
Addenda: A suggested “operational requirement” for L^AT_EX3’s treatment of bibliographic references

David Rhead

[Editor’s note: The original article with this title appeared in *TUGboat* 14, no. 4, December 1994, pp. 425–433. Owing to a production error, the version that appeared was the unrefereed original. The most significant changes made in response to the referees’ comments appear below. The full text of the revised version can be found on a CTAN node as `tex-archive/digests/tugboat/articles/14-4/rhead.ltx`; the associated references are in `rhead.bib`. Owing to the timeliness of the material, these files have been assigned a deletion date corresponding to the release of L^AT_EX3.

The Editor regrets the error.]

2 Doing it yourself

[New subsection; insert at the end of section 2.]

Multi-author documents

I think it desirable that L^AT_EX3’s successors to the “standard styles” should support multi-author documents (e.g., a journal-issue made up of a number of articles, or a conference-proceedings made up of several contributions).

Hence:

- it should be possible to have several reference-lists within a single document
- there should be allowance for the possibility that a single document may use two or more citation schemes. E.g., since the “instructions for authors” in *Mathematische Zeitschrift* gives a choice of three citation schemes, an issue of the journal may involve three distinct schemes.

3 Using bibliography-formatting software

[New subsection; adjust numbering.]

3.1.5 Hybrid approaches 2

Other hybrid approaches might use a proprietary system and B^IB^TE^X “in series”:

- One might regard the proprietary system’s database as a “staging post”, where information stays briefly before being converted to a B^IB^TE^X database. For example, if a proprietary system can import from library catalogues and export to a B^IB^TE^X .bib file, the approach gives a mechanism for getting information from library catalogues to B^IB^TE^X.
- Alternatively, one might regard the B^IB^TE^X database as the “staging post”. If a proprietary

system exports a B^IB^TE^X .bib file, information held in the proprietary database can be converted to a B^IB^TE^X database just before being used in conjunction with L^AT_EX.

The following problems are likely to arise with such approaches:

- The standard “L^AT_EX, B^IB^TE^X, L^AT_EX, L^AT_EX” sequence is already fairly laborious. An additional (“proprietary database to B^IB^TE^X database”) stage will make things worse.
- Questions could arise about “which database is the definitive, up-to-date one — the proprietary system’s or B^IB^TE^X’s?”.
- Mapping problems could arise. The usual B^IB^TE^X analysis of structure (in terms of entry-types and fields) differs from those used by other systems. In literature-areas where the B^IB^TE^X analysis is relatively coarse, subtleties will be lost if a finer analysis is mapped to the B^IB^TE^X analysis (e.g., if Library Master’s public document, manuscript collection, computer program, audio recording, video recording, interview, and artwork record-types are all converted to `@MISC`).
- Documentation may be cumbersome, since the end-user will have to consult that for the proprietary system, that for the conversion procedure, and that for B^IB^TE^X. The user will also need to understand the two lots of terminology, and be able to “translate” from one to the other.

Because of these potential problems, I’m not inclined to pursue this type of hybrid approach either.

3.1.6 The user’s choice

Given some *modus vivendi*, end-users would be able to make their own assessments of which bibliographic software suits their needs.

- Someone who wants ready-made methods of downloading information from commercial bibliographic databases, CD-ROMs, library catalogues, etc., will probably favour one of the proprietary programs. The proprietary systems also offer database administration and searching facilities.
- Different systems implement different analyses of the structure of “the literature” (i.e., using B^IB^TE^X’s terminology, there are different ways of defining entry-types and fields), and different people also have different viewpoints. E.g., an analysis that suits a scientist may be too coarse for keeping track of “primary sources” in the humanities.
- Cost is obviously a factor.

[Add the following at the end of the list.]

- Some software supports “imprecise citations” (e.g., “the item in my database whose author is ... which has ... in the title”). Others, such as `BIBTEX`, require a precise citation via a unique key. People who are continually adding items to their bibliographic databases may prefer the latter, so as to avoid situations in which a match becomes ambiguous even though a document’s text has not changed.

4 Miscellaneous

[Revised subsection.]

4.1 “Local names” for keys

If you are “doing it yourself”, choice of keys (i.e., in `LATEX 2.09` terms, the arguments for `\bibitem`) is unlikely to be a problem. For example, you could equally well use `lamport-86` or `latexbook` as a key for the `LATEX` manual. There is no particular need for consistency from one document to another: you can use `lamport-86` as the key in one document, and use `latexbook` as the key in another.

However, if you have a large bibliographic database (perhaps shared with a group of colleagues), it may be impracticable to keep track of keys assigned on an *ad hoc* basis, and difficult to guarantee that keys will stay unique whenever a new item is added to the database.

Moreover, a `.tex` file to be `\input` may contain bibliographic details and `LATEX` commands that are generated automatically by bibliographic software (even though `LATEX` will have no way of distinguishing the file from one that you might produce when “doing it yourself”). Such bibliographic software might be programmed to assign keys automatically, e.g.,

- based on the ISBN, in the case of books
- of the form `journal-volume-number-page`, in the case of journal-articles
- based on “record number”, if the bibliographic software assigns a unique number to each record in the database
- of the form `lamport-86`, constructed automatically from the “author” and “year” fields in the database.

There may be a dilemma about whether to have automatically assigned keys that are relatively easy-to-remember, or to have keys that are guaranteed to stay unique no matter what else gets added to the database. As an example, consider what key might be used for the `LATEX 2.09` manual: `lamport-86` is easy to remember, but is potentially ambiguous (because Lamport published other work in 1986); if the

ISBN 0-201-15790-X was used as a key, it should stay unique but would be difficult to remember.

To help cater for such situations, it might be useful if `LATEX3` allowed “local names” for keys, i.e., some mechanism whereby an author could declare (e.g., in a document’s root file) that, for the duration of a document, a particular “informal key” (to be used in in-text citation commands) should be treated as a synonym for a “formal key” (which appears in an entry in an automatically generated reference-list). For example, it might be useful to be able to declare that `lamport-86` can be used as a “local name” for 0-201-15790-X.

5 Acknowledgement

I am grateful to the referees for their comments on a draft of this paper. The paper incorporates a number of ideas taken from the referees’ comments.

◊ David Rhead
 Cripps Computing Centre
 University of Nottingham
 Nottingham NG7 2RD
 United Kingdom
 d.rhead@vme.nott.ac.uk