

# L<sup>A</sup>T<sub>E</sub>X

## The “operational requirement” (?) for support of bibliographic references by L<sup>A</sup>T<sub>E</sub>X3

David Rhead

### Abstract

It is suggested that:

- L<sup>A</sup>T<sub>E</sub>X3 should aim to support the principal citation schemes used in conventional publishing
- consideration be given to a *modus vivendi* between L<sup>A</sup>T<sub>E</sub>X3 and mainstream bibliography-formatting software.

### Contents

1	Introduction	425
2	Doing it yourself	425
3	Using bibliography-formatting software	427
3.1	Background	427
3.1.1	Software available	427
3.1.2	<i>Modus vivendi</i> with the main 4?	428
3.1.3	Preferred interface	428
3.1.4	Hybrid approaches	430
3.1.5	The user’s choice	431
3.2	OR for L <sup>A</sup> T <sub>E</sub> X3	431
4	Miscellaneous	431
4.1	“Local names” for keys	431
4.2	Reference-lists that are also indexes	432
A	Some suppliers of mainstream bibliographic software	432
B	E-mail discussion lists about bibliographic software	432

## 1 Introduction

Ideally, when writing software, it is a good idea to write down what the software is intended to achieve—the “operational requirement”—before writing any code.

This article attempts to take an “operational requirement” approach to the “bibliographic reference” aspects of L<sup>A</sup>T<sub>E</sub>X3.<sup>1</sup> The objective is to stimulate debate—if you don’t agree with my suggestions, please suggest specific alternatives! (In the

<sup>1</sup> Obviously, there are limits to the what the OR approach can achieve. For example, it is difficult to quantify “usability”. Nevertheless, the approach should facilitate debate about objectives *before* the “user interface” has been fixed.

remainder of the article, “operational requirement” is abbreviated to “OR”.)

Generalizing the approach taken by the L<sup>A</sup>T<sub>E</sub>X 2.09 manual [1, pp. 73–74], it is convenient to divide the topic into “doing it yourself” and “using bibliography-formatting software”.

## 2 Doing it yourself

In effect, the only scheme that is “fully supported” by L<sup>A</sup>T<sub>E</sub>X 2.09 is “reference by number, where the sequence of numbers is determined by position in the reference-list”.

By contrast, for “real-world publishing”, my impression is that:

1. only a minority of “instructions to authors” specify anything like the default L<sup>A</sup>T<sub>E</sub>X 2.09 scheme. This minority consists of those journals that specify “reference by number, with the reference-list in alphabetical order of author’s names”.
  2. the majority of “instructions to authors”, style-books, etc., specify one of the following:
    - (a) reference by number, with the reference-list in order of first citation
    - (b) author-date
    - (c) “short-form in footnotes”. For publications in the humanities, there seem to be two main variants of this scheme, depending on whether or not there is a reference-list.<sup>2</sup>
- ISO 690 [3, sec. 9] provides a convenient specification of the details of these schemes. The default L<sup>A</sup>T<sub>E</sub>X 2.09 system gives no particular help to anyone wanting to use them.<sup>3</sup>
3. a few publishers specify alternative schemes. E.g.,

<sup>2</sup> If there is no reference-list, the convention is usually “first citation gives full bibliographic details, subsequent citations give cross-reference to first citation”. This variant is common in law publications, when it is used in conjunction with numerous law-specific citation conventions [2].

<sup>3</sup> B<sub>I</sub>B<sub>T</sub>E<sub>X</sub> can help with (a). Anyone wishing to use (b) will probably grope around in archives looking for style-options that: arrange for `\cite` to give ( . . . ) rather than [ . . . ]; omit [ . . . ] from the reference-list; support date-only citations when the author’s name appears naturally in a sentence. Apart from the law-specific Lexi<sub>T</sub>E<sub>X</sub> [4], I’m not aware of any 2.09-related software that helps people who wish to use scheme (c).

- some Springer journals<sup>4</sup> accept citations of the form “first letter of author’s surname, in square brackets”
- Butcher [5] mentions a variation of the reference-by-number system in which there is a separate numerical sequence for each letter, and a variation of the author-date system in which a number is used instead of a date
- a scheme like the BibTEX alpha style is sometimes used (for example, in the journal *Formal Aspects of Computing*).

Therefore, I suggest that the OR for L<sup>A</sup>T<sub>E</sub>X3:

- should include support<sup>5</sup> for the schemes mentioned in items 1 and 2 above, i.e.,
  - a 2.09-like scheme aimed at journals that specify “reference by number, with the reference-list in alphabetical order of author’s names”
  - the schemes specified in ISO 690, namely: “reference-by-number in order of first citation”, author-date, and 2 variations of “short-form in footnotes”.<sup>6</sup>
- should bear in mind the possibility of a “plug-in module” to support law conventions. Since such conventions are crucial only to lawyers, it would probably be inappropriate to delay L<sup>A</sup>T<sub>E</sub>X3 while law-specific commands were finalised, or to increase the bulk of the L<sup>A</sup>T<sub>E</sub>X3

<sup>4</sup> See the “instructions for authors” in, for example, *Mathematische Zeitschrift*.

<sup>5</sup> I assume that “sorting a reference-list” will be beyond the scope of L<sup>A</sup>T<sub>E</sub>X3. Thus, in practice, the L<sup>A</sup>T<sub>E</sub>X3 “support” might be minimal (a “better than nothing” warning that a reference-list needs human intervention, perhaps). People who want anything better would be advised to use bibliography-formatting software.

<sup>6</sup> To support these schemes, it is probably desirable that L<sup>A</sup>T<sub>E</sub>X3 should be able to determine whether a citation of a source is “the first citation” of that source. Clearly this would help to provide support for “reference by number in order of first citation”. In the author-date case, it would allow support for the convention [6, sec. 3.87] that, when there are multiple authors, they should all be named in the first citation but “*et al.*” should be used subsequently. It might also help to provide support for the variant of the short-form scheme in which a “subsequent citation” uses the short-form and gives a cross-reference to the footnote containing the “first citation” (where full details of the source can be found).

manual by including law-specific material. Nevertheless, it might be worth simultaneous experiments with a prototype L<sup>A</sup>T<sub>E</sub>X3 and a prototype law-support module, in the hope that the law-specific commands in such a module might end up with a similar “look and feel” to those for the mainstream “short-form in footnotes” commands.

- need not include support for the alternative schemes mentioned in item 3 above (although the possibility of “plug-in modules” to support these schemes might be borne in mind).

In addition, the following features are desirable:

- for situations where several bibliographic sources are cited simultaneously:
  - a syntax that permits a particular division of each source to be pin-pointed [7, sec. 15.25]. (The L<sup>A</sup>T<sub>E</sub>X 2.09 `\cite{...}{...}` syntax only supports pin-pointing within a single source.)
  - a mechanism for sorting reference-by-number citations into ascending numerical order [8, p. 106].<sup>7</sup>
  - a mechanism for sorting author-date citations<sup>7</sup> into alphabetical order of author’s surnames (or, ideally, the order in which the sources appear in the reference-list) [6, sec. 3.91] or into “date of publication” order [7, sec. 15.24].
- support for types of bibliography that, although not as common as a single undivided list, are appropriate in particular circumstances, namely:
  - a list divided into sections according to kinds of material, subject matter or other appropriate categories
  - an annotated bibliography
  - a bibliographical essay.

See, for example, the *Chicago Manual of Style* [7, chap. 15].

(End-users get confused if they try using L<sup>A</sup>T<sub>E</sub>X 2.09’s `thebibliography` environment for such bibliographies.)

The above might provide the major elements of an OR. Minor elements may be more difficult to specify, but can perhaps be summarized as

<sup>7</sup> Alternatively, if it is not feasible to sort reference-by-number and author-date citations into a desired order, mechanisms for giving warnings if simultaneous citations are in the wrong order would be “better than nothing”.

L<sup>A</sup>T<sub>E</sub>X3 should be able to survive  $\beta$ -testing of whether it can conveniently deliver bibliographic details formatted as specified by influential style-books and “instructions for authors”.

See [2, 3, 5, 6, 7, 9, 10, 11, 12, 13, 14].

### 3 Using bibliography-formatting software

#### 3.1 Background<sup>8</sup>

##### 3.1.1 Software available

The bibliography-formatting software that is “advertised” in the L<sup>A</sup>T<sub>E</sub>X 2.09 manual is BIB<sub>T</sub>E<sub>X</sub> [1, 15]. Tib [16] is also sometimes mentioned in T<sub>E</sub>X circles.

In fact, there are a large number of bibliography-formatting programs available. A recent review article [17] names 52 such programs.

Judging by comments on the `bibsoft` list, the most important bibliographic programs (from the point-of-view of professional librarians and bibliographers) seem to be EndNote, Library Master, Papyrus, ProCite and Reference Manager. (Appendices A and B give details of the `bibsoft` list and of the relevant vendors.)

Of these, EndNote, Papyrus, ProCite and Reference Manager have procedures for processing a “manuscript”, filling in the in-text citations and generating the corresponding reference-list. Although I understand that a similar facility is planned for the next version of Library Master, I don’t know what form this will take. Therefore, when referring to these programs, I will use:

“**main 4**” to mean the programs (EndNote, Papyrus, ProCite and Reference Manager) whose procedures for filling in the in-text citations are currently known

“**main 5**” to mean the “main 4” plus Library Master.

From a L<sup>A</sup>T<sub>E</sub>X-er’s point-of-view, the public-domain BIB<sub>T</sub>E<sub>X</sub> and Tib are obviously attractive, since they were *designed* to work with T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X, and are available for most of the platforms on which T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X are available. By contrast, the “main 5” are:

- proprietary

<sup>8</sup> Warning: I do not currently have “hands on” experience of using L<sup>A</sup>T<sub>E</sub>X in conjunction with software other than BIB<sub>T</sub>E<sub>X</sub> (although I have browsed through as many of the relevant manuals as I could find). Hence, the ideas given in this section, and in section 4, are theoretical and speculative.

- currently aimed at “wordprocessor” users<sup>9</sup>
- only available on a restricted selection of platforms. (All are available for MS-DOS. Some are available for Macintosh or VAX/VMS.)

Nevertheless, there are many things about the “main 5” that are of interest:

- The programs have standard procedures for importing information from standard database programs, online information services, CD-ROMs and library catalogues.
- They generally have good facilities for maintenance of a “personal bibliographic database”, and for searching such a database for entries that satisfy particular criteria.
- It seems likely that the programs will continue to be developed and supported into the future. (By contrast, my understanding is that BIB<sub>T</sub>E<sub>X</sub> will be “frozen” when version 1.0 has been finished.)
- There is a choice. If one program has underlying assumptions that don’t match the assumptions that are usual in your discipline, you can look for an alternative!

Even if you don’t regard the “main 5” as of positive interest, you may be unable to avoid them. If a research-group contains a L<sup>A</sup>T<sub>E</sub>X-ing minority and a non-L<sup>A</sup>T<sub>E</sub>X-ing majority:

- the “majority” may choose one of the “main 5” as the group’s “standard bibliography-formatting software”
- the L<sup>A</sup>T<sub>E</sub>X-ers will then be at a serious disadvantage if they cannot use the group’s bibliographic databases.

Also, if your librarian is providing bibliographic information in electronic form (e.g., from a computerized library catalogue), s/he may offer an off-the-shelf way to get the information into a database for one of the “main 5”, but be unable to help you if you use BIB<sub>T</sub>E<sub>X</sub>.

Overall, it seems to me desirable that, as well as having standard procedures for inter-working with BIB<sub>T</sub>E<sub>X</sub> and Tib, L<sup>A</sup>T<sub>E</sub>X3 should have standard procedures for inter-working with the “main 5”. Such procedures are unlikely to be perfect, but it should be possible to agree on some *modus vivendi*.<sup>10</sup>

<sup>9</sup> Certain vendors state that T<sub>E</sub>X is one of their program’s “supported wordprocessors”. You may or may not regard this as a hopeful sign!

<sup>10</sup> It is unlikely that the vendors will re-focus their products to concentrate on L<sup>A</sup>T<sub>E</sub>X users — and equally unlikely that L<sup>A</sup>T<sub>E</sub>X-ers will start to think of themselves as “wordprocessor users”. Nevertheless,

### 3.1.2 *Modus vivendi* with the main 4?

Before considering how L<sup>A</sup>T<sub>E</sub>X might co-operate with the “main 4”<sup>11</sup> it is convenient to contrast B<sub>I</sub>B<sub>T</sub>E<sub>X</sub>'s approach with that of Tib.

B<sub>I</sub>B<sub>T</sub>E<sub>X</sub>'s approach involves searching a L<sup>A</sup>T<sub>E</sub>X .aux file for details of in-text citations, and then writing out a .bbl file. The .bbl file defines a reference-list that is read in when L<sup>A</sup>T<sub>E</sub>X is next applied to the root file.

Tib's approach is different. It starts with a .tex file that contains “incomplete or keyed citations” within citation-delimiters, and produces another .tex file that contains proper in-text citations plus (optionally) a reference-list.

When the procedures used by the “main 4” are interpreted in terms of L<sup>A</sup>T<sub>E</sub>X, they seem to be more akin to Tib's approach than to B<sub>I</sub>B<sub>T</sub>E<sub>X</sub>'s. It looks as though the end-user would start with a .tex file containing keys, etc., within citation-delimiters, and use the bibliography-formatting program to produce a near-duplicate .tex file that contains proper in-text citations plus a reference-list.

In fact, Tib's citation-delimiters are chosen so that:

The escape characters of Tib do not interfere with T<sub>E</sub>X processing. If T<sub>E</sub>X is applied to the original pre-Tib document, the escape characters and incomplete citations will appear as written.

I.e., the pre-Tib .tex file and the post-Tib .tex file are both valid L<sup>A</sup>T<sub>E</sub>X input files.

This seems a useful precedent. If L<sup>A</sup>T<sub>E</sub>X could inter-work with the “main 4” in an analogous way, it would not be necessary to

Apply bibliography-formatting software.  
Then apply L<sup>A</sup>T<sub>E</sub>X.

every time that a .dvi file is required. For example, if someone is concentrating on getting their equations typeset correctly, they might want to get .dvi files quickly without always having to go through the bibliography-formatting step. At the equation-

with a few minor changes (which might involve the L<sup>A</sup>T<sub>E</sub>X end, the bibliographic program end and/or the documentation), it should be possible for L<sup>A</sup>T<sub>E</sub>X3 and the mainstream bibliographic software to work reasonably well together.

“*Modus vivendi*”, i.e., “an arrangement between peoples who agree to differ”, seems to fit the situation quite well.

<sup>11</sup> Hopefully, it will be possible to use the same general ideas for Library Master when its procedure for “filling in the in-text citations and generating the reference-list” becomes known.

checking stage, they may just want a .dvi file that shows their equations, and not be worried about the appearance of their in-text citations or reference-list.

A potential problem for any L<sup>A</sup>T<sub>E</sub>X-er trying to follow the Tib precedent, is that EndNote and ProCite use # to identify “number within database”. Hence the end-user may need to put a # (which is one of L<sup>A</sup>T<sub>E</sub>X's 10 “special characters”) within the relevant citation-delimiters. (See Table 1 for details of the programs' default citation-delimiters, and the alternatives available.)

One way of imitating Tib (in spite of the possibility of # characters) might be to arrange delimiters such that the proprietary program's “start delimiter” is interpreted by L<sup>A</sup>T<sub>E</sub>X as being equivalent to L<sup>A</sup>T<sub>E</sub>X 2.09's \verb+, and its “end delimiter” is interpreted as equivalent to the + that terminates the text introduced by \verb+. Then:

- if L<sup>A</sup>T<sub>E</sub>X is applied to the original .tex file, the citation keys will be typeset “as is” in a typewriter font (to remind the L<sup>A</sup>T<sub>E</sub>X-er that the bibliographic software needs applying before the document can be regarded as finished)
- if the bibliographic software is applied to the original .tex file, a new .tex file will be produced that, when L<sup>A</sup>T<sub>E</sub>X-ed, has proper in-text citations and a reference-list.

Overall, the L<sup>A</sup>T<sub>E</sub>X-er will be able to apply L<sup>A</sup>T<sub>E</sub>X and the bibliographic software in either order (in much the same way that L<sup>A</sup>T<sub>E</sub>X and Tib can be applied in either order).

This approach could be the major element of a *modus vivendi* between L<sup>A</sup>T<sub>E</sub>X3 and the “main 4”. Table 2 shows some delimiters that might be suitable.

A *modus vivendi* would also need to incorporate an approach to the “root file and \include-ed files” situation. Although I don't have any specific suggestions at this stage, I speculate that support for this feature might be obtained by reference to the bibliographic software's support for analogous features in wordprocessors (e.g., WordPerfect's “master document and subdocument” scheme, and Microsoft Word's “include” scheme).

### 3.1.3 Preferred interface

The suggestions in Table 2 are intended as part of a *modus vivendi* between L<sup>A</sup>T<sub>E</sub>X3 and the *current* versions of the “main 4”. Although the general approach is the same, the details differ from product to product.

It would be open to L<sup>A</sup>T<sub>E</sub>X-ers to decide on a preferred interface, and to inform the vendors of

Software	Citation delimiters	Notes
L <sup>A</sup> T <sub>E</sub> X 2.09 with BIB <sub>T</sub> E <sub>X</sub>	<code>\cite{ }</code>	
Tib	[. .]	The delimiters <. .> are used in some circumstances
EndNote	Default: [ ]	You can tell EndNote to look for alternative 1-character delimiters (e.g., < >)
Library Master	Not known	I understand that a facility for “given the in-text citations, compile a reference-list” is in preparation
Papyrus	Default: %% %%	You can tell Papyrus to look for alternative delimiters (but “start delimiter” must be the same as “end delimiter”)
ProCite	Default: ( )	You can tell ProCite to look for [ ] rather than for ( )
Reference Manager	Default: { }	You can tell Reference Manager to look for alternative delimiters. “Start delimiter” and “end delimiter” can each have up to 7 characters.

Table 1: Citation-delimiters: defaults and alternatives

Biblio. software	Tell bib. software	Tell L <sup>A</sup> T <sub>E</sub> X3	Notes
EndNote	Delimiters are < and >	< ... > is equivalent to 2.09's <code>\verb+ ... +</code>	
Papyrus	Delimiter is "	" ... " is equivalent to 2.09's <code>\verb+ ... +</code>	
ProCite			No obvious alternative to “always apply ProCite before L <sup>A</sup> T <sub>E</sub> X”
Reference Manager	Delimiters are <code>\bsoft{ and }</code>	<code>\bsoft{ ... }</code> is equiv. to 2.09's <code>\verb+ ... +</code>	

*Note:* Clearly the default Papyrus and Reference Manager delimiters (see Table 1) must be changed if the end-user is to have the option of applying L<sup>A</sup>T<sub>E</sub>X without having previously dealt with citations, etc. However, the Papyrus and Reference Manager keys are not liable to contain a # character. Hence, it is not crucial whether Papyrus and Reference Manager keys are “hidden” from L<sup>A</sup>T<sub>E</sub>X.

Table 2: Choice of delimiters for *modus vivendi*?

their preference in the hope that it may be possible to implement the approach more consistently at some time in the future. We wouldn't lose anything by asking!

For example, if the preferred interface involved `\bsoft{key}` (as shown in Table 2 for Reference Manager), it would be open to us to ask the other vendors to relax their rules on citation-delimiters so that future versions of the "main 5" will all accept `\bsoft{key}`. If we were lucky enough to get the vendors' agreement, we might be able to produce notes about "using proprietary bibliographic software with L<sup>A</sup>T<sub>E</sub>X" that would appear simpler to the end-user than Table 2.

**Note** It might be possible to have a *modus vivendi* (e.g., with Reference Manager) involving `\verb+key+`, rather than having an additional command such as `\bsoft` (which would, in any case, be implemented in much the same way as `\verb`). The bibliographic software will probably ignore things within `\verb+` and `+` that don't look like citation keys. Nevertheless, I would be inclined to introduce an extra command (e.g., `\bsoft`) so that `.tex` files can be "marked up logically" to distinguish between:

- delimiters for a key that is intended for processing by bibliographic software
- delimiters for text that is intended to appear in a `typewriter` font in the final document.

### 3.1.4 Hybrid approaches

One can envisage schemes that embed a proprietary bibliographic system's mechanism for dealing with citations and reference-lists within L<sup>A</sup>T<sub>E</sub>X's mechanism (or *vice versa*). Examples might include:

- telling Papyrus to use `!!` as its delimiter, and putting the Papyrus citation markers inside a L<sup>A</sup>T<sub>E</sub>X `\cite` command, thus `\cite{!! ... !!}`.<sup>12</sup>
- trying to get proprietary bibliographic software to read an `.aux` file, and write a `.bbl` file, as BIB<sub>T</sub>E<sub>X</sub> does. (Perhaps this could be done by a shell script which invokes the proprietary software in a suitable way.)

Generally, I fear that such hybrid schemes may lead to confusion, and I would not be inclined to pursue them:

- Anyone constructing a hybrid scheme will have to be very careful about "which software is in charge when" (e.g., whether citation numbers are incremented by L<sup>A</sup>T<sub>E</sub>X, by the proprietary system, or by "one shadowing the other").

<sup>12</sup> Bernard J. Treves Brown, of Manchester University, is experimenting with this technique.

The hybrid scheme will need maintenance (e.g., someone will need to verify that the scheme still works with each new release of the proprietary system). There may be three lots of documentation for the end-user to study: that about L<sup>A</sup>T<sub>E</sub>X3, that about the proprietary system, and that about the hybrid scheme's subtle combination of elements of both. If anything goes wrong, it may be in "a grey area", which is neither the responsibility of the L<sup>A</sup>T<sub>E</sub>X3 project, nor the responsibility of the bibliographic software vendor.

- The proprietary systems seem more akin to Tib than to BIB<sub>T</sub>E<sub>X</sub>. To try and force them into the BIB<sub>T</sub>E<sub>X</sub> stereotype when they are not designed to work like BIB<sub>T</sub>E<sub>X</sub> seems like "asking for trouble". I doubt whether the T<sub>E</sub>X community has the resources to produce interfaces that "make proprietary systems work like BIB<sub>T</sub>E<sub>X</sub>", and I doubt whether the vendors have the inclination to commit such resources.

My instinct is that it would be better to have a simple interface (e.g., conventions such as those outlined in Table 2), so as to put the end-user in a situation where responsibilities are clear:

- typesetting is the responsibility of L<sup>A</sup>T<sub>E</sub>X3
- bibliography-generation is the responsibility of the bibliographic software.

Hence, if using a proprietary bibliographic system, the end-user should ignore the L<sup>A</sup>T<sub>E</sub>X3 manual's descriptions of commands to support the DIY-er (i.e., ignore the L<sup>A</sup>T<sub>E</sub>X commands envisaged in section 2), and ignore anything that is provided to support the BIB<sub>T</sub>E<sub>X</sub>-er.

- The proprietary system will be "in charge" of bibliography generation. The method used will be that envisaged by the vendor, and documented in the vendor's manual: if it's good, the vendor will get the credit; if it's bad, the vendor will get the blame.
- The delimiters in the `.tex` file will be delimiters for the proprietary system (chosen, if possible, in such a way that the `.tex` file is acceptable to L<sup>A</sup>T<sub>E</sub>X even before processing by the proprietary system). They might be as shown in Table 2. The "keys", etc., inside the delimiters will follow the rules given in the vendor's manual (*not* the rules given in the L<sup>A</sup>T<sub>E</sub>X3 manual about keys that the DIY-er can use).
- The proprietary system will be "told to produce T<sub>E</sub>X output". How good or bad they are at this will be the responsibility of the proprietary system (although interested L<sup>A</sup>T<sub>E</sub>X-ers might advise the vendors about what is required).

Overall, the end-user will get in-text citations filled in, and reference-lists generated, in the standard way that is described in the manual that describes the proprietary system. If this standard way does not suit a L<sup>A</sup>T<sub>E</sub>X-er's requirements, it may be better for him/her to seek alternative bibliography-formatting software rather than spending time trying to circumvent the problems.

Of course, if people want to put effort into developing hybrid schemes, and happen to get good *modus vivendi* between L<sup>A</sup>T<sub>E</sub>X and proprietary bibliographic systems, I would be delighted to find that my instinct is wrong!

### 3.1.5 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.

- Cost is obviously a factor.
- An end-user who wants software that has been designed specifically for use in conjunction with L<sup>A</sup>T<sub>E</sub>X, will probably be inclined to choose B<sup>I</sup>B<sup>T</sup>E<sub>X</sub> or Tib.
- B<sup>I</sup>B<sup>T</sup>E<sub>X</sub>'s approach makes good use of disk-space. A .bbl file will be smaller than "near-duplicates of .tex files".
- An end-user who wants ready-made methods of downloading information from commercial bibliographic databases, library catalogues, etc., will probably favour one of the proprietary programs. The proprietary systems also offer database administration and searching facilities.
- Anyone who does not have the time and patience to deduce (from a proprietary system's wordprocessor-oriented documentation/menus) what the L<sup>A</sup>T<sub>E</sub>X-er should do might prefer to wait until someone else has deduced what is required, and has documented the tricks involved.
- The end-user's choice may be constrained by the platform on which they are using L<sup>A</sup>T<sub>E</sub>X (e.g., they may need bibliographic software for a Unix system).
- Wordprocessor-oriented systems may not support typesetting subtleties to the degree that L<sup>A</sup>T<sub>E</sub>X-ers would like.
- Support (or lack of it) for non-English languages may be another factor.<sup>13</sup>

<sup>13</sup> Decisions may be needed about whether to try using a proprietary system's support for diacritics, in the hope of being able to share a database with colleagues who use wordprocessors. The alternative would be to have database entries that use T<sub>E</sub>X encoding for diacritics.

- End-users may be constrained to use the same system as other people in their research group (e.g., so that the group can share databases).

It is unlikely that anyone will find bibliographic software that is perfect for their needs. However, people are more likely to find something that suits them if they have a choice than if they have no choice.

## 3.2 OR for L<sup>A</sup>T<sub>E</sub>X3

Given the situation outlined in section 3.1, I suggest the following as the OR for L<sup>A</sup>T<sub>E</sub>X3's relationship with bibliography-formatting software:

- As far as practicable, L<sup>A</sup>T<sub>E</sub>X3 should be neutral towards the end-user's choice of bibliography-formatting software. Ideally, people should be able to choose typesetting software for typesetting reasons, and bibliographic software for bibliographic reasons — their choice of typesetting software should not restrict their choice of bibliographic software.
- Hence, a *modus vivendi* between L<sup>A</sup>T<sub>E</sub>X3 and each of the "main 5" should be thought up, tested and documented.<sup>14</sup>
- There might be "a preferred interface" between L<sup>A</sup>T<sub>E</sub>X3 and proprietary bibliographic software. It vendors can be persuaded to support this interface, L<sup>A</sup>T<sub>E</sub>X-ers will get a consistent interface to proprietary bibliographic software. If not, things will stay inconsistent (e.g., as shown in Table 2).
- In line with the neutrality suggested above, B<sup>I</sup>B<sup>T</sup>E<sub>X</sub> will continue to be supported, but L<sup>A</sup>T<sub>E</sub>X3 documentation will not be particularly pro-B<sup>I</sup>B<sup>T</sup>E<sub>X</sub>. It is desirable that .bst files should be updated so that B<sup>I</sup>B<sup>T</sup>E<sub>X</sub> produces L<sup>A</sup>T<sub>E</sub>X3 commands (designed to satisfy the requirements listed in section 2) rather than L<sup>A</sup>T<sub>E</sub>X 2.09 commands.

## 4 Miscellaneous

### 4.1 "Local names" for keys

If you are "doing it yourself", choice of keys is unlikely to be a problem. For example, you could equally well use `limport-86` or `latexbook` as a key for the L<sup>A</sup>T<sub>E</sub>X manual. There is no particular need for consistency from one document to another: you

<sup>14</sup> The *modus vivendi* might be along the lines shown in Table 2, or might be something else that emerges from practical experience. It doesn't matter much whether the documentation is provided by the L<sup>A</sup>T<sub>E</sub>X3 project or by the bibliography software vendor, as long as someone provides it!

can use `lampoort-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 L<sup>A</sup>T<sub>E</sub>X commands that are generated automatically by bibliographic software (even though L<sup>A</sup>T<sub>E</sub>X will have no way of distinguishing the file from a one typed in by a DIY-er). Such bibliographic software might be programmed to assign keys automatically. For example, software might write a `.tex` file that contains L<sup>A</sup>T<sub>E</sub>X 2.09 `\bibitem` commands, with keys of the form `lampoort-86` constructed automatically from two fields in the database.<sup>15</sup>

To help cater for such situations, it might be useful if L<sup>A</sup>T<sub>E</sub>X3 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 `latexbook` can be used as a “local name” for `lampoort-86`.

#### 4.2 Reference-lists that are also indexes

Another requirement that needs to be borne in mind is for reference-lists which, as well as providing bibliographic details of sources, provide an index to the pages on which the sources are cited:

- in mainstream academic publications, the requirement will probably be for a “combined list of references and author index” [5, pp. 198 & 258]
- in law books, the requirement is usually for “front matter” units such as “table of cases”, “table of statutes” and “table of treaties”. In a typical “table of cases”, each entry tells the reader

<sup>15</sup> Some thought would need giving to any method of assigning keys automatically. If a bibliographic database is continually growing, there may be no guarantee that keys of the form `lampoort-86` will stay unique when new items are added to the database. It might be safer to assign less memorable keys that can be guaranteed to stay distinct, e.g., the “record number” in the database, or a book’s ISBN

- where further details of the case can be found (e.g., the relevant law report)
- which pages in the book’s main text mention the case.

The other types of tables are analogous.

#### A Some suppliers of mainstream bibliographic software

**EndNote** Niles and Associates. 2000 Hearst St, Berkeley, CA 94709, USA. E-mail: `nilesinc@well.sf.ca.us`.

**Library Master** Balboa Software, P. O. Box 3145, Station D, Willowdale, Ontario, M2R 3G5, Canada. E-mail: `hahne@epas.utoronto.ca`.

**Papyrus** Research Software Design, 2718 S. W. Kelly St, Suite 181, Portland, Oregon 97201, USA. E-mail: `RSD@applelink.apple.com`.

**ProCite** Personal Bibliographic Software, P. O. Box 4250, Ann Arbor, Michigan 48106. E-mail: `sales@pbsinc.com` or `support@pbsinc.com`.

**Reference Manager** Research Information Systems, Camino Corporate Center, 2355 Camino Vida Roble, Carlsbad, CA 92009, USA. E-mail: `sales@ris.risinc.com`.

#### B E-mail discussion lists about bibliographic software

The `bibsoft` list provides a forum for general discussion of personal bibliographic database management systems. You can subscribe by sending a one-line e-mail message of the form

```
subscribe bibsoft last-name,first-name
to
listserv@indycms.iupui.edu.
```

There are also specific discussion lists for EndNote, Library Master and ProCite. See [17].

In the United Kingdom, there is a discussion list for Higher Education institutions that have taken up the CHEST Papyrus deal. You can subscribe by sending a one-line message of the form

```
subscribe
chest-papyrus first-name last-name
to
mailbase@mailbase.ac.uk.
```

#### References

- [1] Leslie Lamport. *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*. Addison-Wesley, 1986.
- [2] *The Bluebook: A Uniform System of Citation*. Harvard Law Review Association, 15th edition, 1991. Obtainable from: Harvard Law Review Association, 1511 Massachusetts Avenue, Cambridge, Massachusetts 02138.



- [3] Documentation — bibliographic references — content, form and structure. ISO 690, International Organization for Standardization, 1987.
- [4] Frank G. Bennett, Jr. *LexiTeX*: a  $\text{\LaTeX}$  macro package for lawyers. Document deposited in electronic archives, 1993.
- [5] Judith Butcher. *Copy-editing*. Cambridge University Press, 3rd edition, 1992.
- [6] *Publication Manual of the American Psychological Association*. American Psychological Association, 3rd edition, 1983. Obtainable from: American Psychological Association, P. O. Box 2710, Hyattsville, MD 20784.
- [7] *The Chicago Manual of Style*. University of Chicago Press, 13th edition, 1982.
- [8] Janet S. Dodd. *The ACS Style Guide*. American Chemical Society, 1986.
- [9] *MHRA Style Book*. Modern Humanities Research Association, 4th edition, 1991.
- [10] Joseph Gibaldi and Walter S. Achtert, editors. *MLA Handbook for Writers of Research Papers*. Modern Language Association of America, 3rd edition, 1988.
- [11] International Committee of Medical Journal Editors. Uniform requirements for manuscripts submitted to biomedical journals. *British Medical Journal*, 302:340–341, February 1991. Note: This article was also published in the *New England Journal of Medicine* (7th Feb. 1991). It specifies the “Vancouver style” for manuscript-preparation, which is accepted by over 400 journals.
- [12] Citing publications by bibliographic references. BS 5605, British Standards Institution, 1978.
- [13] References to published materials. BS 1629, British Standards Institution, 1989.
- [14] Citation of unpublished documents. BS 6371, British Standards Institution, 1983.
- [15] Oren Patashnik. *BibTeXing*. Document deposited in electronic archives, January 1988.
- [16] James C. Alexander. *Tib*: A  $\text{\TeX}$  bibliographic preprocessor. Document deposited in electronic archives, 1989.
- [17] Sue Stigleman. Bibliography formatting software: an update. *Database*, February 1993.

◊ David Rhead  
 Cripps Computing Centre  
 University of Nottingham  
 University Park  
 Nottingham NG7 2RD England,  
 U.K.  
 David\_Rhead@vme.ccc.nottingham.ac.uk

## Relative moves in $\text{\LaTeX}$ pictures

Richard Bland

### 1 Introduction

In this note I hope to do three things:

1. Make a number of observations about why picture-drawing in  $\text{\LaTeX}$ , as described by Lamport, is so difficult and unpleasant.
2. Put forward a suggestion for a very simple mechanism to overcome at least some of these difficulties.
3. Show one way of implementing this suggestion, using the Unix utility `m4`. This particular implementation is presented only to demonstrate the simplicity of the underlying mechanism: no claim is made that it is an optimal implementation.

### 2 An example

Consider the simple picture in Figure 1. As is obvious, this picture has no meaning: it is just a collection of graphic elements such as labelled shapes, text strings, lines and arrows: but it does exemplify the kind of output which many users have in mind when they set out to draw a picture in  $\text{\LaTeX}$ . Such users want some form of diagrammatic representation in which different shapes are used to represent types of entity, lines and arrows are used to connect the entities, and labelling is used to give some domain-specific meaning. Often these pictures are conceptually quite simple.

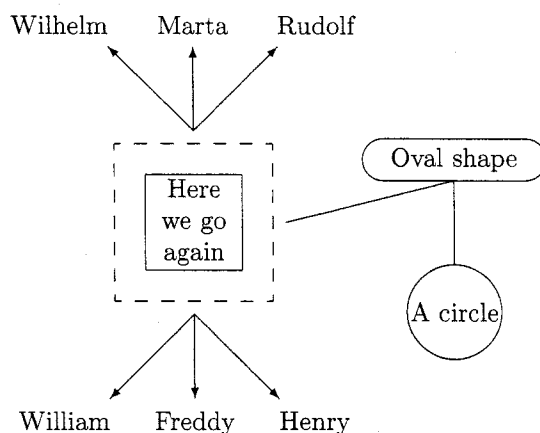


Figure 1: A  $\text{\LaTeX}$  picture