

I added various `@string` commands to provide shorthands for many of the fields in the *.bib file, such as publisher, location, category, that would have the same value. This meant that I could have a shortened field entry that looked like:

```
publisher = CUP,
```

instead of:

```
publisher = {Cambridge University Press},
```

5 Afterthoughts and unresolved problems

BIBTEX uses a stack-based language which I find hard to understand. Many years ago I wrote an interpreter for a stack-based language whose name I have forgotten but even so I was unable to use the language itself. I think that it is a little like crosswords. I like doing ‘cryptic’ crosswords but I find that with some setters I can follow their clues easily but with others I haven’t a clue.

My basic approach was to take an existing *.bst file, try and see what it did, then copy and modify what seemed relevant to my needs.

I did have a couple of infelicities that I did not manage to resolve.

The first was that no matter what I tried I could not stop the **heading** from outputting its **key**, so it should be made as short and unobtrusive as possible.

The second was that I couldn’t stop the warning message issued by BIBTEX if both an **author** and **editor** were supplied although the output was printed including both.

In spite of these, if you are a collector then you may want to consider tweaking a *.bst file to meet your particular needs.

6 Other collections

Perhaps, like me, you have collections other than books. These can also be catalogued via BIBTEX. For instance I collect Japanese woodblock prints and Western engravings, while you might collect pictures in general. I have no need to create a catalogue of my prints as I keep them in folders in one place, together with information about each one. If I did want to create such a catalogue I would start with `bookdb.bst`, renaming it to perhaps `pictures.bst`. Then add in the new fields for the prints and engravings, such as:

artist: The name (and perhaps the date) of the artist. Output like `author`.

engraver: The name of the engraver.

sensor: The name(s) of the censors. Output like collators.

Then define a new `japanese` entry, based on the `book`, with maybe the fields:

```
japanese  A Japanese woodblock print with
           an artist
           Required: artist, title
           Optional: engraver, censor,
                       publisher, date, series,
                       note, size, value, category
```

And a new `engraving` entry, based on the `book`, with possibly the fields:

```
engraving An engraving with an engraver
           Required: engraver, title
           Optional: date, series, note,
                       size, value, category
```

Then delete everything that might be irrelevant, such as `inproceedings`, etc.

You could use similar enhancements to catalogue, say, a collection of watercolours or Dinky toys or model trains.

It's up to you!

References

- [1] Patrick W. Daly. `natbib`—natural sciences citations and references, September 2010. <http://ctan.org/pkg/natbib>.
- [2] Patrick W. Daly. `custom-bib`—customizing bibliographic style files, November 2011. <http://ctan.org/pkg/custom-bib>.
- [3] Philipp Lehman and Philip Kime. `biblatex`—bibliographies in \LaTeX using `BIBTEX` for sorting only, March 2016. <http://ctan.org/pkg/biblatex>.
- [4] Nicolas Markey. Tame the BeaST, October 2009. <http://ctan.org/pkg/tamethebeast>.
- [5] Oren Patashnik. Designing `BIBTEX` styles, February 1988. <http://mirror.ctan.org/biblio/bibtex/base/btxhak.pdf>.
- [6] Peter Wilson. `bookdb`—a personal book catalogue, June 2015. <http://ctan.org/pkg/bookdb>.

◇ Peter Wilson
 12 Sovereign Close
 Kenilworth, CV8 1SQ
 UK
 herries dot press (at)
 earthlink dot net