

BIB_TE_X Tips and FAQ

Michael Shell* David Hoadley†

January 2007, Version 1.1

Contents

1	Introduction	1
2	Helpful Tips	2
2.1	General Usage Tips	2
2.2	Citation Tips	3
2.3	Internet and Other Special Reference Tips	5
2.4	Bibstyle Design Tips	6
3	Frequently Asked Questions	6
3.1	General Usage Questions	6
3.2	Citation Questions	12
3.3	Internet and Other Special Reference Questions	12
3.4	Bibliography Modification Questions	13
3.5	Bibstyle Design Questions	15
3.6	Software Application Questions	19
	Acknowledgments	20
	References	21

1 Introduction

THIS document should be considered as a supplement to the other sources of information about BIB_TE_X and its interaction with L_AT_EX, especially the primary L_AT_EX [2, 3] and BIB_TE_X documentation [4], [2, Appendix B]. Nicolas Markey’s comprehensive *Tame the BeaST — The B to X of BIB_TE_X* [5] is a most excellent tutorial on BIB_TE_X. Ki-Joo Kim’s *A Bib_TE_X Guide via Examples* [6] may also be of help. Furthermore, Robin Fairbairns’ T_EX FAQ [7] has several entries of issues related to BIB_TE_X bibliographies and citations. Finally, Helge Kreuzmann’s *How to Efficiently Use BIB_TE_X* is worthy of mention [8].

*See <http://www.michaelshell.org/> for current contact information.

†Manuscript created on January 10, 2007 based on a post by David Hoadley on December 15, 1998 in the thread “Do we need a BibTeX FAQ?” in the newsgroup comp.text.tex. The latest version of this document can be obtained on CTAN [1]. This work is distributed under the L_AT_EX Project Public License (LPPL) (<http://www.latex-project.org/>) version 1.3. A copy of the LPPL, version 1.3, is included in the base L_AT_EX documentation of all distributions of L_AT_EX released 2003/12/01 or later. The opinions expressed here are entirely that of the authors. No warranty is expressed or implied. User assumes all risk.

Below is a compilation of helpful practical tips and answers to frequently asked questions primarily concerning the use and operation of $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$, but with some related issues on bibliographies and citation as well. It should be noted that some of the advice given here is opinion and should not be taken as unbreakable rules. Nevertheless, by following the tips here, you may be able to avoid some of the problems that many $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$ users often later confront as they use $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$ and related tools in increasingly advanced ways.

2 Helpful Tips

2.1 General Usage Tips

1. Put the authors' full names (and other information) into database whenever you can.

If you want only initials in the reference list, use an appropriate bibstyle (or create one, see [Tip 14](#)) to generate this. One broad principle to remember when entering information into a $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$ database is that you should err on the side of including too much information rather than too little. Database information can always be omitted according to the $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$ style file (which is often called a “bibstyle,” or “.bst file” to distinguish it from a $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ style file), but missing information can't be generated.

2. Put all of the authors' names into the author list.

That is, do not put “and others”, unless you don't actually know their names. You never know when you might want to search for one of these “other” authors' name(s). If you want “et al.” to appear in the reference list instead of all the names, use an appropriate bibstyle to generate this (or create one, see [Tip 14](#)).

3. Put full stops (periods) after abbreviated words in journal names.

If you are required to supply abbreviated journal names without the full stops, use a bibstyle to create this ([Tip 14](#)).

4. Special characters are entered as `{\a}` or `{\{a}}` not `\{a}`.

This is actually documented in the $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$ user's manual [4], but it still trips people up. For more information, see [Question 5](#).

5. Design for multiple database files.

Assume that you will eventually end up with more than one $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$ database (.bib) file (perhaps with each covering a different subject area). You should therefore ensure that the key you choose for each item contains a part that identifies the file, so that there is no chance of ambiguous or duplicated keys when using multiple database files simultaneously.

For example, say you have some electrical engineering references, in file `elec.bib`¹, that are of the form `@article{elec:smith91,...}` and have some software ones, in file `soft.bib`, that are of the form `@article{soft:smith91,...}` (assuming these are two different articles, probably by different Smiths). Don't put the same reference in more than one file. It may be a good idea to use a unique, private, id prefix, such as `MY_`, to help prevent namespace clashes with publicly released databases produced by others.

6. Put your usual journal names into @string definitions, in a separate (.bib) file.

The standard bibstyles (plain, abbrev, unsrt, alpha) do contain some journal names, all computing-related. This is not really a sensible place because bibstyles are intended to set the format for the publication you are

¹Note that it is the convention of this document not to hyphenate command or file names and to display them in typewriter font. Within such constructs, spaces are not implied at a line break and will be explicitly carried into the beginning of the next line.

contributing *to*, not keeping a complete list of the publications you read and cite *from*. But, if you want the ability to choose full or abbreviated titles, you must go one step further and create two files. For example, in one file (say, `jrnlfull.bib`) you might have:

```
@string{ieee-pas="{IEEE} Transactions on Power Apparatus and Systems"}
```

and in the other (say, `jrnlabbr.bib`):

```
@string{ieee-pas="{IEEE} Trans. Power App. Syst."}
```

and many other entries. Note that both have the same id-strings. To select full or abbreviated titles, you can place the appropriate file (first) into the `\bibliography{}` list (Tip 8).

IEEE authors should take note that (almost) all IEEE journal name definitions, both full and abbreviated, are included with the IEEEtran bibstyle package [9]. Users should avoid (re)defining journal name strings that are covered by existing official string definition files already in public distribution. Furthermore, take care to use id-strings that are unlikely to clash with those provided by official files from other users (`ieee-pas` in the previous example is a poor choice in this regard). As with database entry keys, using a unique, private, id prefix, such as `MY_`, will help to prevent problems with such namespace clashes.

7. Create separate files for cross-referenced entries.

Typically this will be for conferences (proceedings) or books. Again, you may wish to create two string definition files, one with full and one with abbreviated titles (if the titles in the cross referenced entries are not all handled by your existing string definition files). Because cross-referenced entries must appear *after* entries that reference them, it is better to put them in a separate file. Then you need not be worried about using any database access software that might sort or re-order your database entries.

8. Use the form `\bibliography{jrnl-names,other-database(s),cross-entries}`.

This ensures that all `@strings` are defined first and that all the cross-referenced entries appear after any entries that reference them. The selection of full or abbreviated titles is just a matter of selecting the appropriate files for this list. With this method, the abbreviation of titles does not need to be coupled to the abbreviation of authors' names to initials (the way standard bibstyle `abbrv.bst` works) so you will have much more flexibility.

2.2 Citation Tips

One of the aims of \LaTeX is to enable the author to think about the structure of a piece of writing rather than its visual appearance. This principle should also apply to citations—one should be able to write without considering the citation format until later. Unfortunately, this is not always possible. Indeed, one of the reasons for Patashnik's dislike of author-date referencing is that it affects one's writing style, encouraging vagueness and a passive voice [4, section 4]. However, if you have to use author-date referencing, it *might* be worthwhile to try to structure your work so that the citation format can be easily changed at will. The following points relate to the appearance of the citations in the text and to writing for numerical and/or author-date referencing styles.

9. Place `\cite` commands ahead of any punctuation like commas or full stops (periods), not just after them.

This applies even if you are intending to use superscripts for your references, despite the fact that superscripts should appear after such punctuation (Tip 10).

10. Use Donald Arseneau's `cite.sty` or `overcite.sty` [10].

If you are using a numerical referencing style (the standard form) obtain and use one of these packages.

Cite.sty and overcite.sty both order index numbers and replace consecutive numbers with a range (e.g., [7,9,5,6,4,3] becomes [3-7,9]). They also allow you to customize the punctuation and spacing of the citation numbers. Cite.sty will fix up any spacing problems, and overcite.sty will place superscripts after any punctuation (which is the correct behavior) even when you have followed [Tip 9](#). Note that as of version 4.0 (2003-05-27) cite.sty contains all the functionality of overcite.sty. To invoke overcite mode, simply load cite.sty with its `superscript` package option.

11. Use Patrick Daly's natbib.sty (or Peter Williams' harvard.sty, etc.) for author-date references.

If you use an author-date style of citations, your choice of style will probably be dictated by the publications you write for. If there is an existing bibstyle that exactly suits your needs, use it in conjunction with the most recent author-date package(s) that support it. Otherwise, use `makebst` ([Tip 14](#)) to generate a suitable bibstyle that can then be used with `natbib.sty` or `harvard.sty`.

`Natbib` is the most recent, powerful and widely used author-date citation package [11]. `Harvard.sty` [12] is also commonly used and respected — although potential users should be warned that it was last updated in 1995. There may be limitations and problems when using author-date packages older than these. For instance, if you are also using `chapterbib.sty`, `harvard.sty` works without alteration, but `chicago.sty` [13] must be modified according to the instructions in `chapterbib.sty`.

If you must use an older author-date package, you might still want to use `makebst` to create a modern bibstyle that is superior to the original. For example, `chicago.bst` does not handle cross references from inproceedings entries to proceedings entries correctly, but its `makebst` equivalent does.

Don't bother with `authordate1-4.sty` because these have been vastly superseded by the more recent and/or powerful packages such as `natbib.sty`, `harvard.sty`, etc.

12. Writing for both numerical and author-date referencing styles:

In principle, you should write so that you can change the referencing style when required without re-writing everything. A standard citation presents no problems:

The phase-coordinate representation [18] requires ...
 The phase-coordinate representation (Laughton 1968) requires ...

A standard `\cite` command will produce whichever of these is determined by the style/bibstyle combination. But, problems can arise when the authors' names are integrated into the text:

Laughton [18] developed the theory ...
 Laughton (1968) developed the theory ...

The author-date styles provide a means of generating both the authors' names and the date in this form, using variations of the `\cite` command, but the standard numerical styles do not. There are two approaches to getting around this problem when writing for both styles.

One is not to use the facilities of author-date styles for supplying the names, but to type the names into the text explicitly just as you would have done if using only the numeric form of citation. This means you must check the consistency and spelling of the authors' names yourself. Following the authors' names must be a citation command that (if using numerical referencing) will generate a reference number in the usual way, but if using author-date referencing, will generate only a date (not a full reference). The first requires a `\cite` command, the second a `\citeyear` command from `natbib.sty`, `harvard.sty`, `chicago.sty`, etc. (`Natbib` also has a separate `\citeyearpar` command to get the year within parentheses.) This can be done by defining a new command, say `\citey`, and using that:

Laughton `\citey{soft:laughton68}` developed the theory ...

For numerical referencing, you would define it as:

```
\newcommand{\citey}{\cite}
```

and for author-date referencing, it would be:

```
\newcommand{\citey}{\citeyearpar}% natbib
```

or:

```
\newcommand{\citey}{\citeyear}
```

The second approach to this problem is to use a style that can supply authors' names integrated into the text—even when using numerical referencing. Patrick Daly's `natbib.sty` can do this. `Natbib` is very flexible in terms of citation style and it offers many forms of citation commands to choose from. In numerical mode, the standard `\cite` command produces a normal numeric reference. However, in author-date mode, it does not produce a standard reference, but instead places the author's name outside the parentheses. For example, the standard `\cite` command will produce the following when using `natbib`:

The phase-coordinate representation [18] requires ...
 The phase-coordinate representation Laughton (1968) requires ...

which is not what is desirable for transparent switching between numerical and author-date referencing styles.

To get a reference with the authors' names in the text in both numerical and author-date referencing modes, the `\citet` command is provided. Page references can be accomplished by using `\citet`'s second optional argument, e.g., `\citet[p. 35]{soft:laughton68}`:

... the opposite view is taken by Laughton [18, p. 35] ...
 ... the opposite view is taken by Laughton (1968, p. 35) ...

It should also be mentioned that `natbib.sty` does not provide all of the functionality of `cite.sty` or `overcite.sty` in numerical mode. It does order the numbers and calculate ranges, but it does not fix spacing problems. In superscript form, it does not move the superscripts to the correct side of any punctuation the way that `overcite.sty` does.

Some authors prefer to use the first approach to this problem (typing the authors' names explicitly and defining a `\citey` type command), while other authors prefer the second (`Natbib`) option.

Thanks to Jody Klymak for posting some information to the newsgroup `comp.text.tex` on December 14, 1998 in the thread "Do we need a BibTeX FAQ?" that was used to revise this section.

2.3 Internet and Other Special Reference Tips

13. If you must include URLs, use Donald Arseneau's `url.sty` to provide good formatting of them.

Think carefully about references to World Wide Web or other internet-based addresses. Your readers should be able to follow up the references you give them. If they might be reading years after you have written, will the URLs you give still exist? Will the contents at those addresses still be the same, or might someone have modified them? If you decide that it is appropriate to supply URLs (or e-mail addresses, or whatever), use `url.sty` [14] to control the format of them.

Recent versions of `makebst` (Tip 14) can produce bibstyles that support a URL field. `Harvard.sty` also recognizes a URL field, and formats it using the `\harvardurl` command. The results are not as good as those produced by `url.sty`. However, `url.sty` can be used with `harvard.sty`. To do so, place

```
\renewcommand{\harvardurl}{URL: \url}
```

in your document preamble somewhere after loading `harvard.sty`. If you are using a bibstyle that does not recognize the URL field, you'll have to place `\url{your-URL-string}` into the note field (or make a new bibstyle with `makebst`).

The recent version of `hyperref.sty` also provides a good `\url` command [15]. Also, see Section 3.3.

2.4 Bibstyle Design Tips

14. Use Patrick Daly's `makebst` package to create custom `.bst` styles.

Not only can you make bibstyles to order, but `makebst`-created versions of the standard bibstyles are often better than the originals. Your bibstyles can strip periods from journal names (Tip 3), or generate limited author lists followed by “et al.” (Tip 2) and many, many other variations. `Makebst` is part of the custom-bib package [16].

3 Frequently Asked Questions

3.1 General Usage Questions

Q1: Where do I go for support for `BIBTEX`?

`opbibtex@cs.stanford.edu` no longer works. Instead, post your question to the Usenet group `comp.text.tex`.

Q2: When will the next version of `BIBTEX` (e.g., `BIBTEX 1.00`) be released?

When Oren is finished with it. ;) In the meantime, the above tips should be enough to get `BIBTEX 0.99c` to serve as needed. The slides from Oren's 2003 TUG talk on the future of `BIBTEX` are available online [17] for those of you who are interested.

Q3: Where can I find collections of `BIBTEX` database files?

There are many of them scattered among the various schools and societies—use your favorite internet search engine to find those most relevant to your work. However, Nelson H. F. Beebe's `TEX` User Group Bibliography Archive stands apart as one of the largest `BIBTEX` database collections ever assembled [18].

Q4: When should I enclose my `.bib` file entry fields with `"` and when should I enclose them with `{ }` and why?

Functionally, it does not matter whether braces or quotes are used to delimit the entire field. Some people prefer to use quotes for aesthetic reasons. However, with the current versions of `BIBTEX`, if you need to use a `"` character within the field, then you'll have to enclose the `"` character in braces or use braces as delimiters for the entire field. In `BIBTEX`, there are no other valid field delimiters besides quotes and braces.

Q5: I am confused about the difference between special characters and the use of braces to protect text from case changes. Also, when should I “shield” things with braces?

The rules that govern all this are simple, but confusing. `BIBTEX` considers everything within a `{\ .. }` construct at brace level 0 (and *only* brace level 0), that is the top level of bracing of the field (which is not affected by whether quotes or braces are used to delimit the entire field), to be a “special character” and will treat it as though the entire construct is a single character. Within special characters, control sequences (`LATEX` commands) will be preserved as is, but all other text may be case changed or otherwise processed as needed. Furthermore, within special characters, additional levels of braces do *not* increment the brace level. On the other hand, `{ .. }` constructs at brace level 0 (the key here is that a `\` does not immediately follow the opening brace—if so, it would make it a special character) do increment the brace level as well as do nested braces within them. All text and control sequences at brace level 1 or higher is protected from case changes or other processing.

It is perhaps easier to understand from an example. Consider:

```
title = "L0 {\relax S0 {S0 {S0}}} L0 {L1 {\relax L2 {L3}} L1} \LaTeX\ L0"
```

where L_x indicates brace level x and S_x indicates a (part of a) special character at brace level x . Again, nothing would change here if braces had been used to delimit the entire field instead of quotes. The three S_0 's are all considered to be part of the same special character because they are all within a $\{\backslash \dots\}$ construct at brace level 0. Text at S_0 may be case changed, but the \backslashrelax as well as other control sequences in S_0 will not be changed. Note that the additional nested braces within the special character do not increment the brace level. Thus, there is no such thing as S_1 or higher. The first L_1 is at brace level 1 because it is a $\{\dots\}$ (and not a special character). Note that in this case, additional levels of braces do increment the brace level counter. Furthermore, the second $\{\backslashrelax \dots\}$ is *not* treated as a special character because it occurs at brace level 1. All characters and control sequences at L_1 and higher will be protected from case changes. Note that in this example the control sequence \backslashLaTeX is subject to case changes. Thus, if the bibstyle set the title to lowercase, the resultant \backslashlatex command would likely generate an error.

With all this in mind, we can look at some practical examples. Consider:

```
title = "Secrets of {NASA}"
```

NASA needs to be enclosed in braces because it is an acronym that must remain in uppercase. Likewise, we usually need to protect math from case changes:

```
title = "The {\$A\_beta$} Protocol"
```

Simple argumentless \LaTeX commands are also easily protected:

```
title = "The {\LaTeX} Book"
```

However, note that in this case, \backslashLaTeX will be treated as a special character and so the text “ \backslashLaTeX ” will not be considered when sorting. Thus, in cases where the name of the \LaTeX command is identical to the text it represents, it may be better to use an extra set of braces so that the letters that make up the command will be taken into consideration when sorting:

```
title = "The {\{\LaTeX\}} Book"
```

Thanks to Nicolas Markey for clarifying this issue [5].

Another application of brace shielding is with titles in languages (such as German) in which title capitalization must be preserved with some words (such as nouns and names):

```
title = "{M}essung von {S}t{\o}rpfeldern an {A}nlag  
und {L}eitungen der {T}elekkommunikation im  
{F}requenzbereich 9 {kHz} bis 3 {GHz}";
```

Note that $\{\backslasho\}$ is treated as a special character and the “o” is not protected against case changes. However, the first letter of the nouns are protected because they are at brace level 1.

Things get a bit more complicated when a \LaTeX command has an argument. The correct approach depends on whether the argument needs to be protected from case changes. Consider:

```
note = "Volume~2 is listed under Knuth \cite{TEX:book}"
```

if the bibstyle changes the note field to lowercase, we will get:

```
volume~2 is listed under knuth \cite{TEX:book}
```

so we will likely want to enclose first letter of Knuth’s last name in braces. Furthermore, if we had an unusual bibstyle that rendered the note field in uppercase, we would get:

```
VOLUME~2 IS LISTED UNDER KNUTH \CITE{TEX:book}
```

which would result in an error when the nonexistent \backslashCITE is executed. We might be tempted to try something like this:

```
note = "Volume~2 is listed under Knuth {\cite}{TEX:book}"
```

but this won’t work because the extra braces around the \backslashcite command will prevent it from seeing its argument:

VOLUME~2 IS LISTED UNDER KNUTH `{\cite}{TEX:book}`

Instead, we might try something like this:

```
note = "Volume~2 is listed under {K}nuth {\cite{TEX:book}}"
```

However, this is not safe either because the cite key “TEX:book” is now considered to be part of a special character and so it may be case changed (just like the second `SO` in the example before)! Therefore, we need to employ an additional set of braces to get the `\cite` command and its argument to brace level(s) greater than zero:

```
note = "Volume~2 is listed under {K}nuth {{\cite{TEX:book}}}"
```

so as to ensure everything will work regardless of what the bibstyle does to the note field.

It is usually a good idea to let the .bst file convert/format the fields as it sees fit — so don’t force things with extra braces unless you have to. Future versions of `BIBTEX` may be more intelligent with respect to case changing and thus may require fewer “manual interventions” with braces.

Special thanks to Oren Patashnik for contributing much material to this answer.

Q6: Where can I get more information about the entry and formatting of names with `BIBTEX`?

There is an excellent summary of this topic in Nicolas Markey’s *Tame the Beast — The B to X of `BIBTEX`* [5] in the section titled “The author field”.

Q7: How do I handle the case of multiletter initials in author names (transcribed initials)?

This is covered in the `TEX` FAQ [7, “Transcribed initials in `BIBTEX`” entry]. However, a bit of reiteration here may be of help. The solution is to use `BIBTEX`’s special character mechanism so that the multiple letters will be treated as a single character. `\relax` can be employed to get the needed `BIBTEX` special character `\` sequence:

```
author = "S. {\relax Yu}. Epifanov and A. A. Vigasin",
```

(this example borrows the names from the example in the `TEX` FAQ) and thus avoiding the need to store the multiletter initial in an external `LATEX` macro which would have to be declared using the `BIBTEX`’s `@pre` mable mechanism.

Q8: How do I handle the case of compound last names. For example, “Karen Spärck Jones” where the last name is “Spärck Jones”?

There are two approaches. The first is to use the “last, first” entry format:

```
author = "Sp{\a}rck Jones, Karen",
```

The second approach is to use braces to hide the spaces between the names:

```
author = "Karen Sp{\a}rck{ J}ones",
```

or

```
author = "Karen Sp{\a}rck{ }{J}ones",
```

The braces around the “J” serve to protect it from case changes.

Thanks to Tristan Miller, David Eppstein and Donald Arseneau for posting this solution to the newsgroup `comp.text.tex` on October 29, 2004 in the thread “BibTeX, accents, AND compound names [Bug?]”.

Q9: How do I handle the case of a “von” part that is capitalized? For example, “Maria De La Cruz”?

You have to hide the capitals from `BIBTEX`:


```
author = "Maria {\uppercase{d}e La} Cruz",
```

Thus, $\text{BIB}\text{T}_{\text{E}}\text{X}$ will see the “de La” as the “von” part, and $\text{L}\text{T}_{\text{E}}\text{X}$, via `\uppercase`, will restore the “D”. Thanks to Sven Bovin for posting this solution from [19] to the newsgroup `comp.text.tex` on June 29, 2001 in the thread “BibTeX Author “von” token”.

However, Nicolas Markey points out [5] that this may not be an entirely satisfactory solution if an “alphanumeric” style is used as the label would be “[De LaC]”.² Another approach would be to use

```
author = "{\uppercase{d}}e {\uppercase{l}}a Cruz, Maria",
```

which would result in the label “[DLC]”. However, the simplest approach is

```
author = "{D}e {L}a Cruz, Maria",
```

which works because $\text{BIB}\text{T}_{\text{E}}\text{X}$ looks only at characters at brace level zero when determining what part a name belongs to.

Q10: How do I handle the case of a middle or last name that has a part that looks like a “von” name? For example, “Ithiel de Sola Pool”, where “de Sola” is his middle name (and should be abbreviated as “I. de S. Pool”) or “Carlos Canudas de Wit”, where “Canudas de Wit” is the last name (as is common with Spanish names)?

You can use the special character trick to join the names:

```
author = "Pool, Ithiel {\relax de S}ola",
```

or, if using the “first name first” entry form, you have to also hide the lowercase “d” from $\text{BIB}\text{T}_{\text{E}}\text{X}$:

```
author = "Ithiel {\lowercase{D}e S}ola Pool",
```

For the latter case, simple braces will suffice to prevent the “d” from being seen as a “von” part because $\text{BIB}\text{T}_{\text{E}}\text{X}$ looks for the “von” part only at brace level 0—that is, because names entirely in brace level 1 (and thus being indeterminate) default to the last rather than the “von” part:

```
author = "Canudas {de} Wit, Carlos",
```

However, to prevent the “de” from appearing in the labels of alphanumeric styles, you can use:

```
author = "Canudas{ }de Wit, Carlos",
```

or:

```
author = "{Canudas de} Wit, Carlos",
```

The same approach can be used for multiple last name “von-like” parts:

```
author = "Gomez {de} {la} Serna, Ramon",
```

However, to prevent the “de la” from appearing in the labels of alphanumeric styles, you can use:

```
author = "Gomez{ }de{ }la Serna, Ramon",
```

or join them all together in brace level 1:

```
author = "{Gomez de la} Serna, Ramon",
```

Be aware that, if you omit the braces, the mistake may not be visible when using bibstyles that place the “von” just before the last part because, when using the “von last, first” format, $\text{BIB}\text{T}_{\text{E}}\text{X}$ will consider everything up through the last “von” part to be the “von” part (in this case “Gomez de la”).

²Assuming that the bibstyle uses the “von” part when constructing the label, as is usually the case. As Markey mentioned, perhaps doing so is not the correct thing to do.

Thanks to Nicolas Markey for posting these solutions to the newsgroup comp.text.tex on February 22, 2002 in the thread “BibTeX “von” in middle name” and on March 26, 2002 in the thread “Strange Citation {BibTeX question}”. See also the November 1, 2004 comp.text.tex thread “More about bibtex formatting of compound surnames”, the November 27, 1999 comp.text.tex thread “bibTeX: first name + last name + von + last name” and “How to get lowercase letters in the Last?” in [5].

Q11: How about names that do not have a space between the “von” and last parts (e.g., Jean d’Ormesson)?

Just insert a `\relax` to gobble up the space:

```
author = "d'\relax Ormesson, Jean",
```

Thanks to Nicolas Markey for answering this in [5].

Q12: BIB_TE_X is not correctly sorting names with letters such as “Æ”, “Ø”, etc. How can I fix this?

On systems, such as MiK_TE_X, that have 8-bit BIB_TE_X [20] (the executable is called `bibtex8`) you can use the `-c` or `--csfile` option to specify a code page and sorting order file. There is a code page for Scandinavian (88591sca.csf).

Thanks to Piet van Oostrum for posting this solution to the newsgroup comp.text.tex on June 13, 2006 in the thread “BibTeX doesn’t get the alphabet right”.

Alternatively, you can provide a sort key, but it can be tricky to produce one that places names after (all) those that begin with “Z”:

```
key = "Zzzname",
```

Thanks to Sara Larson for posting this solution to the newsgroup comp.text.tex on January 26, 1998 in the thread “BibTeX”.

Q13: Should I use words or numerals for the month, edition, etc., fields, and why?

You should always use the three letter month macros (undelimited) for the name of the month (jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec) as this lets each .bst file format the month names as is required for that bibliography style. There is usually no excuse for doing things like:

```
month = "{Sept.}",
```

to compensate for .bst files that do not format their month names correctly, because such things are easy to manually fix within the .bst file. Just find the month macro definitions:

```
MACRO {sep} {"Sep."}
```

and correct them as needed.

If you should need a “multiple month,” you can still build one using the BIB_TE_X concatenation operator “#” in conjunction with the month name macros:

```
month = jun # "/" # jul,
```

However, the edition field poses a bit of a challenge. The BIB_TE_X standard way of specifying edition numbers is to use ordinal words with capital first letters such as “First”, “Second”, “Third” and so forth. The problem is that some bibliography styles use words for edition numbers while others use numerical values (e.g., “First”, “1st”, “1”, etc.). Converting between these forms is currently non-trivial because the algorithms and lookup tables required for such conversions can become quite complex. Bibstyle files made with recent versions of `makebst` use an algorithm that will convert edition numbers as needed up to about the fifth. After that, they will fallback and use the form as it is given in the edition field. But, when converting between the Arabic-numeral (e.g., “17”) and Arabic-ordinal (e.g., “17th”) forms, which is a much easier task, they are usually smart enough to handle any number.

So, with this in mind, it might be a good idea to use “First”, “Second” all the way up to “Fifth” and then use the form of the bibstyle that you use most often for editions after “Fifth”:

```
edition = "{17th}",
```

One way around this problem might be to have .bst files provide an secondary edition field so that the user can specify one or both the ordinal word and Arabic numeral forms. Then, the .bst files can choose which to work with based on the information that has been given (this pushes the tough part of the conversion into “human space” — the database author). However, such extensions are considered to be non-standard as well as awkward. Another, superior approach, would be to have an algorithm within the .bst files that is clever enough to accept either the ordinal-word or Arabic-numeral forms, or even just the Arabic-numeral form, and then to convert as needed.

Finally, numbers like the year often do not need delimiters, but it might be a good idea to use them anyway as they will be needed if the year field ever contains non-numeric characters:

```
year = "2002/2003",
```

Q14: Where am I supposed to put my .bst and .bib files if I want them to be accessible system-wide?

The standard place for these is in the `bst` and `bib` subdirectories of `<texmf>/bibtex`, where `<texmf>` is the top level directory of your \TeX /METAFONT installation. (The local tree, `<texmf-local>`, which may be called `localtexmf`, is perhaps an even better place because it will not get overwritten if the system is later upgraded.) Unix users will need to have root access to be able to add things here. You can also modify the `BSTINPUTS` and `BIBINPUTS` environment variables (Unix) or the “Input Dirs” path in the `bibtex` section of the `miktex.ini` configuration file (MiK \TeX) to alter the `BIB \TeX` search paths. Note that some/most \TeX systems require that the directory lookup hash tables be refreshed before files added to the `<texmf>` tree are recognized. (te \TeX and fp \TeX users should run “`texhash`” as root, MiK \TeX users can run “`initexmf -u`”).

Q15: When I use cross references, `BIB \TeX` inserts the cross referenced entry into the bibliography list. How do I control this feature? For instance, I want to use the cross reference facility to allow me to make “parent” entries so that I will not have to retype the same information all the time in the “child” entries that “inherent” from them (via `crossref=` fields), but I don’t want to have the “parents” appear in the bibliography and I want the child entries to be formatted the same way as they are when not using `crossref` (i.e., no `\cite` or otherwise altered formatting in them).

This is controlled by the `-min-crossrefs` option on the `BIB \TeX` command line. When a number of entries equal to or greater than `-min-crossrefs` uses a cross reference, that cross reference is given an entry of its own and the format of the entries that reference it may (depending on the .bst file) change to refer to it (via using `\cite`).³ The default value of `-min-crossrefs` is usually 2. With current versions (0.99c) of `BIB \TeX` , this cannot be altered from within the .bst files. So, if you don’t want the cross referenced entries to appear, set `-min-crossrefs` to a high value:

```
bibtex -min-crossrefs=9000 myfile
```

Regardless of the value of `-min-crossrefs`, any entry using a cross referenced entry will inherit any field values that it is missing from the cross referenced entry.

Users should also be aware that the book entry definition in most .bst files use the `series` field of a cross referenced entry to format the title of the inclusive work, but `incollection` and `inbook` uses `booktitle` field to contain the title of the inclusive work. Thanks to Matt Swift for posting these clarifications to the newsgroup `comp.text.tex` on August 14-15, 1998 in the thread “Bib \TeX perts: is this a bizarre Bib \TeX crossref bug?”.

Q16: How do I get multiple citations within a single bibliography entry as is sometimes done in

³Note to .bst file designers: entries using cross references will have `crossref missing$` test true if less than `-min-crossrefs` entries use that cross reference. This allows the .bst to alter the formats of the entries to cite the extra cross reference that will be present (when `crossref missing$` will test false).

physics journals?

See the T_EX FAQ [7, “Multiple citations” entry].

Q17: What kinds of support and bibstyles are there for languages other than English?

See the T_EX FAQ [7, “Non-english bibliographies” entry].

3.2 Citation Questions**Q18: When I put `\cite` commands in the captions of figures or tables (or in section headings) under a numerical style, the citation order gets corrupted resulting in the “real” first reference not being labeled as “1”. Why does this happen and how can I fix it?**

The problem stems from the fact that the citations get moved into the List of Figures/Tables or Table of Contents where they become numbered as the first citations. The solution is to load Donald Arseneau’s `notoccite.sty` package [21].

Q19: Why do I get errors when I put `\cite` commands in the captions of figures or tables?

You are using a version of L^AT_EX that is older than December 1995. You might even be still using L^AT_EX 2.09. Some class files can have this problem too. The best thing to do is to update your L^AT_EX or class file, but you can get around the problem by preceding the `\cite` command with `\protect`.

Q20: I am having trouble getting `cite.sty` to work with `hyperref.sty`. What can I do?

At present (2007), these two packages will not work perfectly together. Recent versions (4.0 and later) of `cite.sty` will work with `hyperref.sty` in that you can load both and still get sorted and compressed citation lists. However, they will not be hyperlinked. Future versions of `cite.sty` may fix this.

Q21: How do I reference where the citation occurred in the bibliography?

See the T_EX FAQ [7, “References from the bibliography to the citation” entry].

3.3 Internet and Other Special Reference Questions**Q22: How do I reference internet URLs, patents and/or standards?**

Some `.bst` files have provisions for these (such as `IEEEtran.bst` [22]), but standard BIB_TE_X styles do not provide them yet. The `misc` type can be used for internet references with the `note` field containing the URL. The `misc` type can also be used for patents. Nelson Beebe has an example `.bib` file showing how he handled his references of various standards (via `@manual`) [23]. See also [Tip 13](#).

Q23: Within the `url` field, how should I enter `%`, `_`, `&`, `#` and other special characters that normally need to be escaped (e.g., `_`)?

If you are not using a bibstyle that renders the URLs within a `\url` command in conjunction with a package that provides a `\url` that automatically escapes these special characters then you have to escape these characters (as is normally required in L^AT_EX) or an error will occur. However, the solution recommended by the vast majority of T_EX gurus is to load `url.sty` [14] or `hyperref.sty` (version 6.72y 2002-09-12 or later) [15] in conjunction with a bibstyle that places the URL within a `\url` command and then *not* to escape any special characters when entering the `url` field into the BIB_TE_X database. Note that it is not safe to escape all the special characters and still use `url.sty` and/or `hyperref.sty`. For example, under the `\url` command from `url.sty`,

```
url = "http://one\two",
```

will result in `http://one\%two` (complete with backslash), even though valid URLs can not contain backslashes. Similarly, under the `\url` command of `hyperref.sty`,

```
url = "http://one&two",
```

will result in `http://one&two`.

Thanks to Robin Fairbairns and Donald Arseneau for helping the authors to better understand this issue.

Q24: My URLs don't want to "break" which results in badly formatted entries accompanied by under/overfull hbox warnings. What can I do to fix this?

Properly breaking URLs is a non-trivial matter. The `url.sty` package [14] will help out a lot and it may be all you need. However, even it can sometimes benefit from a little assistance/configuration. `url.sty`'s configuration can be altered (via the `\UrlBreaks`, `\UrlBigBreaks`, etc. commands) to allow for more breakpoints. See the `url.sty` documentation (in the `url.sty` source code) for details.

This issue can also arise when using `hyperref.sty` [15] under DVI (but not PDF from `pdfLATEX`) output. One good workaround for this is to use Vilar Camara Neto's `breakurl.sty` [24] package which is designed to be loaded *after* `hyperref.sty` to provide a breakable hyperlinked `\url` command under DVI output.

Q25: I am getting unwanted % signs in my URLs. Why? How do I stop this?

`BIBTEX` works by scanning the `.aux` file of your document along with the `.bib` databases, and then writing out the bibliography `LATEX` code to the `.bbl` file for later importation by the `.tex` file during the next `LATEX` run. But, writing out `TEX` code is not an easy thing to do — especially when one does not interpret it (and therefore does not know what it means) as it is being written. `BIBTEX` has to break lines every now and then to keep the lines of the `.bbl` `LATEX` bibliography code to a reasonable length. Normally, `BIBTEX` breaks at spaces. However, URLs can have long lengths of characters with no spaces. In such cases, `BIBTEX` will break between characters. But this poses a new danger. If `BIBTEX` does not shield the end-of-line break point with a `%`, `TEX` will add an unwanted space.

So, `BIBTEX` places a `%` before the break point. For example, a `.bbl` file might contain something like:

```
\url{http://www.bibtex.org/topfiledirectory/middlefiledirector%
y/lowerfiledirectory/bottomfiledirectory}
```

Now, ordinarily, `TEX` would ignore that `%`. But, `%` is a valid URL character, and within the `\url` command argument, it will be considered as being part of your URL and show up in the bibliography.⁴

You can get around this problem by putting your URLs within `LATEX` macros (such as those obtained by using `url.sty`'s `\urldef` command). But, this is a pain for more than a few URLs.

Manually editing the `.bbl` to remove the offending `%` characters or using scripts that do it for you are also possible options.

The best and easiest approach is to use packages with `\url` commands that are programmed to ignore `%` characters that are immediately followed by a line break. Recent versions (6.70s, 2000-08-31 and later) of the `hyperref` packages as well as `url.sty` (version 1.6 and later) have a `\url` command that does this.

Future versions of `BIBTEX` may allow the `.bst` file to better control the line breaking of their `.bbl` output so as to eliminate this problem altogether.

3.4 Bibliography Modification Questions

Q26: How do I get the bibliography/references list into the table of contents?

The simple answer to this question is to insert the line

```
\addcontentsline{toc}{chapter}{\bibname}
```

⁴Those of you who are familiar with `TEX` may wonder why the `%` is not tokenized as a comment character when the `\url` command acquires its argument. The answer is that `\url` commands are usually implemented in an argumentless way that allows them to change catcodes as needed just prior to argument scanning.

if you are using book or report class, or

```
\addcontentsline{toc}{section}{\refname}
```

if you are using article class, or

```
\addcontentsline{toc}{section}{whatever-you-want}
```

immediately after the line containing `\begin{thebibliography}`. The problem here (if you are using \LaTeX) is that this line will occur in the `.bbl` file (created by \LaTeX) not the `.tex` source created by you. You can place the `\addcontentsline` line before `\begin{thebibliography}`, but you've got to watch out if the bibliography starts on a new page (causing the table of contents entry to point to the page just prior to where the bibliography actually starts). Under such circumstances, a quick hack is to add a `\clearpage` or `\cleardoublepage` just prior to the `\addcontentsline`.

The better solution requires modifying the document class file to redefine the `\thebibliography` environment so as to include an `\addcontentsline`. Another, related, approach is to place the `\addcontentsline` within the `\@openbib@code` macro ([Question 27](#)).

Last, but certainly not least, there is Peter Wilson's `tocbibind.sty` package [25] that is designed just for this purpose. However, be forewarned that `tocbibind.sty` may not work with every \LaTeX class file (it will of course work with the standard article, report, book and proc classes).

Q27: How do I alter the spacing between items in the references list?

The bibliography environment depends on the `\list` environment for formatting. With `\list`, one can control the separation between the items by altering the values of `\itemsep`, `\parsep`, etc. The problem is how to insert commands like these into the bibliography environment. One possible approach is to redefine `\thebibliography` by making a copy of the one in the class file you are using and insert the needed commands into the second argument of the `\list` within it to make a "hacked" version that will do what you want. Another approach is to take advantage of the fact that, in most modern class files (including the standard ones of \LaTeX 2_ϵ), the command `\@openbib@code` occurs at the above mentioned point to provide an external way to switch between open and closed bibliography styles.

So, you could do some commands like:

```
\makeatletter
\newcommand{\adjustmybblparameters}{\setlength{\itemsep}{2\baselineskip}\setlength{\parsep}{0pt}}
\let\ORIGINALlatex@openbib@code=\@openbib@code
\renewcommand{\@openbib@code}{\ORIGINALlatex@openbib@code\adjustmybblparameters}
\makeatother
```

where `\itemsep` and the other parameters are adjusted as you wish. This code should to be placed before the start of the bibliography. The above works by redefining `\@openbib@code` to contain code that alters list parameters in the desired way. It will not work with class files that don't use the `\@openbib@code` (and will generate an error if that command has not been defined).

If you are using the `natbib` package [11], note that it provides a `\bibsep` length that allows for easy adjustment of the bibliography list spacing.

Q28: How do I get numbers in the references list to appear without the square brackets around them?

Before the bibliography section, insert the following:

```
\makeatletter
\renewcommand\@biblabel[1]{\hfill #1.}
\makeatother
```

and then customize as needed. If this does not change the format, then the class file has “hard coded” the label definitions within its definition of `\thebibliography`.

Q29: How do I obtain separate bibliographies for each chapter? How do I get multiple bibliographies in my document?

For separate bibliographies in each chapter (or section) use Donald Arseneau’s⁵ `chapterbib.sty` [26] or Thorsten Hansen’s `bibunits.sty` [27]. For multiple bibliographies (not tied to sections) there is Thorsten Hansen’s `multibib.sty` [28], Stefan Ulrich and Pierre Basso’s `bibtopic.sty` [29], and Apostolos Syropoulos’ `multibbl.sty` [30]. For more information, see the T_EX FAQ [7, “Separate bibliographies per chapter?” and “Multiple bibliographies?” entries]. Also, [8, “If you need several indices” section] may be of interest.

Q30: How do I get bibliography entries inline with the main text?

See the T_EX FAQ [7, “Putting bibliography entries in text” entry].

Q31: How do I get multiple entries within a single bibliography entry?

See the T_EX FAQ [7, “Multiple citations” entry].

3.5 Bibstyle Design Questions

Q32: How can I learn how to make my own .bst file? What is that strange looking code in the .bst files?

Patrick Daly’s custom-bib package [16] (Tip 14) has an automated .bst file builder called “makebst”. Makebst has an amazing ability to create many, many, different bibliography styles. However, using it does involve answering about 80 questions—some of which the user might not know the proper answer to. In cases of unusual bibliography styles, which, unfortunately, there seems to be no shortage of, makebst may not be a suitable substitute for manual programming. Often a minor programming modification is all that is needed to bring a style (pre-existing or made with makebst) to acceptability—if the current results are close to what you need.

The original guide to the BIB_T_EX language (which was never named—“BIB_T_EX script” is as good a name as any) is Oren Patashnik’s “Designing BIB_T_EX Styles” [31]. BIB_T_EX script programming is much like other stack-based languages such as PostScript and Forth.

Nicolas Markey’s *Tame the BeaST — The B to X of BIB_T_EX* [5] has a section with an excellent tutorial on .bst files and the BIB_T_EX language.

It can be helpful to study an existing .bst file to see how it works. The standard BIB_T_EX style files (such as `unsrt`, `plain`) [32] (for the original source file with comments, see [33]) are perhaps easier to understand than the more complex code produced by makebst. Makebst’s automated production code can also be more difficult to follow than if it had been manually ordered by a person. However, the makebst produced styles may have features that you need, such as URL fields—examples of which are currently lacking in the standard styles. Also, the `IEEEtran.bst` source code [9] is commented and demonstrates the implementation of a number of advanced features.

Q33: I am designing/modifying a .bst file to provide an entry type for internet references. Are there any tools to help? Which name should I use for such an entry type?

You might want to have a look at Norman Gray’s `Urlbst` package [34]. `Urlbst` is Perl script that patches existing styles to support a `url` field and `webpage` entry type. It will work if the bibstyle to be modified does not deviate too far from the way the standard styles are coded.

According to Oren Patashnik, the next release of the standard styles of BIB_T_EX (V1.00) may well feature an entry type for online references. Of the possible names “internet”, “online”, and “electronic”, Oren feels

⁵Niel Kempson wrote the original version.

that “electronic” is the most likely candidate. However, some authors of potential rivals of $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$ may disagree here. In particular, Philipp Lehman’s `biblatex` [35] uses the entry name “online”.

Remember that, if a `.bst` file does not recognize an entry name, it will result in an error. So, users of your `.bst` file need to be warned that they might want to keep all their `@electronic` or `@online` entries in a separate `.bib` file to help flag the fact that they can’t be used with every `.bst` file. (Their mere presence in the `.bib` file will not cause an error, but if they are ever actually referenced under a `.bst` that does not recognize the type, there will be an error.) Furthermore, some `.bst` files already use other names such as “webpage” or “www” — so you may eventually end up having to rename database entry types and/or creating aliases in other `.bst` files. Sigh.

Q34: How can I modify a `.bst` file so that it will handle an “electronic”, “online”, etc., entry type that it currently does not understand?

The easiest solution is to create aliases with the desired names that refer to an existing type that is most compatible. The `misc` entry type is often the most appropriate. This can be done by placing commands (in the `.bst` file) such as:

```
FUNCTION {online} {misc}
FUNCTION {internet} {misc}
FUNCTION {www} {misc}
FUNCTION {webpage} {misc}
FUNCTION {electronic} {misc}
```

A good place is around the same place the `.bst` file defines:

```
FUNCTION {default.type} {misc}
```

Of course, you won’t get anything other than `misc` type formatting, but the results might be quite usable if the `.bst` file supports the `url` field (adding this latter capability is more difficult).

The same trick can be used to provide aliases for existing `electronic/online` entry types:

```
FUNCTION {online} {webpage}
```

Q35: I need to have $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$ format the abbreviated names with no space between the initials. However, when formatting like this, $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$ no longer places a hyphen between the initials of the hyphenated names. How can I overcome this and have both capabilities?

This is a known problem. Briefly, when using a `format.name$` string like:

```
{f{.}.~}{vv~}{ll}{, jj}
```

“Jean-Pierre Talpin” will be formatted as “J.P. Talpin” when “J.-P. Talpin” is desired. The only known workaround is to modify the `.bst` file so that each part of the name is formatted separately, all spaces are manually removed from the first name part, and then the parts are joined back together. The code to do this looks like:

```
STRINGS {z}
FUNCTION {remove.spaces}
{ 'z :=
  ""
  { z empty$ not }
  { z #1 #1 substring$
    z #2 global.max$ substring$ 'z :=
    duplicate$ "" =
      'pop$
      { duplicate$ " " =
        'pop$
        { * }
      }
    }
}
```



```

        if$
    }
    if$
}
while$
}
INTEGERS { nameptr namesleft numnames }
FUNCTION {forname.format.string}
{ "{f.}" }
FUNCTION {surname.format.string}
{ "{vv~}{ll}" }
FUNCTION {jnrname.format.string}
{ "{, jj}" }
FUNCTION {format.names}
{ 's :=
  #1 'nameptr :=
  s num.names$ 'numnames :=
  numnames 'namesleft :=
  { namesleft #0 > }
  { s nameptr forname.format.string format.name$ remove.spaces
    duplicate$ empty$ 'skip$ { "~" * } if$
    s nameptr surname.format.string format.name$ *
    s nameptr jnrname.format.string format.name$ *
    't :=
  .
  .

```

Obviously, this code will have to be adjusted to the specific name format desired and .bst file used. Thanks to Philip G. Ratcliffe for posting this solution to the newsgroup comp.text.tex on June 28, 2002 in the thread “BibTeX, bst hackers, format.name\$, please help”.

Q36: I want to make .bst file with a macro that contains an umlaut accent. For instance, I need a macro for March in German (März). `MACRO{mar}{“M\“arz”}` will not work because BibTeX sees the quote as a delimiter (even if shielded in braces) and I can’t use the `BibTeX quote$` command within a BibTeX macro. How do I do this?

Perhaps the best way out of this jam is to use TeX’s `^^` notation for characters. So you can do:

```
MACRO{mar}{“M\^^b{a}rz”}
```

The braces prevent TeX from seeing the “a” as a hex digit. This approach is safe—it does not depend on font or character encoding, just TeX’s fixed internal coding. Thanks to Gerd Neugebauer and Daniel Luecking for posting this answer to the newsgroup comp.text.tex on October 17, 2000 in the thread “Can a BibTeX MACRO definition include ‘”’?”.

Q37: When running BibTeX, I get an error message like:

```

Sorry-you’ve exceeded BibTeX’s single function space 100
or
Sorry-you’ve exceeded BibTeX’s wizard-defined function space 3000
or
Warning-you’ve exceeded 100, the entry-string-size ...
or
Warning-you’ve exceeded 1000, the global-string-size, for entry ...
or
Sorry-you’ve exceeded BibTeX’s buffer size 1000

```

or

Sorry-you've exceeded BibTeX's pool size 65000

How can I fix this?

Early distributions of BIB_TE_X were bad about problems like these. Later versions provide much more memory for processing. Probably the best way around this is to upgrade your version of BIB_TE_X for whatever T_EX installation you have. Alternatively, you can recompile the BIB_TE_X source after enlarging parameters such as `wiz_fn_space=3000` in the appropriate places in the source files.⁶

Here is a list of compile time parameters and their default values, meanings and/or limits. The default values shown are for the original release of BIB_TE_X. Current versions typically use larger defaults. Data obtained from page 5 of *The BIB_TE_X Preprocessor* (`btxsrsrc.pdf`) by Oren Patashnik, May 28, 2002.

`buf_size` (1000); Maximum number of characters in an input line (or string).

`min_print_line` (3); Minimum .bbl line length; must be ≥ 3 .

`max_print_line` (79); Maximum .bbl line length; must be $> \text{min_print_line}$ and $< \text{buf_size}$.

`aux_stack_size` (20); Maximum number of simultaneous open .aux files.

`max_bib_files` (20); Maximum number of .bib files allowed.

`pool_size` (65000); Maximum number of characters in strings.

`max_strings` (4000); Maximum number of strings, including pre-defined; must be $\leq \text{hash_size}$.

`max_cites` (750); Maximum number of distinct cite keys; Must be $\leq \text{max_strings}$.

`min_crossrefs` (2); Minimum number of cross-refs required for automatic citation list inclusion.

`wiz_fn_space` (3000); Maximum amount of “wiz defined” function space.

`single_fn_space` (100); Maximum amount for a single “wiz defined” function.

`max_ent_ints` (3000); Maximum number of entry integer variables (entries \times entry integer variables).

`max_ent_strs` (3000); Maximum number of entry string variables (entries \times entry string variables).

`ent_str_size` (100); Maximum size of an entry string variable; must be $\leq \text{buf_size}$.

`glob_str_size` (1000); Maximum size of a global string variable; must be $\leq \text{buf_size}$.

`max_fields` (17250); Maximum number of fields (entries \times fields, about $23 \times \text{max_cites}$ for consistency).

`lit_stk_size` (100); Maximum number of literal functions on the stack.

`hash_size` (5000); Must be $\geq \text{max_strings}$ and $\geq \text{hash_prime}$.

`hash_prime` (4253); A prime number about 85% of hash size, ≥ 128 and $< 2^{14} - 2^6$.

`file_name_size` (40); File names shouldn't be longer than this.

`max_glob_strs` (10); Maximum number of global string variable names.

Some versions of BIB_TE_X have been patched to allow for the memory limits to be changed via the contents of a “`bibtexmemory.config`” file (without having to recompile), but, unfortunately, such systems don't seem to be very common.

There is also a version of BIB_TE_X with a much larger capacity. “8-bit” BIB_TE_X was written by Niel Kempson and Alejandro Aguilar-Sierra [20]. On many systems with 8-bit BIB_TE_X (MiK_TE_X), the executable is called `bibtex8`.⁷

The first two error messages indicate that BIB_TE_X ran out of memory when running the .bst script because a function definition (first case) or all the function definitions (second case) in the .bst file consumed too much memory. It may be possible to fix these problems by altering the .bst file so that the single large

⁶If you are changing the source, it might be best to make the changes in the source change file (e.g., `bibtex.ch`) rather than the source files created using it (e.g., `bibtex.c`) and then running `make` so that the changes persist if `make` ever is run again. Most change files already increase the BIB_TE_X parameters somewhat. Note: within change files (.ch) the original code is between `@x` and `@y` while the code that replaces it is between the `@y` and `@z` — so you want change the values of the parameters that are between instances of `@y` and `@z`.

⁷However, command options may be needed to obtain the higher capacity, e.g., `bibtex8 -H myfile`. Use `bibtex8 -help` to list the possible options.

offending function (first case) is divided into two smaller functions or removing unneeded functions (second case).

The “entry-string-size” error message is caused by an entry field that is too long for `BIBTEX` to handle. One workaround is store the long text into a `LATEX` file (say `longfield.tex`) and then replace the field with `\input{./longfield.tex}` instead of the text itself. Of course, you may have to manually adjust things if `BIBTEX` is supposed to do case changes or abbreviations within it. Note that the 100 bytes indicated by the example error message above is a very low limit — most modern `BIBTEX`s (especially those on Unix systems) provide a limit much larger than this. As covered in the `TEX` FAQ [7, “String too long in BibTeX” entry], many (most) `.bst` files use a coding technique in their output routine that uses memory inefficiently. These `output.null` functions can be improved by changing the first line and deleting the last. That is, this form:

```
FUNCTION {output.nonnull}
{ 's :=
.
.
.
s
}
```

can be changed to this:

```
FUNCTION {output.nonnull}
{ swap$
.
.
.
}
```

Sometimes the increase in memory efficiency is enough to stop the problem. The “global-string-size” error is similar to the “entry-string-size” except that it applies to the length of internal string work variables within the `.bst` file.

The “buffer size” error means that an input line or field exceeded the size of the input read buffer. This can occur in older versions of `BIBTEX` when reading a `.bib` file that does not use Unix newline characters.

Finally, the “pool size” error means that `BIBTEX` ran out of total string space. This can occur with very large bibliographies.

Make sure that you do not have a mistake in your database entries (especially misplaced/omitted/extra braces, semicolons or commas) as these can sometimes trigger errors like the above.

3.6 Software Application Questions

Q38: What kinds of `BIBTEX`-related software applications (database managers, etc.) are there?

There are too many to name here, but you can find current lists of `BIBTEX` tools at Google [36] and The Open Directory Project [37].

Q39: What kinds of tools are there for converting between `BIBTEX` and other formats?

Some of the more notable ones include:

- | | |
|------------------|---|
| Bibutils | by Chris Putnam, is a set of tools for converting between <code>BIBTEX</code> and other formats using the Library of Congress’s Metadata Object Description Schema (MODS) [38] as an intermediate [39]. |
| BibTeXML | by Vidar Gundersen and Zeger Hendrikse, is an XML DTD and schema for <code>BIBTEX</code> conversion to/from XML and then from XML to other formats (HTML, DocBook, etc.) [40]. |
| bib2xhtml | by Diomidis Spinellis and David Hull, is an application that converts <code>BIBTEX</code> <code>.bib</code> or <code>LATEX</code> bibliography <code>.aux</code> files to (X)HTML [41]. |

bibtex2html	by Jean-Christophe Filliâtre and Claude Marché, is an application, written in Objective Caml, that converts $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$.bib files to HTML [42].
BibTeX-XML-HTML	by Michael Auth and Frank Richter, is a system of applications that converts $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$.bib files to HTML via XML [43].
bib2html	by Stéphane Galland, is an application that converts $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$.bib files to HTML [44].
Kiri's bib2html	by Kiri Wagstaff, is another application that converts $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$.bib files to HTML [45].

Q40: What about .bbl ($\text{BIB}_{\text{T}_{\text{E}}\text{X}}$ generated bibliography) file to HTML conversion?

Remember, .bbl files, although created by $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$, are really \LaTeX files. So these are best handled by applications such as $\text{\LaTeX}2\text{HTML}$ [46], $\text{T}_{\text{E}}\text{X}4\text{ht}$ [47] and TtH [48].

Q41: What alternatives are there to $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$?

Despite its age and shortcomings with respect to use with languages other than English, $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$'s maturity, simplicity and ease-of-use have helped it remain the primary bibliography generator of choice by most \LaTeX users. Some of the more notable alternatives include:

amsrefs	by Michael Downes and David M. Jones, is a $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$ -like implementation in \LaTeX which can be used with $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$ or to replace it [49].
biblatex	by Philipp Lehman, is a recent (and at the time of this writing still in beta development) and powerful reimplement of the bibliographic facilities provided by \LaTeX [35]. $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$ is used only to sort the bibliography and generate labels. The formatting of the bibliography is controlled by \LaTeX macros.
BibTeX++	a project from ENST Bretagne (University), is a $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$ replacement written in Java [50].
BiblioX	by Peter Flynn, is an XML-based system for formatting bibliographic citations and references using XSLT [51].
Bibulus	by Thomas M. Widman, is an XML-based $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$ replacement which is written in Perl [52].
cl-BibTeX	by Matthias Koepe, is a $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$ replacement implemented in Common Lisp [53].
MBibTeX	by Jean-Michel Hufflen, is a new implementation of $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$, originally written in Scheme, but now in C, with multilingual features and which uses a new bibstyle language that is close to XSLT [54].

That's all folks. Best wishes, mds.

Acknowledgments

The authors would like to thank Oren Patashnik for his help in answering questions about the operation of $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$ as well as taking the time to review this document and providing additional material. Robin Fairbairns' $\text{T}_{\text{E}}\text{X}$ FAQ helped with identifying which issues $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$ users most often encountered as well as their proper solutions. Finally, Donald Arseneau helped us to better understand the intricacies of URL handling in \LaTeX .

References

- [1] M. Shell and D. Hoadley. (2007, Jan.) `BIBTEX` tips and FAQ. `btxFAQ.pdf`. [Online]. Available: <http://www.ctan.org/tex-archive/biblio/bibtex/contrib/doc/>
- [2] L. A. Lamport, *L_AT_EX: A Document Preparation System*, 2nd ed. Harlow, England: Addison-Wesley, 1994.
- [3] H. Kopka and P. W. Daly, *Guide to L_AT_EX*, 4th ed. Harlow, England: Addison-Wesley, 2003.
- [4] O. Patashnik. (1988, Feb.) `BIBTEX`ing. `btxdoc.pdf`. [Online]. Available: <http://www.ctan.org/tex-archive/biblio/bibtex/contrib/doc/>
- [5] N. Markey. (2005, Oct.) Tame the beast — the b to x of `BIBTEX`. [Online]. Available: <http://tug.ctan.org/tex-archive/info/bibtex/tamethebeast/>
- [6] K.-J. Kim. (2004, Apr.) A `BIBTEX` guide via examples. `bibtex_guide.pdf`. [Online]. Available: <http://www.geocities.com/kijoo2000/>
- [7] R. Fairbairns. (2007, Jan.) The `TEX` FAQ. [Online]. Available: <http://www.tex.ac.uk/cgi-bin/texfaq2html/>
- [8] H. Kreuzmann. (2005, Aug.) How to efficiently use `BIBTEX`. [Online]. Available: http://www.itp.uni-hannover.de/~kreutzm/en/doc_bibtex.html
- [9] M. Shell. (2007, Jan.) The IEEEtran `BIBTEX` style. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/IEEEtran/bibtex/>
- [10] D. Arseneau. (2006, Sep.) The `cite.sty` package. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/cite/>
- [11] P. W. Daly. (2006, Sep.) The `natbib.sty` package. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/natbib/>
- [12] P. Williams. (1995, Apr.) The `harvard.sty` package. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/harvard/>
- [13] G. Paulley. (1992, Sep.) The `chicago.sty` package. [Online]. Available: <http://www.ctan.org/tex-archive/biblio/bibtex/contrib/chicago/>
- [14] D. Arseneau. (2005, Jun.) The `url.sty` package. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/misc/>
- [15] S. Rahtz and H. Oberdiek. (2006, Nov.) The `hyperref.sty` package. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/hyperref/>
- [16] P. W. Daly. (2005, Oct.) The `custom-bib` package. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/custom-bib/>
- [17] O. Patashnik. (2003, Jul.) `BIBTEX` yesterday, today, and tomorrow. [Online]. Available: <http://www.tug.org/tug2003/abstracts/final/Patashnik/patashnik-abstract/>
- [18] N. H. F. Beebe. (2006, Aug.) `TEX` user group bibliography archive. [Online]. Available: <http://www.math.utah.edu:8080/pub/tex/bib/index-table.html>
- [19] F. Mittelbach, M. Goossens, J. Braams, D. Carlisle, and C. Rowley, *The L_AT_EX Companion*, 2nd ed. Harlow, England: Addison-Wesley, 2004.
- [20] N. Kempson and A. Aguilar-Sierra. (1996, Aug.) 8-bit `BIBTEX`. [Online]. Available: <http://www.ctan.org/tex-archive/biblio/bibtex/8-bit/>
- [21] D. Arseneau. (2000, Jul.) The `notoccite.sty` package. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/misc/>
- [22] M. Shell. (2007, Jan.) The IEEEtran `cls` package. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/IEEEtran/>
- [23] N. H. F. Beebe. (2003, Mar.) References of standards. [Online]. Available: <http://www.math.utah.edu/pub/tex/bib/standard.html>

- [24] V. C. Neto. (2006, Aug.) The `breakurl.sty` package. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/breakurl/>
- [25] P. R. Wilson. (2004, May) The `tocbibind.sty` package. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/tocbibind/>
- [26] D. Arseneau. (2004, Mar.) The `chapterbib.sty` package. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/cite/>
- [27] T. Hansen. (2004, May) The `bibunits.sty` package. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/bibunits/>
- [28] ——. (2004, Jan.) The `multibib.sty` package. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/multibib/>
- [29] S. Ulrich and P. Basso. (2006, Sep.) The `bibtopic.sty` package. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/bibtopic/>
- [30] A. Syropoulos. (2004, Jul.) The `multibbl.sty` package. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/multibbl/>
- [31] O. Patashnik. (1988, Feb.) Designing $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$ styles. `btXHak.pdf`. [Online]. Available: <http://www.ctan.org/tex-archive/biblio/bibtex/contrib/doc/>
- [32] ——. (1988, Feb.) The $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$ standard styles. [Online]. Available: <http://www.ctan.org/tex-archive/biblio/bibtex/distrib/styles/>
- [33] ——. (1993, Nov.) The $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$ standard styles source code. `btXbst.doc`. This is a text file despite its `.doc` suffix. [Online]. Available: <http://www.ctan.org/tex-archive/biblio/bibtex/distrib/doc/>
- [34] N. Gray. (2005, Jun.) The `urlbst` package. [Online]. Available: <http://www.ctan.org/tex-archive/biblio/bibtex/contrib/urlbst/>
- [35] P. Lehman. (2006, Nov.) The `biblatex` package. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/exptl/biblatex/>
- [36] (2006, Nov.) Directory of $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$ related software. Google, Inc. [Online]. Available: <http://directory.google.com/Top/Computers/Software/Typesetting/TeX/BibTeX/>
- [37] (2006, Nov.) Directory of $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$ related software. The Open Directory Project. [Online]. Available: <http://dmoz.org/Computers/Software/Typesetting/TeX/BibTeX/>
- [38] (2004, Dec.) Metadata object description schema (MODS). United States Library of Congress. [Online]. Available: <http://www.loc.gov/standards/mods/>
- [39] C. Putnam. (2005, Oct.) `Bibutils`. [Online]. Available: <http://www.scripps.edu/~cdputnam/software/bibutils/>
- [40] V. B. Gundersen and Z. W. Hendrikse. (2005, May) `BibTeXML`. [Online]. Available: <http://bibtexml.sourceforge.net/>
- [41] D. Spinellis and D. Hull. (2006, Mar.) `bib2xhtml`. [Online]. Available: <http://www.spinellis.gr/sw/textproc/bib2xhtml/>
- [42] J.-C. Filliâtre and C. Marché. (2006, Nov.) `bibtex2html`. [Online]. Available: <http://www.lri.fr/~filliatr/bibtex2html/index.en.html>
- [43] M. Auth and F. M. Richter. (2006, Jun.) The `BibTeX-XML-HTML` bibliography project. [Online]. Available: <http://www.authopilot.com/xml/home.htm>
- [44] S. Galland. (2006, Oct.) `bib2html`. [Online]. Available: http://www.arakhne.org/rubrique.php?id_rubrique=19
- [45] K. Wagstaff. (2003, Nov.) `Kiri's bib2html`. [Online]. Available: <http://www.litech.org/~wkiri/bib2html/>
- [46] N. Drakos. (2004, Oct.) $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}2\text{HTML}$. [Online]. Available: <http://www.latex2html.org/>
- [47] E. M. Gurari. (2006, Nov.) `T_{\text{E}}\text{X}4ht`. [Online]. Available: <http://www.cse.ohio-state.edu/~gurari/TeX4ht/>

- [48] I. Hutchinson. (2006, Aug.) TtH. [Online]. Available: <http://hutchinson.belmont.ma.us/tth/>
- [49] M. Downes and D. M. Jones. (2004, Jun.) The amsrefs package. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/required/amslatex/amsrefs/>
- [50] (2005, Jul.) Bibtex++. ENST Bretagne. [Online]. Available: <https://info.enstb.org/projets/bibtex/>
- [51] P. Flynn. (2004, Apr.) Bibliox. [Online]. Available: <http://www.silmaril.ie/bibliox/>
- [52] T. M. Widman. (2004, May) Bibulus. [Online]. Available: <http://www.nongnu.org/bibulus/>
- [53] M. Koeppe. (2005, Mar.) cl-bibtex. [Online]. Available: <http://www.nongnu.org/cl-bibtex/>
- [54] J.-M. Hufflen. (2004, May) Mlbibtex. [Online]. Available: <http://lifc.univ-fcomte.fr/~hufflen/texts/mlbibtex/mlbibtex/>