BibTeXing

Oren Patashnik

February 8, 1988

1 Overview

[This document will be expanded when BibTeX version 1.00 comes out. Please report typos, omissions, inaccuracies, and especially unclear explanations to me (`patashnik@SCORE.STANFORD.EDU`). Suggestions for improvements are wanted and welcome.]

This documentation, for BibTeX version 0.99b, is meant for general BibTeX users; bibliography-style designers should read this document and then read "Designing BibTeX Styles" [?], which is meant for just them.

This document has three parts: Section 2 describes the differences between versions 0.98i and 0.99b of BibTeX and between the corresponding versions of the standard styles; Section 3 updates Appendix B.2 of the LaTeX book [?]; and Section 4 gives some general and specific tips that aren't documented elsewhere. It's assumed throughout that you're familiar with the relevant sections of the LaTeX book.

This documentation also serves as sample input to help BibTeX implementors get it running. For most documents, this one included, you produce the reference list by: running LaTeX on the document (to produce the `aux` file(s)), then running BibTeX (to produce the `bbl` file), then LaTeX twice more (first to find the information in the `bbl` file and then to get the forward references correct). In very rare circumstances you may need an extra BibTeX/LaTeX run.

BibTeX version 0.99b should be used with LaTeX version 2.09, for which the closed bibliography format is the default; to get the open format, use the optional document style `openbib` (in an open format there's a line break between major blocks of a reference-list entry; in a closed format the blocks run together).]

Note: BibTeX 0.99b is not compatible with the old style files; nor is BibTeX 0.98i compatible with the new ones (the new BibTeX, however, is compatible with old database files).

Note for implementors: BibTeX provides logical-area names `TEXINPUTS:` for bibliography-style files and `TEXBIB:` for database files it can't otherwise find.

2 Changes

This section describes the differences between BibTeX versions 0.98i and 0.99b, and also between the corresponding standard styles. There were a lot of differences; there will be a lot fewer between 0.99 and 1.00.

2.1 New BibTeX features

The following list explains BibTeX's new features and how to use them.

1. With the single command '`\nocite{*}`' you can now include in the reference list every entry in the database files, without having to explicitly `\cite` or `\nocite` each entry. Giving this command, in essence, `\nocite`s all the enties in the database, in database order, at the very spot in your document where you give the command.

2. You can now have as a field value (or an `@STRING` definition) the concatenation of several strings. For example if you've defined

   ```
   @STRING( WGA = " World Gnus Almanac" )
   ```

   then it's easy to produce nearly-identical `title` fields for different entries:

   ```
   @BOOK(almanac-66,
     title = 1966 # WGA,
     .  .  .
   @BOOK(almanac-67,
     title = 1967 # WGA,
   ```

   and so on. Or, you could have a field like

   ```
   month = "1~" # jan,
   ```

   which would come out something like '`1~January`' or '`1~Jan.`' in the `bbl` file, depending on how your bibliography style defines the `jan` abbreviation. You may concatenate as many strings as you like (except that there's a limit to the overall length of the resulting field); just be sure to put the concatenation character '`#`', surrounded by optional spaces or newlines, between each successive pair of strings.

3. BibTeX has a new cross-referencing feature, explained by an example. Suppose you say `\cite{no-gnats}` in your document, and suppose you have these two entries in your database file:

```
@INPROCEEDINGS(no-gnats,
  crossref = "gg-proceedings",
  author = "Rocky Gneisser",
  title = "No Gnats Are Taken for Granite",
  pages = "133-139")
. . .
@PROCEEDINGS(gg-proceedings,
  editor = "Gerald Ford and Jimmy Carter",
  title = "The Gnats and Gnus 1988 Proceedings",
  booktitle = "The Gnats and Gnus 1988 Proceedings")
```

Two things happen. First, the special `crossref` field tells BibTeX that the `no-gnats` entry should inherit any fields it's missing from the entry it cross references, `gg-proceedings`. In this case it in inherits the two fields `editor` and `booktitle`. Note that, in the standard styles at least, the `booktitle` field is irrelevant for the `PROCEEDINGS` entry type. The `booktitle` field appears here in the `gg-proceedings` entry only so that the entries that cross reference it may inherit the field. No matter how many papers from this meeting exist in the database, this `booktitle` field need only appear once.

The second thing that happens: BibTeX automatically puts the entry `gg-proceedings` into the reference list if it's cross referenced by two or more entries that you `\cite` or `\nocite`, even if you don't `\cite` or `\nocite` the `gg-proceedings` entry itself. So `gg-proceedings` will automatically appear on the reference list if one other entry besides `no-gnats` cross references it.

To guarantee that this scheme works, however, a cross-referenced entry must occur later in the database files than every entry that cross-references it. Thus, putting all cross-referenced entries at the end makes sense. (Moreover, you may not reliably nest cross references; that is, a cross-referenced entry may not itself reliably cross reference an entry. This is almost certainly not something you'd want to do, though.)

One final note: This cross-referencing feature is completely unrelated to the old BibTeX's cross referencing, which is still allowed. Thus, having a field like

```
note = "Jones \cite{jones-proof} improves the result"
```

is not affected by the new feature.

4. BibTeX now handles accented characters. For example if you have an entry with the two fields

```
author = "Kurt G{\"o}del",
year = 1931,
```

and if you're using the **alpha** bibliography style, then BibTeX will construct the label [Göd31] for this entry, which is what you'd want. To get this feature to work you must place the entire accented character in braces; in this case either `{\"o}` or `{\"{o}}` will do. Furthermore these braces must not themselves be enclosed in braces (other than the ones that might delimit the entire field or the entire entry); and there must be a backslash as the very first character inside the braces. Thus neither `{G{\"{o}}del}` nor `{G\"{o}del}` will work for this example.

This feature handles all the accented characters and all but the nonbackslashed foreign symbols found in Tables 3.1 and 3.2 of the LaTeX book. This feature behaves similarly for "accents" you might define; we'll see an example shortly. For the purposes of counting letters in labels, BibTeX considers everything contained inside the braces as a single letter.

5. BibTeX also handles hyphenated names. For example if you have an entry with

   ```
   author = "Jean-Paul Sartre",
   ```

   and if you're using the `abbrv` style, then the result is 'J.-P. Sartre'.

6. There's now an `@PREAMBLE` command for the database files. This command's syntax is just like `@STRING`'s, except that there is no name or equals-sign, just the string. Here's an example:

   ```
   @PREAMBLE{ "\newcommand{\noopsort}[1]{} "
            # "\newcommand{\singleletter}[1]{#1} " }
   ```

   (note the use of concatenation here, too). The standard styles output whatever information you give this command (LaTeX macros most likely) directly to the `bbl` file. We'll look at one possible use of this command, based on the `\noopsort` command just defined.

   The issue here is sorting (alphabetizing). BibTeX does a pretty good job, but occasionally weird circumstances conspire to confuse BibTeX: Suppose that you have entries in your database for the two books in a two-volume set by the same author, and that you'd like volume 1 to appear just before volume 2 in your reference list. Further suppose that there's now a second edition of volume 1, which came out in 1973, say, but that there's still just one edition of volume 2, which came out in 1971. Since the `plain` standard style sorts by author and then year, it will place volume 2 first (because its edition came out two years earlier) unless you help BibTeX. You can do this by using the `year` fields below for the two volumes:

```
year = "{\noopsort{a}}1973"
  .  .  .
year = "{\noopsort{b}}1971"
```

According to the definition of `\noopsort`, LaTeX will print nothing but the true year for these fields. But BibTeX will be perfectly happy pretending that `\noopsort` specifies some fancy accent that's supposed to adorn the 'a' and the 'b'; thus when BibTeX sorts it will pretend that 'a1973' and 'b1971' are the real years, and since 'a' comes before 'b', it will place volume 1 before volume 2, just what you wanted. By the way, if this author has any other works included in your database, you'd probably want to use instead something like `{\noopsort{1968a}}1973` and `{\noopsort{1968b}}1971`, so that these two books would come out in a reasonable spot relative to the author's other works (this assumes that 1968 results in a reasonable spot, say because that's when the first edition of volume 1 appeared).

There is a limit to the number of `@PREAMBLE` commands you may use, but you'll never exceed this limit if you restrict yourself to one per database file; this is not a serious restriction, given the concatenation feature (item 2).

7. BibTeX's sorting algorithm is now stable. This means that if two entries have identical sort keys, those two entries will appear in citation order. (The bibliography styles construct these sort keys—usually the author information followed by the year and the title.)

8. BibTeX no longer does case conversion for file names; this will make BibTeX easier to install on Unix systems, for example.

9. It's now easier to add code for processing a command-line `aux`-file name.

2.2 Changes to the standard styles

This section describes changes to the standard styles (`plain`, `unsrt`, `alpha`, `abbrv`) that affect ordinary users. Changes that affect style designers appear in the document "Designing BibTeX Styles" [?].

1. In general, sorting is now by "author", then year, then title—the old versions didn't use the year field. (The `alpha` style, however, sorts first by label, then "author", year, and title.) The quotes around author mean that some entry types might use something besides the author, like the editor or organization.

2. Many unnecessary ties (`~`) have been removed. LaTeX thus will produce slightly fewer '`Underfull \hbox`' messages when it's formatting the reference list.

3. Emphasizing (`{\em ...}`) has replaced italicizing (`{\it ...}`). This will almost never result in a difference between the old output and the new.

4. The `alpha` style now uses a superscripted '$^+$' instead of a '*' to represent names omitted in constructing the label. If you really liked it the way it was, however, or if you want to omit the character entirely, you don't have to modify the style file—you can override the '$^+$' by redefining the `\etalchar` command that the `alpha` style writes onto the `bbl` file (just preceding the `\thebibliography` environment); use LaTeX's `\renewcommand` inside a database `@PREAMBLE` command, described in the previous subsection's item 6.

5. The `abbrv` style now uses 'Mar.' and 'Sept.' for those months rather than 'March' and 'Sep.'

6. The standard styles use BibTeX's new cross-referencing feature by giving a `\cite` of the cross-referenced entry and by omitting from the cross-referencing entry (most of the) information that appears in the cross-referenced entry. These styles do this when a titled thing (the cross-referencing entry) is part of a larger titled thing (the cross-referenced entry). There are five such situations: when (1) an `INPROCEEDINGS` (or `CONFERENCE`, which is the same) cross references a `PROCEEDINGS`; when (2) a `BOOK`, (3) an `INBOOK`, or (4) an `INCOLLECTION` cross references a `BOOK` (in these cases, the cross-referencing entry is a single volume in a multi-volume work); and when (5) an `ARTICLE` cross references an `ARTICLE` (in this case, the cross-referenced entry is really a journal, but there's no `JOURNAL` entry type; this will result in warning messages about an empty `author` and `title` for the journal—you should just ignore these warnings).

7. The `MASTERSTHESIS` and `PHDTHESIS` entry types now take an optional `type` field. For example you can get the standard styles to call your reference a 'Ph.D. dissertation' instead of the default 'PhD thesis' by including a

   ```
   type = "{Ph.D.} dissertation"
   ```

   in your database entry.

8. Similarly, the `INBOOK` and `INCOLLECTION` entry types now take an optional `type` field, allowing 'section 1.2' instead of the default 'chapter 1.2'. You get this by putting

   ```
   chapter = "1.2",
   type = "Section"
   ```

   in your database entry.

9. The `BOOKLET`, `MASTERSTHESIS`, and `TECHREPORT` entry types now format their `title` fields as if they were `ARTICLE` `title`s rather than `BOOK` `title`s.
10. The `PROCEEDINGS` and `INPROCEEDINGS` entry types now use the `address` field to tell where a conference was held, rather than to give the address of the publisher or organization. If you want to include the publisher's or organization's address, put it in the `publisher` or `organization` field.
11. The `BOOK`, `INBOOK`, `INCOLLECTION`, and `PROCEEDINGS` entry types now allow either `volume` or `number` (but not both), rather than just `volume`.
12. The `INCOLLECTION` entry type now allows a `series` and an `edition` field.
13. The `INPROCEEDINGS` and `PROCEEDINGS` entry types now allow either a `volume` or `number`, and also a `series` field.
14. The `UNPUBLISHED` entry type now outputs, in one block, the `note` field followed by the date information.
15. The `MANUAL` entry type now prints out the `organization` in the first block if the `author` field is empty.
16. The `MISC` entry type now issues a warning if all the optional fields are empty (that is, if the entire entry is empty).

3 The Entries

This section is simply a corrected version of Appendix B.2 of the LaTeX book [?], © 1986, by Addison-Wesley. The basic scheme is the same, only a few details have changed.

3.1 Entry Types

When entering a reference in the database, the first thing to decide is what type of entry it is. No fixed classification scheme can be complete, but BibTeX provides enough entry types to handle almost any reference reasonably well.

References to different types of publications contain different information; a reference to a journal article might include the volume and number of the journal, which is usually not meaningful for a book. Therefore, database entries of different types have different fields. For each entry type, the fields are divided into three classes:

required Omitting the field will produce a warning message and, rarely, a badly formatted bibliography entry. If the required information is not meaningful, you are using the wrong entry type. However, if the required information is meaningful but, say, already included is some other field, simply ignore the warning.

optional The field's information will be used if present, but can be omitted without causing any formatting problems. You should include the optional field if it will help the reader.

ignored The field is ignored. BIBTEX ignores any field that is not required or optional, so you can include any fields you want in a `bib` file entry. It's a good idea to put all relevant information about a reference in its `bib` file entry—even information that may never appear in the bibliography. For example, if you want to keep an abstract of a paper in a computer file, put it in an `abstract` field in the paper's `bib` file entry. The `bib` file is likely to be as good a place as any for the abstract, and it is possible to design a bibliography style for printing selected abstracts. Note: Misspelling a field name will result in its being ignored, so watch out for typos (especially for optional fields, since BIBTEX won't warn you when those are missing).

The following are the standard entry types, along with their required and optional fields, that are used by the standard bibliography styles. The fields within each class (required or optional) are listed in order of occurrence in the output, except that a few entry types may perturb the order slightly, depending on what fields are missing. These entry types are similar to those adapted by Brian Reid from the classification scheme of van Leunen [?] for use in the *Scribe* system. The meanings of the individual fields are explained in the next section. Some nonstandard bibliography styles may ignore some optional fields in creating the reference. Remember that, when used in the `bib` file, the entry-type name is preceded by an `@` character.

article An article from a journal or magazine. Required fields: `author`, `title`, `journal`, `year`. Optional fields: `volume`, `number`, `pages`, `month`, `note`.

book A book with an explicit publisher. Required fields: `author` or `editor`, `title`, `publisher`, `year`. Optional fields: `volume` or `number`, `series`, `address`, `edition`, `month`, `note`.

booklet A work that is printed and bound, but without a named publisher or sponsoring institution. Required field: `title`. Optional fields: `author`, `howpublished`, `address`, `month`, `year`, `note`.

conference The same as `INPROCEEDINGS`, included for *Scribe* compatibility.

inbook A part of a book, which may be a chapter (or section or whatever) and/or a range of pages. Required fields: `author` or `editor`, `title`, `chapter` and/or `pages`, `publisher`, `year`. Optional fields: `volume` or `number`, `series`, `type`, `address`, `edition`, `month`, `note`.

incollection A part of a book having its own title. Required fields: `author`, `title`, `booktitle`, `publisher`, `year`. Optional fields: `editor`, `volume` or `number`, `series`, `type`, `chapter`, `pages`, `address`, `edition`, `month`, `note`.

inproceedings An article in a conference proceedings. Required fields: `author`, `title`, `booktitle`, `year`. Optional fields: `editor`, `volume` or `number`, `series`, `pages`, `address`, `month`, `organization`, `publisher`, `note`.

manual Technical documentation. Required field: `title`. Optional fields: `author`, `organization`, `address`, `edition`, `month`, `year`, `note`.

mastersthesis A Master's thesis. Required fields: `author`, `title`, `school`, `year`. Optional fields: `type`, `address`, `month`, `note`.

misc Use this type when nothing else fits. Required fields: none. Optional fields: `author`, `title`, `howpublished`, `month`, `year`, `note`.

phdthesis A PhD thesis. Required fields: `author`, `title`, `school`, `year`. Optional fields: `type`, `address`, `month`, `note`.

proceedings The proceedings of a conference. Required fields: `title`, `year`. Optional fields: `editor`, `volume` or `number`, `series`, `address`, `month`, `organization`, `publisher`, `note`.

techreport A report published by a school or other institution, usually numbered within a series. Required fields: `author`, `title`, `institution`, `year`. Optional fields: `type`, `number`, `address`, `month`, `note`.

unpublished A document having an author and title, but not formally published. Required fields: `author`, `title`, `note`. Optional fields: `month`, `year`.

In addition to the fields listed above, each entry type also has an optional `key` field, used in some styles for alphabetizing, for cross referencing, or for forming a `\bibitem` label. You should include a `key` field for any entry whose "author" information is missing; the "author" information is usually the `author` field, but for some entry types it can be the `editor` or even the `organization` field (Section 4 describes this in more detail). Do not confuse the `key` field with the key that appears in the `\cite` command and at the beginning of the database entry; this field is named "key" only for compatibility with *Scribe*.

3.2 Fields

Below is a description of all fields recognized by the standard bibliography styles. An entry can also contain other fields, which are ignored by those styles.

address Usually the address of the `publisher` or other type of institution. For major publishing houses, van Leunen recommends omitting the information entirely. For small publishers, on the other hand, you can help the reader by giving the complete address.

annote An annotation. It is not used by the standard bibliography styles, but may be used by others that produce an annotated bibliography.

author The name(s) of the author(s), in the format described in the LaTeX book.

booktitle Title of a book, part of which is being cited. See the LaTeX book for how to type titles. For book entries, use the `title` field instead.

chapter A chapter (or section or whatever) number.

crossref The database key of the entry being cross referenced.

edition The edition of a book—for example, "Second". This should be an ordinal, and should have the first letter capitalized, as shown here; the standard styles convert to lower case when necessary.

editor Name(s) of editor(s), typed as indicated in the LaTeX book. If there is also an `author` field, then the `editor` field gives the editor of the book or collection in which the reference appears.

howpublished How something strange has been published. The first word should be capitalized.

institution The sponsoring institution of a technical report.

journal A journal name. Abbreviations are provided for many journals; see the *Local Guide*.

key Used for alphabetizing, cross referencing, and creating a label when the "author" information (described in Section 4) is missing. This field should not be confused with the key that appears in the `\cite` command and at the beginning of the database entry.

month The month in which the work was published or, for an unpublished work, in which it was written. You should use the standard three-letter abbreviation, as described in Appendix B.1.3 of the LaTeX book.

note Any additional information that can help the reader. The first word should be capitalized.

number The number of a journal, magazine, technical report, or of a work in a series. An issue of a journal or magazine is usually identified by its volume and number; the organization that issues a technical report usually gives it a number; and sometimes books are given numbers in a named series.

organization The organization that sponsors a conference or that publishes a manual.

pages One or more page numbers or range of numbers, such as `42--111` or `7,41,73--97` or `43+` (the '+' in this last example indicates pages following that don't form a simple range). To make it easier to maintain *Scribe*-compatible databases, the standard styles convert a single dash (as in `7-33`) to the double dash used in TeX to denote number ranges (as in `7--33`).

publisher The publisher's name.

school The name of the school where a thesis was written.

series The name of a series or set of books. When citing an entire book, the the `title` field gives its title and an optional `series` field gives the name of a series or multi-volume set in which the book is published.

title The work's title, typed as explained in the LaTeX book.

type The type of a technical report—for example, "Research Note".

volume The volume of a journal or multivolume book.

year The year of publication or, for an unpublished work, the year it was written. Generally it should consist of four numerals, such as `1984`, although the standard styles can handle any `year` whose last four nonpunctuation characters are numerals, such as '(about 1984)'.

4 Helpful Hints

This section gives some random tips that aren't documented elsewhere, at least not in this detail. They are, roughly, in order of least esoteric to most. First, however, a brief spiel.

I understand that there's often little choice in choosing a bibliography style—journal X says you must use style Y and that's that. If you have a choice, however, I strongly recommend that you choose something like the `plain` standard style. Such a style, van Leunen [?] argues convincingly, encourages better writing than the alternatives—more concrete, more vivid.

The Chicago Manual of Style [?], on the other hand, espouse the author-date system, in which the citation might appear in the text as '(Jones, 1986)'. I argue that this system, besides cluttering up the text with information that may or may not be relevant, encourages the passive voice and vague writing. Furthermore the strongest arguments for using the author-date system—like "it's the most practical"—fall flat on their face with the advent of computer-typesetting technology. For instance the *Chicago Manual* contains, right in the middle of page 401, this anachronism: "The chief disadvantage of [a style like `plain`] is that additions or deletions cannot be made after the manuscript is typed without changing numbers in both text references and list." LaTeX, obviously, sidesteps the disadvantage.

Finally, the logical deficiencies of the author-date style are quite evident once you've written a program to implement it. For example, in a large bibliography, using the standard alphabetizing scheme, the entry for '(Aho et al., 1983b)' might be half a page later than the one for '(Aho et al., 1983a)'. Fixing this problem results in even worse ones. What a mess. (I have, unfortunately, programmed such a style, and if you're saddled with an unenlightened publisher or if you don't buy my propaganda, it's available from the Rochester style collection.)

Ok, so the spiel wasn't very brief; but it made me feel better, and now my blood pressure is back to normal. Here are the tips for using BibTeX with the standard styles (although many of them hold for nonstandard styles, too).

1. With BibTeX's style-designing language you can program general database manipulations, in addition to bibliography styles. For example it's a fairly easy task for someone familiar with the language to produce a database-key/author index of all the entries in a database. Consult the *Local Guide* to see what tools are available on your system.

2. The standard style's thirteen entry types do reasonably well at formatting most entries, but no scheme with just thirteen formats can do everything perfectly. Thus, you should feel free to be creative in how you use these entry types (but if you have to be too creative, there's a good chance you're using the wrong entry type).

3. Don't take the field names too seriously. Sometimes, for instance, you might have to include the publisher's address along with the publisher's name in the `publisher` field, rather than putting it in the `address` field. Or sometimes, difficult entries work best when you make judicious use of the `note` field.

4. Don't take the warning messages too seriously. Sometimes, for instance, the year appears in the title, as in *The 1966 World Gnus Almanac.* In this case it's best to omit the `year` field and to ignore BibTeX's warning message.

5. If you have too many names to list in an `author` or `editor` field, you can end the list with "and others"; the standard styles appropriately append an "et al."

6. In general, if you want to keep BibTeX from changing something to lower case, you enclose it in braces. You might not get the effect you want, however, if the very first character after the left brace is a backslash. The "special characters" item later in this section explains.

7. For *Scribe* compatibility, the database files allow an `@COMMENT` command; it's not really needed because BibTeX allows in the database files any comment that's not within an entry. If you want to comment out an entry, simply remove the '`@`' character preceding the entry type.

8. The standard styles have journal abbreviations that are computer-science oriented; these are in the style files primarily for the example. If you have a different set of journal abbreviations, it's sensible to put them in `@STRING` commands in their own database file and to list this database file as an argument to LaTeX's `\bibliography` command (but you should list this argument before the ones that specify real database entries).

9. It's best to use the three-letter abbreviations for the month, rather than spelling out the month yourself. This lets the bibliography style be consistent. And if you want to include information for the day of the month, the `month` field is usually the best place. For example

   ```
   month = jul # "~4,"
   ```

   will probably produce just what you want.

10. If you're using the `unsrt` style (references are listed in order of citation) along with the `\nocite{*}` feature (all entries in the database are included), the placement of the `\nocite{*}` command within your document file will determine the reference order. According to the rule given in Section 2.1: If the command is placed at the beginning of the document, the entries will be listed in exactly the order they occur in the database; if it's placed at the end, the entries that you explicitly `\cite` or `\nocite` will occur in citation order, and the remaining database entries will be in database order.

11. For theses, van Leunen recommends not giving the school's department after the name of the degree, since schools, not departments, issue degrees. If you really think that giving the department information will help the reader find the thesis, put that information in the `address` field.

12. The `MASTERSTHESIS` and `PHDTHESIS` entry types are so named for *Scribe* compatibility; `MINORTHESIS` and `MAJORTHESIS` probably would have been better names. Keep this in mind when trying to classify a non-U.S. thesis.

13. Here's yet another suggestion for what to do when an author's name appears slightly differently in two publications. Suppose, for example, two journals articles use these fields.

    ```
    author = "Donald E. Knuth"
    . . .
    author = "D. E. Knuth"
    ```

There are two possibilities. You could (1) simply leave them as is, or (2) assuming you know for sure that these authors are one and the same person, you could list both in the form that the author prefers (say, 'Donald E. Knuth'). In the first case, the entries might be alphabetized incorrectly, and in the second, the slightly altered name might foul up somebody's electronic library search. But there's a third possibility, which is the one I prefer. You could convert the second journal's field to

```
author = "D[onald] E. Knuth"
```

This avoids the pitfalls of the previous two solutions, since BibTeX alphabetizes this as if the brackets weren't there, and since the brackets clue the reader in that a full first name was missing from the original. Of course it introduces another pitfall—'D[onald] E. Knuth' looks ugly—but in this case I think the increase in accuracy outweighs the loss in aesthetics.

14. LaTeX's comment character '`%`' is not a comment character in the database files.

15. Here's a more complete description of the "author" information referred to in previous sections. For most entry types the "author" information is simply the `author` field. However: For the `BOOK` and `INBOOK` entry types it's the `author` field, but if there's no author then it's the `editor` field; for the `MANUAL` entry type it's the `author` field, but if there's no author then it's the `organization` field; and for the `PROCEEDINGS` entry type it's the `editor` field, but if there's no editor then it's the `organization` field.

16. When creating a label, the `alpha` style uses the "author" information described above, but with a slight change—for the `MANUAL` and `PROCEEDINGS` entry types, the `key` field takes precedence over the `organization` field. Here's a situation where this is useful.

```
organization = "The Association for Computing Machinery",
key = "ACM"
```

Without the `key` field, the `alpha` style would make a label from the first three letters of information in the `organization` field; `alpha` knows to strip off the '`The `', but it would still form a label like '[Ass86]', which, however intriguing, is uninformative. Including the `key` field, as above, would yield the better label '[ACM86]'.

You won't always need the `key` field to override the `organization`, though: With

```
organization = "Unilogic, Ltd.",
```

for instance, the `alpha` style would form the perfectly reasonable label '[Uni86]'.

17. Section 2.1 discusses accented characters. To BibTeX, an accented character is really a special case of a "special character", which consists of everything from a left brace at the top-most level, immediately followed by a backslash, up through the matching right brace. For example in the field

    ```
    author = "\AA{ke} {Jos{\'{e}} {\'{E}douard} G{\"o}del"
    ```

    there are just two special characters, '`{\'{E}douard}`' and '`{\"o}`' (the same would be true if the pair of double quotes delimiting the field were braces instead). In general, BibTeX will not do any processing of a TeX or LaTeX control sequence inside a special character, but it *will* process other characters. Thus a style that converts all titles to lower case would convert

    ```
    The {\TeX BOOK\NOOP} Experience
    ```

    to

    ```
    The {\TeX book\NOOP} experience
    ```

    (the '`The`' is still capitalized because it's the first word of the title).

    This special-character scheme is useful for handling accented characters, for getting BibTeX's alphabetizing to do what you want, and, since BibTeX counts an entire special character as just one letter, for stuffing extra characters inside labels. The file `XAMPL.BIB` distributed with BibTeX gives examples of all three uses.

18. This final item of the section describes BibTeX's names (which appear in the `author` or `editor` field) in slightly more detail than what appears in Appendix B of the LaTeX book. In what follows, a "name" corresponds to a person. (Recall that you separate multiple names in a single field with the word "and", surrounded by spaces, and not enclosed in braces. This item concerns itself with the structure of a single name.)

    Each name consists of four parts: First, von, Last, and Jr; each part consists of a (possibly empty) list of name-tokens. The Last part will be nonempty if any part is, so if there's just one token, it's always a Last token.

    Recall that Per Brinch Hansen's name should be typed

    ```
    "Brinch Hansen, Per"
    ```

The First part of his name has the single token "Per"; the Last part has two tokens, "Brinch" and "Hansen"; and the von and Jr parts are empty. If you had typed

```
"Per Brinch Hansen"
```

instead, BibTeX would (erroneously) think "Brinch" were a First-part token, just as "Paul" is a First-part token in "John Paul Jones", so this erroneous form would have two First tokens and one Last token.

Here's another example:

```
"Charles Louis Xavier Joseph de la Vall{\'e}e Poussin"
```

This name has four tokens in the First part, two in the von, and two in the Last. Here BibTeX knows where one part ends and the other begins because the tokens in the von part begin with lower-case letters.

In general, it's a von token if the first letter at brace-level 0 is in lower case. Since technically everything in a "special character" is at brace-level 0, you can trick BibTeX into thinking that a token is or is not a von token by prepending a dummy special character whose first letter past the TeX control sequence is in the desired case, upper or lower.

To summarize, BibTeX allows three possible forms for the name:

```
"First von Last"
"von Last, First"
"von Last, Jr, First"
```

You may almost always use the first form; you shouldn't if either there's a Jr part, or the Last part has multiple tokens but there's no von part.