Search
WWW http://www.math.utah.edu/~beebe

BibTeX Information and Tutorial

by

Nelson H. F. Beebe
University of Utah
Department of Mathematics, 110 LCB
155 S 1400 E RM 233
Salt Lake City, UT 84112-0090
USA
Email: beebe@math.utah.edu, beebe@acm.org, beebe@computer.org (Internet)
WWW URL: http://www.math.utah.edu/~beebe
Telephone: +1 801 581 5254

Last updates: **Thu Feb 7 11:50:51 2013 ... Wed Jun 11 10:38:04 2025**

Abstract

This report provides a detailed description of major parts of the BibTeX system for bibliographic markup. It supplies many examples of BibTeX fields and entries, discusses recommended practices, and warns about problems and pitfalls associated with particular fields.

It gives pointers to numerous software tools that can be of significant help in creating and maintaining bibliographic databases. Two of those tools provide an interface to SQL databases that provide powerful search facilities, and the ability to extract subsets of bibliographic data. One of those SQL databases is simple enough that any computer user can easily create, maintain, and use it.

Table of contents

- Fonts and color
- What is BibTeX?
- Why does BibTeX exist [and persist!]?
- Anatomy of a BibTeX entry
- Using BibTeX

- A sample BibTeX entry
  - BibTeX comments
  - BibTeX string concatenation
  - BibTeX and braces
  - Missing values
  - Standard BibTeX abbreviations
  - The `author` and `editor` fields
  - The `title` field
  - The `bookpages` field
  - The `booktitle` field
  - The `edition` field
  - The `journal` field
  - The `volume` and `number` fields
  - The `pages` and `pagecount` fields
  - The `day`, `month`, and `year` fields
  - The `publisher` and `address` fields
  - The `note`, `annote`, and `remark` fields
  - The `CODEN` field
  - The `DOI` field
  - The `ISBN` and `ISBN-13` fields
  - The `ISSN` field
  - The `ISSN-L` field
  - The `LCCN` field
  - The `MRclass`, `MRnumber`, and `MRreviewer` fields
  - The `onlinedate` and related fields
  - The `URL` field
  - The `ZMnumber` field
  - The `ZMreviewer` field
  - The `acknowledgement` field
  - The `ajournal` and `fjournal` fields
  - The `bibdate` field
  - The `bibsource` field
  - Other field names in our sample entry
  - The `crossref` field
- Dissertations and theses
- BibTeX and online documents
- BibTeX and patent documents
- BibTeX software tools
- BibTeX editing tools
- Other software tools
- Conclusions

Fonts and color

To aid in understanding, this report makes extensive use of displays of text in a fixed-width font on a pastel background, with these meanings:

```
... computer program input ...
```

```
... computer program output ...
```

```
... sample fragments of BibTeX entries ...
```

In descriptive prose, you will encounter ***document and product titles***, `field names`, `filenames` and `pathnames`, `highlighted text`, `program names`, and short `BibTeX entry fragments`.

Finally, important cautions, caveats, and warnings are displayed in separate brightly colored paragraphs, like this one.

What is BibTeX?

BibTeX is a highly portable software system for maintaining and using bibliographic data. Its files are easy for humans, and computers, to create and maintain, and BibTeX can automatically extract data from them and reformat that data into literature-reference-list items in any of hundreds of formats.

BibTeX was developed by Oren Patashnik at the Department of Computer Science at Stanford University during the 1980s, as part of the TeX typesetting system project of Donald Knuth and his students, and the LaTeX document-markup system created by Leslie Lamport at the nearby SRI laboratories.

Although BibTeX is most commonly used with TeX and LaTeX documents, apart from minor issues with backslashes and braces, BibTeX styles could be written to produce reference-list items for other document formatters and typesetting systems.

BibTeX and LaTeX are superficially based on the pioneering work by Brian Keith Reid described in his influential 1981 Carnegie-Mellon University (CMU) Ph.D. thesis ***Scribe: A Document Specification Language and its Compiler*** . *Scribe* later became a commercial product, but is no longer marketed. Carnegie-Mellon and Stanford have two of the world's leading computer-science departments, and there was good contact between their researchers. After he left CMU, Brian Reid worked at the SRI laboratories where Leslie Lamport worked. However, BibTeX, LaTeX, and TeX are completely independent software designs that share no code whatsoever with the Scribe system. A Scribe bibliography file is, in most cases, acceptable to BibTeX, although the reverse may not be true.

The BibTeX, LaTeX, and TeX software systems are now maintained by teams of wonderful volunteers who work under the auspices of the TeX User Group and affiliated language and national TeX interest groups in many countries.

Members of those groups also contribute to the Comprehensive TeX Archive Network (CTAN) repository of TeX-related software, which has many mirror sites around the world, including the North American master CTAN archive at the site that hosts the BibNet Project.

BibTeX, TeX, LaTeX, and their many companion tools are highly portable, and *free*. They have been implemented on supercomputers, mainframe computers, minicomputers, desktop and laptop computers, mobile (cell) phones and tablets, and even the US$25 Raspberry Pi computer.

A team of hardworking volunteers produces an annual software release on a DVD with implementations for many popular platforms. The release is called TeX Live , and media for it are sent annually to user group members as a benefit of membership in the nonprofit organizations. The release is also available over the Internet, including a mirror at this site, and a development, early access, and testing site here.

Why does BibTeX exist [and persist!]?

Anyone who has ever created a reference list of publications for a document quickly discovers that it is tedious, repetitive, and hard to get right. For journal and book publication, there are almost as many reference-list formats as there are publishers and editors, and many of them can be extremely picky about precise formatting details. Nevertheless, reference lists contain essentially the same information, and differ primarily in typographic details (switching between bold, italic, and roman fonts, and possibly changing font sizes) and in data layout (punctuation, abbreviation of names and long author lists, and ordering of author names).

BibTeX was written to allow a computer to take over the troublesome reformatting: the raw data that describe the essential information about a publication are the same for all output formats, and a style file controls how that information is converted to a particular reference-list format.

Because the raw data are the same for all outputs, it makes most sense for that data to come directly from the source that produced it, in the interests of minimizing errors of abbreviation, spelling, transcription, and typing. A few journal publishers, and the journal databases of the *American Mathematical Society (AMS)*, the *European Mathematical Society (EMS)*, the *American Insitute of Physics (AIP)*, the *American Physical Society (APS)*, the *Society for Industrial and Applied Mathematics (SIAM)*, *Project Euclid*, the *Association for Computing Machinery (ACM)*, the *Wiley Online Library* , and to a lesser extent, the *Institute for Electrical and Electronic Engineers (IEEE)*, may allow search results to be returned in BibTeX format. In most cases, some further work (described later) is needed to clean them up to acceptable standards for reuse and distribution.

A useful tool, `doi-to-bibtex`, uses a Digital Object Identifier (DOI) to lookup bibliographic data on the Web and return it in rough BibTeX form. Extensive experience with that tool shows that, while it is certainly useful as a start, its output almost always needs correction, and further checking against other sources.

When databases, library catalogs, and publishers fail to provide data in BibTeX format, it may still be possible to convert one or more of their marked-up formats to BibTeX. Otherwise, data can sometimes be extracted from Web pages in HTML or XML and converted to BibTeX form, but it takes a *lot* of additional software, and often much tedious editing, to bring that conversion to a successful conclusion. Much of the effort in producing and maintaining the bibliography archives at Utah has been devoted to development and continued maintenance of many different software tools, some of which are described later .

Once the conversion to BibTeX format is complete, there are now many software tools that can be used to further curate and validate the data, and then make them available on the Web for everyone to use freely, and without restriction. If the producers of the BibTeX entries exercise proper care, the bibliographic work for that publication need never be done again, and users of the data can have reasonable confidence in its correctness. Publishers are happier too if they don't have to expend resources to clean up author-provided reference lists.

Some publishers recognize the value of bibliographic data, and are making strong efforts to link such data in both directions: *who-cites-this* and *who-is-cited-by-this* are both important questions that have value for researchers, and can provide revenues for publishers.

Anatomy of a BibTeX entry

Although users often call a BibTeX file a *database*, that word may conjure up an image of a commercial, complicated, and expensive, system for data management and retrieval.

\relax!

A BibTeX file is just a *plain text file* that you can create with your favorite text editor on any common computing platform. If you can read this paragraph, you can read (and create) a BibTeX entry and store it in a BibTeX file. You can then share that work with your friends and colleagues, and if you are willing, and have access to a Web site, with every other Internet user on, or near, Planet Earth.

There is a brief description of the syntax of a BibTeX entry in an appendix of Leslie Lamport's book ***LaTeX: a Document Preparation System: User's Guide and Reference Manual*** . However, that description is not as precise is it needs to be, and a full account, with a proper language grammar, can be found in the journal article ***Bibliography Prettyprinting and Syntax Checking*** .

A BibTeX (and Scribe) bibliography entry takes this general form:

```
@DocumentType{citationlabel,
  field-1 = "value",
  field-2 = "value",
  ...
  field-k = abbrev,
  ...
}
```

The `DocumentType` is one of several standard values known to all BibTeX styles: `Article`, `Book`, `Booklet`, `InBook`, `InCollection`, `InProceedings`, `Manual`, `MastersThesis`, `Misc`, `PhdThesis`, `Proceedings`, `TechReport`, and `Unpublished`. Some styles also recognize `Periodical`.

Lettercase is not significant in `DocumentTypes` and field names; our capitalization follows common practice.

The assigned values can also be delimited by braces, and in Scribe only, by any matching delimiters (parentheses, braces, square brackets, or angle brackets). However, that syntactic flexibility is confusing to both humans and computer software, and for good reasons described in the full-account paper, we strongly encourage using only the quoted-value form shown here.

The quotation marks shown in the general form are the ASCII character decimal number 34 (hexadecimal 22, octal 042), typically found on computer keyboards near the RETurn key. They are widely used to delimit string constants in programming languages, markup systems, library catalogs, and Web search engines. However, left- and right-quotation marks in a variety of languages and scripts that are supplied in the Unicode character set, and now used by default in some text editors, are *not acceptable*.

Characters inside the quoted strings can be anything acceptable to the host operating system, but for use with TeX and LaTeX, and for many of the world's major human languages, except for the Cyrillic, Greek, Oriental, and Semitic language groups, plain ASCII characters may suffice, with TeX control sequences used to supply any needed accents. Because ASCII supplies the first 128 characters of the UTF-8 encoding of Unicode, ASCII files are also valid Unicode files.

Modern versions of BibTeX can handle any byte values inside the quoted strings, so such strings can also hold data for languages that do not use the Latin alphabet, such as Arabic, Berber, Devanagiri, Cherokee, Georgian, Hebrew, Indian subcontinent languages (with dozens of different scripts), Inuit, Japanese, Korean, Mandarin (and other dialects of Chinese), Mongolian, Persian, Russian, Ukrainian, Urdu, Yi, and

others. In such a case, the typesetting system that uses BibTeX's output must be capable of handing such characters, and there must also be suitable font support in that software, and in the operating system, printing software, printer hardware, and screen-display software. Some extensions of TeX and LaTeX can already do that.

Inside the entry, the field names can appear in any order, and if any field name is repeated, the ***last assignment holds***, just like in conventional programming languages.

Our example shows that an assignment can also use an abbreviation, recognized as such because it follows the equals sign, and does not begin with a quotation mark. We show shortly how to create such abbreviations, and why they are useful.

The comma between the last value and the closing brace that terminates the bibliography entry is *optional*. However, it is a good idea to always supply it, so that you do not have to remember to insert a comma if you later add another field/value pair to the end of the entry.

When BibTeX receives a request to format data from a particular entry identified by its *unique citation label*, it stores in memory all of the field/value pairs in the entry. Once all of the required data have been collected, it then processes the style-file instructions for that `DocumentType` to create the output reference-list data.

It is of *critical importance* that any field/value pair whose field is not required by the selected bibliography style file is simply ignored. It is *not an error* for a field to be unrecognized, and thus unused, by a particular style. Different styles may therefore use different subsets of data from the complete entry, and users can invent new field names as needed, and extend existing styles, or create new ones from scratch, to make use of the new field names. As an extreme example, at the home site of this report, our equipment-inventory records look like BibTeX entries, but with completely different field names; that format proved much easier to create and maintain than data in a spreadsheet.

Using BibTeX

It is easy to use BibTeX data in a LaTeX or TeX document. All that is needed is to give the name of the desired bibliography style and the basename(s) of the BibTeX file(s), and then at each place in the document where a literature citation is needed, to supply the `citationlabel` as an argument of a **`\cite`** macro, or one of its related macros in some extended citation styles.

Here is a minimal LaTeX wrapper file that can be used to typeset *all* of the entries in a BibTeX file:

```
\documentclass{article}

\bibliographystyle{abbrv}

\begin{document}

\nocite{*}

\bibliography{\jobname}

\end{document}
```

The `\documentclass` command declares the document type. The preamble (the part before the start of the document, where no typesetter output is produced) selects one of the original four standard sample styles, the **`abbrv`** bibliography style, which produces abbreviated numbered entries, sorted by lead author. The document body contains the special command **`\nocite{*}`** that requests that all entries are to be selected by

BibTeX, but that there is no mention of them at that point in the typeset output. The reference list is produced by the `\bibliography` command, whose argument is normally a comma-separated list of basenames of BibTeX files. Here, `\jobname` means the basename of the LaTeX file. Our sample is therefore a *generic wrapper* for typesetting *any* single BibTeX bibliography file; we just have to give them the same basename, with extensions `.ltx` (or `.tex`) and `.bib`. Such wrappers are provided for all of the bibliographies in the Utah archives, although some have more extensive bodies.

Suppose that we have adopted the basename `myrefs`. Then we can typeset our bibliography like this:

```
latex myrefs.ltx

bibtex myrefs

bibtex myrefs  % second pass needed only if entries cite other entries

latex myrefs.ltx
```

Each `latex` run produces several files with the same basename, but different extensions. The primary output is the typeset TeX DVI (DeVice Independent) file with extension `.dvi`. There is also an auxiliary file (`.aux`) that contains such things as the names of citation labels and bibliography files, and a log file (`.log`) that contains information about program and file versions, a progress report as pages are typeset, possible warning and error messages, and a final status report about the use of various internal resources, and the number of pages and bytes output.

There may be additional output files for the *index phrases* (`.idx`), the *list of figures* (`.lof`), the *list of tables* (`.lot`), the *table of contents* (`.toc`), and possibly others.

If you prefer *Adobe Portable Document Format (PDF)* output for the typeset pages, simply replace the command `latex` by `pdflatex`.

LaTeX and TeX know nothing about BibTeX bibliography files, and do not read them. All that those programs can do is produce information in the auxiliary file for later use by BibTeX. Thus, on the *first run*, there is no bibliography available yet to typeset.

The `bibtex` command does the job of collecting citation labels from the `.aux` file, finding all of the cited references in the specified bibliography file(s), formatting the references according to the specified style, and writing them to a reference-list file with the same basename, and extension `.bbl`. It also writes a log file with extension `.blg` with information about internal limits, and any warnings and errors caused by deficiencies in the BibTeX entries, such as missing expected field values, and duplicate citation labels.

Some bibliographies, including many of those in the Utah archives, contain entries that cite other entries through ***note*** field values. At least one additional BibTeX run is then required. One important use of such citations is recording the names of other entries that report comments, corrigenda, errata, and remarks about the document. That way, a database user need only cite the main paper, but the bibliography produced will automatically include the additional entries. Thus, if a document is later typeset after more updates to the BibTeX file(s), the bibliography may be larger than before.

When LaTeX is run a second time, it now finds the reference-list `.bbl` file, and the typeset output contains the bibliography. Such multiple processing steps are needed in any document formatter to resolve cross references.

The ***ConTeXt*** system, which provides an alternative to LaTeX for high-level document markup, supplies a powerful command script that examines output files and determines from them what additional programs must be run, and then runs them in the proper order, as many times as needed to ensure

consistency. No such tool yet exists for LaTeX and TeX users.

Our example illustrates the special, but useful, case of typesetting an entire bibliography. More typically, the document body would contain prose, such as this fragment for a report about Albert Einstein:

```
In his \emph{miracle year} of 1905, Albert Einstein wrote his
doctoral dissertation \cite{Einstein:1905:NBM}, and four
landmark papers in physics
\cite{Einstein:1905:EBK,%
      Einstein:1905:EVL,%
      Einstein:1905:MTW,%
      Einstein:1905:TKS}.
At least three of those papers could have garnered him the Nobel
Prize in Physics, and one of them, on the photoelectric effect
in metals, eventually did, but not until 1922, when he was
awarded the 1921 prize at the same time as Niels Bohr received
the 1922 prize. Einstein did not attend the ceremony in
Stockholm, because he was lecturing in Japan at that time.
```

Because of a silly design flaw in older LaTeX implementations, the `\cite` macro did not permit spaces in its arguments. To avoid a lengthy one-line list of citation labels for the four papers, the author wrote them on multiple lines, and exploited the fact that TeX's comment mechanism runs from the percent to the end of its line, and also *through all leading space on the next line*.

As you revise your document, and add new references, you must run LaTeX (or TeX) and BibTeX enough times to make the output consistent. However, once the citations have stabilized, the BibTeX runs are no longer needed.

The typeset document from our example contains text like this:

> **In his *miracle year* of 1905, Albert Einstein wrote his doctoral dissertation [2], and four landmark papers in physics [6, 5, 4, 3].**

The peculiar order in the second bracketed list can be collapsed to the range **[3–6]** simply by adding this command to the preamble of the LaTeX file:

```
\usepackage{cite}  % formerly: citesort, but cite is more powerful
```

The standard `abbrv`, `plain`, and `unsrt` bibliography styles produce numbered reference lists. The `alpha` style instead produces alphanumeric labels, so the typeset output would then contain

> **In his *miracle year* of 1905, Albert Einstein wrote his doctoral dissertation [Ein05a], and four landmark papers in physics [Ein05e, Ein05d, Ein05c, Ein05b].**

If we change to one of the more verbose citation styles, `chicago`, by adding

```
\usepackage{chicago}
```

to the preamble of the LaTeX file, then we get typeset output like this:

> **In his *miracle year* of 1905, Albert Einstein wrote his doctoral dissertation (Einstein 1905a), and four landmark papers in physics (Einstein 1905e; Einstein 1905d; Einstein 1905c; Einstein 1905b).**

The verbose citation styles offer additional macros for citing the author and year separately, and for getting short and long forms of the authors, with and without parentheses. You could then get output like these fragments, *without* naming any authors or years explicitly in your document, apart from their likely use in citation labels:

Einstein (1905) reported that ...
Bridgman, Einstein, and others showed in 1955 that ...
In prior work (Einstein 1901), he proved that ...

The citation macros needed to achieve such variations depend on the LaTeX package that accompanies the bibliography style file; consult their documentation for details.

The citation macros permit an optional leading bracket-delimited argument to provide more precise location information in a large document. Here are some examples from a TeX document:

```
The paper is reprinted in \cite[pp.~561--563]{Klein:1996:CPA}.

There is an English translation in \cite[Volume~3]{Beck:1987:CPAa}.

More details can be found in Watson's famous treatise on Bessel functions \cite[Chapter 17, \S 3.1]
{Watson:1922:TTN}.
```

If you use only `\cite{...}` commands in your document, you can be confident that they will be typeset as a suitably delimited list of tags that allow the reader to find them in the reference list at the end of your document, and you can use any bibliography style. Otherwise, with fancier citation macros, you commit yourself to a particular style. If you later change styles, then you might have to reword the text around all of the citations in your document.

Warning: Because the `\cite` command can produce a numeric list, *never* begin a sentence with that command. The citations should always follow a noun or preposition, like this: `Smith \cite{Smith:1992:ABC} showed that ...`.

Compact numbered and alphanumerically tagged citation styles are common in engineering, mathematics, medicine, and the physical sciences, where citation lists are often long. The reference lists may be collected at the end of the chapter or document, or may be typeset as footnotes throughout the document. In the humanities, verbose citation styles are normal. In the field of law, there are sometimes precise legal requirements for citations, and special styles, and additional BibTeX fields, have been defined to handle those needs.

A sample BibTeX entry

Enough of theory! Let's now examine a real BibTeX entry, this one for Albert Einstein's first paper, published when he was just 22 years old:

```
@String{j-ANN-PHYS-1900-4       = "Annalen der Physik (1900) (series 4)"}

@Article{Einstein:1901:FCG,
  author =       "Albert Einstein",
  title =        "{Folgerungen aus den Capillarit{\"a}tserscheinungen}.
                 ({German}) [{Consequences} of the capillarity
                 phenomenon]",
  journal =      j-ANN-PHYS-1900-4,
  volume =       "309",
  number =       "3",
  pages =        "513--523",
  year =         "1901",
  CODEN =        "ANPYA2",
  DOI =          "https://doi.org/10.1002/andp.19013090306",
  ISSN =         "0003-3804",
  ISSN-L =       "0003-3804",
  bibdate =      "Wed Nov 23 14:13:37 MST 2005",
  bibsource =    "http://www.math.utah.edu/pub/tex/bib/einstein.bib",
  note =         "This is Einstein's first published paper.",
```

```
  ZMnumber =     "32.0816.03",
  acknowledgement = ack-nhfb,
  Calaprice-number = "1",
  ajournal =     "Ann. Physik (1900) (ser. 4)",
  fjournal =     "Annalen der Physik (1900) (series 4)",
  language =     "German",
  Schilpp-number = "1",
  Whittaker-number = "1",
  xxvolume =     "4",
  ZMreviewer =   "Reg.-Rat Dr. Brix (Steglitz)",
}
```

In and around the entry, whitespace is not significant, and in the absence of repeated fields, the order of field names does not matter. The spacing and field order shown here is recommended, and supported by several software tools. Most of the fields here are given in the order of their usual appearance in a reference-list item.

BibTeX does not care how much whitespace separates entries. For ease of use by humans, and by other software, ***it is strongly recommended that entries be separated by blank lines***. That makes them look like paragraphs, and easier to isolate inside text editors, some of which provide easy-to-use commands for copying, deleting, and moving paragraphs. It also makes it easier to apply external software tools to single entries with text editors that permit such operations.

Most of the field names are obvious, and the **author**, **title**, **journal**, **volume**, **number**, **pages**, **year**, and **note** values would be used by most styles to format an item for the reference list.

In the subsections of this entry, we look at various BibTeX field names, describe their use through examples of real data, and sometimes, discuss problems and pitfalls associated with particular fields.

BibTeX comments

The current implementation of BibTeX does not have an official comment syntax, but its design is such that it ignores unrecognized material between entries, as long as that material contains no at-signs (@), and, if braces are present, those braces are balanced.

The rigorous grammar for BibTeX includes comments that begin with a percent, and run to end of line, just as in other TeX-related software. All of the bibliographies in the archives at Utah include extensive comment preambles, and in many cases, the entries are grouped into sections with leading comments. Thus, with care, BibTeX files can certainly have comments; you just cannot have them inside a BibTeX entry, unless they appear as field values (see the discussion of the **remark** field .)

BibTeX string concatenation

BibTeX provides an easy way to join strings in a field value with the string-concatenation operator, #. Here are some examples:

```
  acknowledgement = ack-nhfb # "\slash " # ack-wl,
  month =        oct # "\slash " # nov,

  publisher =    pub-PRINCETON # " and l'Acad{\'e}mie Royale de Belgique",
  address =      pub-PRINCETON:adr # "and Bruxelles, Belgique",
```

Sometimes, the same book is issued by different publishers, often ones separated by an ocean. BibTeX does not yet have a clean way to handle such cases properly, so string concatenation may be the most reasonable approach. Fortunately, co-publishing is rare, so you may never need to deal with the problem.

BibTeX and braces

The TeX typesetting system assigns ASCII backslash, braces, and backquote, and a few other characters, special significance. The first group were absent from traditional typewriters and computer keypunches, so TeX's author felt that they were unlikely to be used in most existing documents, outside of special areas, like mathematics, where they may carry important meaning.

BibTeX knows almost nothing about TeX markup, *except* for TeX's handling of backslash and braces. In TeX, braces delimit text that forms a single argument to a macro, or is the scope of a font change. In all cases, for TeX, LaTeX, and BibTeX, braces must always be balanced, and properly nested; failure to do so can produce error messages that report the problem, but often not the exact character position where the error was first detected. It is therefore important to handle braces carefully, and if possible, use a text editor that provides visual, and on-demand, brace matching to ensure that you get them right. The delimiter-balance-checking tool, `chkdelim`, has proven to be essential for solving such sometimes-hard problems.

BibTeX itself uses braces for delimiting the field/value assignments in a bibliography entry, and optionally, for delimiting value strings. In addition, it uses braces for its own notions of grouping (see the later discussion of bracing parts of human names), and also, for protection against downcasing. That operation occurs in some bibliography styles, notably in **`chapter`**, **`edition`**, **`number`**, **`title`**, **`note`**, and **`type`** values. Lettercase in the **`booktitle`** value is normally preserved.

No other bibliographic markup system known to this author provides a downcasing feature, even though some journals require that practice. In BibTeX values, it is easy to suppress downcasing:

- if the text begins with a backslash and a letter, and has more than one word, add ***two levels*** of bracing; otherwise
- add ***one level*** of bracing.

Here are some examples from **`einstein.bib`**:

```
  title =        "{Der starre K{\"o}rper und das
                 Relativit{\"a}tsprinzip}. ({German}) [{The} rigid body
                 and the {Principle of Relativity}]",

  title =        "Recent Publications: Reviews: {{\em Grundz{\"u}ge der
                 Einsteinschen Relativit{\"a}tstheorie}}, by {August
                 Kopff}",

  title =        "Comment on book review of {{\booktitle{Brotherhood of
                 the Bomb: The Tangled Lives and Loyalties of Robert
                 Oppenheimer, Ernest Lawrence, and Edward Teller}}, by
                 Gregg Herken [Am. J. Phys. {\bf 71}(4), 411--415
                 (2003)]}",

  title =        "{\TeX}: the Program",
```

In German, nouns are always capitalized, so all German-language strings in downcaseable field values need a single outer level of braces.

In the first title, after the braced German title, three additional word groups also need protecting braces.

In the second title, two brace levels are needed around the group that is typeset in an *emphasized* font style.

In the third title, the `\booktitle{...}` part of the value looks like a font change to BibTeX, even though it has

no knowledge of what the `\booktitle` macro does to its argument. Thus, two levels of braces are needed there. The volume number in the journal reference is set in boldface type, but is immune to downcasing because it contains only digits, so no additional bracing is needed. Had the volume number been something like **C-71**, it would appear as `{{\bf C-71}}` in the value string. In this example, the outer braces extend to the end of the value, so no additional interior bracing is needed.

In the fourth example, we have the exceptional case of something that looks like a font change, but contains only a single word. BiBTeX leaves that word unchanged in a downcasing operation, so no additional bracing is needed. Had we omitted the surrounding braces, a downcasing style would produce `\tex: the program`, and that is incorrect, because TeX macro names are *case sensitive*, and the macro `\tex` probably does not exist.

TeX and LaTeX recognize *control symbols* `\{` and `\}` for literal left and right braces, and they also support equivalent *control words*, `\lbrace` and `\rbrace`. Because BibTeX's brace-matching algorithm does not treat backslashed braces as literal characters, braces in BibTeX files must always be properly balanced. Thus, in those rare cases where you need *unbalanced* braces in BibTeX values, just represent them by their control words. For example, code the string `"$\{x]$"` as `"$\lbrace x]$"`, or better, as `"$\lbrace x \rbrack$"`, using one of the control words for left and right delimiters. That way, you can avoid spurious warnings from delimiter-matching tools, such as `chkdelim`.

One sometimes encounters bracing practices like this:

```
  title =        "An {I}rish patriot: {J}ohn {F}lynn",
```

That is awkward to type, hard to read, ensures that simple text searches for *John Flynn* fail to find the entry, and prevents TeX from recognizing the ligature *Fl*. Please avoid such uses, and brace entire words or phrases:

```
  title =        "An {Irish} patriot: {John Flynn}",
```

If you have BibTeX entries that need similar cleanup, that task can be done automatically with a Unix shell command like this:

```
  % bibclean --fix-braces < in.bib > out.bib
```

Missing values

Any attempt at collecting data from the real world, especially data created by other humans, has to face many problems, including *deficiencies*, *errors*, *inconsistencies*, and *irregularities*. Among the deficiencies is the problem of *missing data*, a topic that caused much debate when computer database systems were first designed in the 1960s and 1970s. With some thought about the problem, we can classify it by the answers to four one-word questions: *absent?*, *complete?*, *correct?*, and *exist?*. Those questions can be answered in at least four ways: *true*, *false*, *neither true nor false*, and *unknown*.

In the context of a BibTeX entry, it is important to address those possibilities. For example, BibTeX recognizes the special word `others` in `author` and `editor` values to indicate that there are additional names that ought to be there, but are not, usually because someone was lazy. Most BibTeX styles format that special word as the Latin abbreviation ***et al.*** [*et alii*] that means the same thing.

If BibTeX sees an empty value string for a particular field, it treats it as it would if the field were omitted entirely. The bibliography style processing then might produce a warning message, but no output. The convention adopted in the bibliography archives is that if a particular field/value pair *should* be present,

but the value string is entirely unknown to the bibliographer, then the string should contain two or more consecutive question marks. Otherwise, if there are partial data, but some may be uncertain, use the traditional mathematical symbol for an unknown, *x*. For the special case of a missing author or editor, use `Anonymous`. A standard BibTeX `note` field provides an opportunity for further explanation, and that field's value is sent to the output reference list. Here are some examples:

```
  author =       "Anonymous",

  number =       "??",

  month =        "????",

  pages =        "73--??",

  year =         "197x",

  year =         "1914 (??)",

  note =         "The document date is uncertain. It is believed
                  from the author's diaries that are preserved in
                  his collected papers at the Niels Bohr Institute
                  in Copenhagen, Denmark, to be some time between
                  July 1914 and November 1917.",
```

Some newer bibliography styles recognize the case of a value string containing only two or more question marks, and suppress the output of that field. However, the uncertainty is still recorded in the BibTeX entry, in the hope that a correct value can supplied at some future time. Older styles that lack that feature just output the question marks, and their appearance in a reference list should be a visible flag to the author **and** journal editors **and** production staff that something is seriously wrong, and in need of repair before publication can be approved. It is the author(s)' responsibility to ensure that every literature citation is complete; it is not a job for the human who later reads the document.

Standard BibTeX abbreviations

BibTeX has several standard built-in abbreviations, one set for selected computer-science journals, and the other for months. The built-in journal abbreviations should be avoided entirely in favor of clear `@String{...}` definitions. The month abbreviations are the three initial letters of their English names, in lowercase: `jan`, `feb`, ..., `nov`, `dec`. Those abbreviations are strongly recommended, because it is then possible to redefine them for reference lists in languages other than English.

The `author` and `editor` fields

These fields supply the names of the author(s) or editor(s) of the publication. Except for the *in-something* `DocumentTypes` discussed later, only one of those two fields should be supplied in a single BibTeX entry.

Because commas are sometimes used in personal names, BibTeX requires names of people to be separated by the word `and` in `author` and `editor` fields, as in these examples:

```
  author =       "P. W. Bridgman and Albert Einstein and L. Infeld and
                 H. J. Muller and C. F. Powell and J. Rotblat and
                 Bertrand Russell and Hideki Yukawa",

  author =       "Albert Einstein and Bertrand Russell",
```

Warning: Forgetting to convert commas or semicolons to `and` in lists of names that are copied from reference lists is a common human error that BibTeX may not be able to diagnose.

If only the lead authors or editors are known (a common problem in library catalogs), then the omission of trailing names must be indicated by giving a final special name of `others`.

```
  author =       "Albert Einstein and Michel Biezunski and others",
```

However, a database should be as complete as possible, and concerted efforts should always be made to find the missing names. If that were not done, than any attempt at collecting the complete works of a particular author would necessarily fail.

In some fields, such as large physics research projects, and large industrial product development, documents are issued with dozens to hundreds of authors. The archives contain some extreme examples like these two from Einstein's bibliography:

```
  author =       "B. Abbott and {373 others}",

  author =       "A. A. Abdo and M. Ackermann and M. Ajello and K. Asano
                 and W. B. Atwood and M. Axelsson and L. Baldini and
                 ....
                 C. Wilson-Hodge and B. L. Winer and K. S. Wood and
                 X. F. Wu and R. Yamazaki and T. Ylinen and M. Ziegler",
```

The second, when not elided as we did here, contains 208 authors, the current record for the most listed authors in all of the archives!

BibTeX understands that some human names are complex, and it has default rules that allow it to distinguish personal (or given) names from family names without requiring additional markup. Thus, these names are all handled as expected:

```
  author =       "Albert Einstein",

  author =       "Werner von Braun",

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

The lowercase intermediate names are BibTeX's clue that the family name begins there. However, in many other cases, BibTeX needs help from the bibliographer. Here are some examples where bracing supplies that help:

```
  author =       "J. M. {Van Kats} and H. A. van der Vorst",

  author =       "Jan {Van den Bussche} and Dirk {Van Gucht}",

  author =       "A. Michels and A. Bijl and J. {De Boer}",

  author =       "H. {De la Cruz}",
```

In the first value, the family names are *Van Kats* and *van der Vorst*; the braces in the first of them group the two words together as a single family name. You cannot assume that a *Van* (in Dutch, meaning *of*) in a name is part of the family name; it can also be a given name (e.g., *Van Johnson* and *H. Van Dyke Parunak*), and is common in Vietnamese names (e.g., *Le Van Minh* and *Van Bang Le*). In Dutch, sometimes members of the same family differ in the capitalization of *Van* in their names.

For some people, the spaces and initial capitalization may disappear from multipart family names of the form *Van den X* and *Van der X*. Such names therefore require additional care by the bibliographer, because they are sometimes transcribed incorrectly in published reference lists and journal databases. Here are some examples:

```
  author =       "C. {Vanden Eynden}",

  author =       "Peter Vandendriessche",
```

```
author =       "George VandenBerghe",

author =       "Bernard Vandermeersch",

author =       "M. vandenBrand",

author =       "Steven P. VanderWiel",
```

The Hungarian mathematical physicist Cornelius Lanczos, as he was known during his life in America, published in Hungarian, German, French, and English, and under *six different forms* of his name. Hungarian is unusual in Europe in placing the family name first. Here is how it appears in a BibTeX value for a Hungarian-language publication:

```
author =       "L{\'a}nczos{ }Korn{\'e}l",
```

The braced space ensures that BibTeX will treat his Hungarian name as a single unit, and alphabetize it under *Lánczos*. However, BibTeX cannot reduce his given name to the initial *K*, though its final version ought to be able to do so.

Several oriental cultures, including Chinese, Japanese, and Korean, also put the family name first. Here is how some former national leaders might appear in BibTeX values:

```
author =       "Chiang{ }Kai-shek",

author =       "Mao{ }Tse-Tung",

author =       "Sun{ }Yat-sen",

author =       "Noda{ }Yoshihiko",

author =       "Rhee{ }Syng-man",
```

However, it is common for most oriental names to be put in western order when they are authors of documents in western languages:

```
author =       "Yoshihiko Noda",

author =       "Syng-man Rhee",
```

Recommendation: Particularly in Chinese languages, personal (given) names are often multiple words, and individuals vary in their use of capitalization and hyphenation of those words in the Latin alphabet. Journals are sometimes inconsistent in their presentation of such names, so bibliographers must take special care in creating BibTeX author/editor data to ensure that the family name is correctly identified, because it is needed for sorting reference-list items in styles that do so.

Spanish names pose yet another difficulty for BibTeX, because the name order is usually *given name(s)*, *father's family name*, and *mother's family name*, with the latter two names sometimes separated by the conjunction *y* (*and*). Here is an example that shows several variations:

```
author =       "Gonz{\'a}lo {Ares de Parga} and Oscar {Chavoya A.} and
               Jos{\'e} L. {L{\'o}pez Bonilla}",
```

The first author has three words in his family name. The second author has abbreviated his mother's family name to an initial, and the third author has used both parents' names. In some publications, the second author might abbreviate his name even further to *O. Chavoya*, dropping the maternal family name entirely.

Our next examples show how junior-like name suffixes can be handled:

```
author =       "Hyoung M. Kim and Roy R. {Craig, Jr.}",

author =       "J. B. {White III} and J. J. Dongarra",

author =       "Edward T. {Foote, S.J.}",
```

BibTeX's documentation says that they should be entered with the family name first, such as *Craig, Roy, Jr.*, but that is awkward and unusual, and should be considered a defect of BibTeX's design that will be remedied in its final version. The archives at Utah use the conventional name order, with braces around the part that is the family name. That means that if a bibliography style outputs the family name first, it will appear as *Craig, Jr., Roy R.*, whereas some journal editors would prefer to have *Craig, Roy R., Jr.*. This author believes firmly that defective software should be fixed, rather than forcing users to contort their input data to get `correct' output data. Few people read reference lists in detail, even fewer would notice the difference between the two positions of the *Jr.* suffix, and almost no one would even care which of the two were used. No one would be confused by either of the two choices.

Reference lists, library catalogs, and publishers often abbreviate author/editor personal names to initials. That regrettable practice causes data loss, and sometimes leads to ambiguities: does *D. Knuth* mean *Donald Knuth* or *Dennis Knuth*? Both are computer scientists. For that reason, librarians often try to help catalog users by following initials used in the publication with a parenthesized given name, and that practice has been followed in the bibliography archives. Here are some examples:

```
author =       "Albert Einstein and H. (Hermann) Minkowski",

author =       "Albert Einstein and Edwin P. (Edwin Plimpton) Adams",

author =       "H. A. (Hendrik Antoon) Lorentz and Albert Einstein and
               H. (Hermann) Minkowski and Hermann Weyl and Arnold
               Sommerfeld",

author =       "Albert Einstein and S. W. (Stephen W.) Hawking",

author =       "V. A. (Vladimir Aleksandrovich) Fok",

author =       "J. P. (Jong-Ping) Hsu",

author =       "Viscount R. B. (Richard Burdon) Haldane",
```

The last author value shows another feature that BibTeX cannot yet handle properly: honorary or noble titles, military ranks, and academic degrees. Although academic ranks and titles (*Dean*, *Doctor*, *Docent*, *Professor*, ...) should normally be omitted from human names in author/editor lists (because otherwise almost all such names would be thus qualified in a given publication area), that is *not* true of other ranks and titles. Here are some examples:

```
author =       "President George W. Bush",

author =       "Prime Minister Winston Churchill",

author =       "Senator Albert Gore",

author =       "Sir John Frederick William Herschel",

author =       "General Leslie Groves",

author =       "Academician Andrei Sakharov",

author =       "Marie Fran{\c{c}}oise Biarnais and Sir Isaac Newton",

author =       "Professor Sir Ernest {Kennaway, F.R.S.}",

author =       "Sir Gavin {de Beer, D.Sc., Sc.D., D.-{\`e}s-L., F.R.S., F.S.A.}",

author =       "William {Thomson (Lord Kelvin)}",
```

```
  author =       "William {Thomson (first Baron Kelvin)}",

  author =       "King {Edward VIII}",

  author =       "Pope {Benedict XVI}",

  author =       "Sir Paul McCartney and Sir Mick Jagger",

  author =       "Lord Bertrand Russell",
```

Academic degree lists were once quite common in author credits, and the ***Annals of Science*** journal did so from its first volume in 1936, until the practice was discontinued at the end of 1973. Our bracing ensures that they are treated as junior-like suffixes, and handled reasonably well.

The personal titles, however, pose a problem, because the current BibTeX design has no way, other than modification of style files, of preventing the titles from being reduced to initials when the style calls for name shortening. It is *completely incorrect* to reduce our examples to *P. G. W. Bush*, *P. M. W. Churchill*, and so on. It is also incorrect to abbreviate the personal name for some titles, because the polite familiar address for the last two examples would be *Sir Paul*, *Sir Mick*, and *Lord Russell*. Bracing single words in a name does not suppress abbreviations. If the entire title and author's name are braced together, abbreviation to initials is suppressed, but `{Lord Bertrand Russell}` would then be sorted incorrectly under *Lord* instead of *Russell* in styles that order entries by the first author's family name.

Thus, until BibTeX reaches its final version, all that we can recommend for such cases is corrective manual tweaks of the formatted reference list, as discussed later.

The `title` field

The **`title`** field in our sample entry for Albert Einstein's first paper looks like this:

```
  title =        "{Folgerungen aus den Capillarit{\"a}tserscheinungen}.
                 ({German}) [{Consequences} of the capillarity
                 phenomenon]",
```

It illustrates a practice recommended by the *American Mathematical Society* for titles in languages other than English: the value follows the original title with a period (dot, or full stop), a parenthesized brace-protected foreign-language name, and a bracketed English translation of the title. In addition, the language of publication, when it is not English, is also recorded in the **`language`** field value, because that can be useful later for classification and searching.

Another typographic practice recommended by the AMS and by SIAM is to use an *en-dash* (two dashes in TeX coding) to separate names of people in an adjective phrase:

```
  title =        "Does the Mesotron Obey {Bose--Einstein} or
                 {Fermi--Dirac} Statistics?",

  title =        "A {Runge--Kutta} for all seasons",

  title =        "{Navier--Stokes} Equations",

  title =        "The {Riemann--Hilbert} Problem",

  title =        "{Der Beweis des Hilbert--Schmidt-Theorems}. ({German})
                 [{The} proof of the {Hilbert--Schmidt} theorem]",

  title =        "Efficient spectral-{Petrov--Galerkin} methods",

  title =        "Comments on the {Smith-Danielson--Brown} algorithm",
```

In the last example, *Smith-Danielson* is the compound family name of one author, and *Brown* is the other

author. The differing dash lengths provide a subtle clue that two, rather than three, people are credited with the algorithm. There are similar distinctions in the fifth and sixth titles.

Because the title always starts a new clause or sentence in the formatted reference list, it is never necessary to brace its first word. Nevertheless, we do so when that word is a proper noun that would require such protection if it appeared later in the title. Here are some examples:

```
  title =        "{Brownian} movement and molecular reality",

  title =        "{Relativity} and the electron theory",

  title =        "{Einstein}'s Law of Gravitation",

  title =        "{Euclid}, {Newton}, and {Einstein}",

  title =        "{Einstein} for dummies",

  title =        "{Einstein} Versus the {Physical Review}",
```

Capitalization must *not* be applied to mathematical text that begins a title, because mathematics is always case sensitive. The next prose word therefore requires bracing:

```
  title =        "$p$-{Adic} Congruences Between Binomial Coefficients",

  title =        "$\delta$-{Continuous} selections of small multifunctions",

  title =        "$\pi$-{Calculus} with noisy channels",
```

Because proper nouns are common in the titles of technical documents, much of the bibliographer's work in producing entries for new journal articles is identifying such words, and bracing them. That work is tedious and error prone, so as the archives have grown, it has made sense to mine them for braced proper nouns, and write a new software tool that can automatically brace those nouns in newly created BibTeX entries. In most cases, there is no ambiguity in doing so: *Bose--Einstein* and *Moore--Penrose* always require protecting braces. However, sometimes it takes human intelligence, and even examination of the use of a candidate word, to know whether braces are needed. Examples are color names that can also be family names (*Black*, *Brown*, *Gray*, *Green*, *White*, ...) and common words that have been trademarked and thus might be proper nouns (*Ball*, *Ford*, *Iris*, *Oracle*, *Shell*, *Wall*, *Windows*, ...).

For large new bibliographies, you can use a small shell script to extract lowercase words from the output reference list for visual inspection, in the hope of spotting downcased words that should have been brace protected. The inspection is a mind-numbing and tedious exercise, but this bibliographer has seen significant numbers of incorrectly downcased entries in published reference lists, so such checks should be routinely practiced by *authors*, *bibliographers*, and *journal production staff*. Here are the commands that the script conceals:

```
tr -cs A-Za-z_0-9- '
'    < einstein.bbl |
    grep '^[a-z]'   |
        sort -u     > einstein.tmp
```

The first pipeline stage translates to newlines all characters that are *not* letters, underscore, digits, or hyphen. The second stage removes all lines that do not start with a lowercase letter. The third stage sorts the input lines into a list of unique words that are sent to the `.tmp` file for examination by the bibliographer. If this is done routinely, it is easy to accumulate a list of words that are known never to require bracing, and add a fourth stage with the `comm` utility to report only the words that are *not* in that word list. That removes most of the false positives from the output report, reducing the human time required to examine it.

The *American Mathematical Society* journal database downcases title words (apart from the first) that are not proper nouns, so that all capitalized words that remain are therefore proper nouns that need braces in BibTeX entries. While that appears to solve the bracing problem, in this bibliographer's view, it is an abhorrent practice. It destroys information that cannot be recovered automatically, because downcasing is a noninvertible two-to-one mapping. Incorrect lettercasing is a common problem in publisher Web sites, and sometimes considerable effort is needed by the bibliographer to correct such errors.

The `bookpages` field

Current BibTeX styles make no provision for recording page counts for the containing volume of *in-something* entries, but they do support both `booktitle` and `title` fields. The Utah bibliography archives make extensive use of an additional `bookpages` field, which future style files may recognize. No changes to BibTeX itself are required to support a new field name.

For books, conference proceedings, manuals, and theses, the practice in the Utah archives for `bookpages` and `pages` fields is that roman numerals count pages in the front matter, and arabic numerals count pages in the body. Here are some examples:

```
  pages =        "471",

  pages =        "xii + 405 + 16",

  pages =        "xii + 405 + 16 color plates",

  pages =        "xix + 598 (vol. 1), vii + 540 (vol. 2)",

  pages =        "635 (est.)",
```

The first example is for a book that lacks front matter, or else uses arabic numerals for all pages. The second and third show how pages of additional material, such as photographic plates, are recorded. The fourth example is from a single entry for a two-volume work. The last indicates an estimated size; that situation is common for new books, when publishers announce books in Web pages and advertisements, and supply preliminary cataloging information to national libraries. Eventually, correct counts need to be supplied.

The `booktitle` field

Our sample BibTeX entry is for a journal article. However, had that article instead been published as a chapter in an edited book, it is necessary to distinguish the two titles. Thus, in `@Book{...}` entries with an `editor` field instead of an `author` field, and in `@Proceedings{...}` entries, which should always have an `editor` field, it is proper to duplicate the publication title in both fields:

```
@Book{Janssen:2014:CCE,
  editor =       "M. Janssen and C. Lehner",
  booktitle =    "The {Cambridge} Companion to {Einstein}",
  title =        "The {Cambridge} Companion to {Einstein}",
  publisher =    pub-CAMBRIDGE,
  address =      pub-CAMBRIDGE:adr,
...
}
```

Unlike the `title` field, which is downcased by many bibliography styles, lettercase is preserved in the `booktitle`, and thus, brace protection should not be necessary. However, doing so is recommended, because it makes the values identical, and guards against some future style that might apply downcasing to both.

The `edition` field

Books and manuals sometimes are issued in new editions that may be quite similar to previous ones, or radically different. The **`edition`** field, if present, should normally contain only an ordinal number as an English word, because most styles output that value followed by the word `edition`, or the abbreviations `ed.` or `edn.`. Here are examples:

```
  edition =      "Second",

  edition =      "Third",

  edition =      "15th",
```

Edition values are sometimes nonnumeric, or are phrases, such as these examples:

```
  edition =      "Centennial",

  edition =      "Expanded {Princeton} Science Library",

  edition =      "Second revised and enlarged",

  edition =      "Updated and expanded tenth anniversary",
```

All of those examples deviate from our recommendation, and would have to be rewritten if the reference list were output in a language other than English. By contrast, if ordinal number words are used, then a style file could easily be extended to map them into equivalent phrases in the target language. The variety of possibilities for the **`edition`** field perhaps explains why BibTeX's author did not choose to provide built-in abbreviations for the commonest values.

The `journal` field

The **`journal`** field in our sample entry looks like this:

```
  journal =      j-ANN-PHYS-1900-4,
```

It uses an abbreviation in the form adopted for the bibliography archives at Utah: a prefix `j-` signifying a journal, followed by an uppercase hyphen-separated string derived from the conventional abbreviated form of the entry (shown in the value of the **`ajournal`** field). Using a standard string abbreviation for journal and publisher names ensures output consistency, makes it easy to identify papers from the same journal, and allows the definition of the abbreviation to be changed in one place, instead of in every entry that names that journal.

Traditional practice in reference lists has been to abbreviate journal names, because that made it easier for authors, typists, and typesetters, and possibly saved a bit of space on the pages holding the reference lists. However, abbreviations introduce problems for humans and computers, because abbreviations are frequently ambiguous: is a journal cited as ***J. Chem. Phys.*** the English-language ***Journal of Chemical Physics***, or the different ***Journal of Chemistry and Physics***, or the French-language ***Journal de chimie et physique***, or the German-language ***Jahrbuch der Chemische Physik*** or ***Jahresbericht der chemische Physikverein***, or the medical ***Journal of Chemical Physiology***?

For that reason, almost all string definitions in the cited bibliography archives use the *full* journal name. A user who wants a shorter form can easily insert a following alternate definition that overrides the longer form for formatting purposes, without losing the important record of the full journal name, because string definitions work like field name assignments when the left-hand side is repeated — the *last one* is the final definition:

```
@String{j-ANN-PHYS-1900-4       = "Annalen der Physik (1900) (series 4)"}

@String{j-ANN-PHYS-1900-4       = "Ann. Physik (1900) (ser. 4)"}
```

The `volume` and `number` fields

Most journals from the 20th Century onward appear in (often annual) volumes divided into issues that may be printed irregularly, or at fixed intervals within the year (half-yearly, quarterly, seasonally, monthly, weekly, or even daily, as for newspapers). The issue number is recorded in the `number` field. Libraries normally bind all of the issues of a journal volume into one or more physical volumes to keep the issues together, and provide a hard cover to protect the contents.

Here are some examples that illustrate some of the variations that have been encountered in the archives:

```
  volume =       "17",
  number =       "3",

  volume =       "AC-30",
  number =       "7",

  volume =       "PAMI-9",
  number =       "5",

  volume =       "VIII",
  number =       "32",
```

When a publication is assigned to a volume range, there is usually no associated issue number, so the `number` field is omitted:

```
  volume =       "27--28",
```

The `pages` and `pagecount` fields

The `pages` field records a list of pages on which the article appears. For traditional journals, it is usually given as a page range, separated by two ASCII dashes, which is TeX's compact convention for an en-dash that identifies a range, and is distinct from a *hyphen* (coded as a single dash), a *minus sign* (coded as a single dash in TeX math mode, e.g., `$e^(i \pi) = -1$`), and an *em-dash* (coded as three ASCII dashes). It is important to use those four codings correctly, because they each have different lengths in a typeset document, and their incorrect use can be jarring to typographically sensitive readers.

For magazines, the `pages` value may be a comma-separated list of page numbers and/or page ranges, as in this example from an entry in a bibliography of ***Byte Magazine***:

```
  pages =        "177--180, 182, 184, 186, 188--189",
```

That extra complexity arises because magazines, and also some journals, may intersperse article pages with advertising pages. Sometimes, longer articles are continued later in the issue, in order to attract reader attention to articles near the beginning of the issue.

Warning: Page ranges should *never* elide leading digits: use `12345--12379`, *not* `12345--79`!

Article page numbers do not always advance: some journals have a column in each issue that begins on the last page, and then continues somewhere inside the issue. For example, the monthly ***Last Byte*** column of the ***Communications of the ACM*** that all ACM members receive might have this field value:

```
  pages =        "136, 135",
```

When an article is short enough to fit on a single page, its value should be recorded as a range:

```
  pages =        "17--17",
```

The reason is that it is then clear that the bibliographer has checked the article page range. If only a single page number is recorded, there is an ambiguity: is the article really only one page long, or was the bibliographer lazy? Many journals require correct page-range information in reference lists, so it is important to get the values right. BibTeX style files for some journals reduce a range with identical page values to a single number, if that is their editorial convention. However, the original BibTeX data should retain the truth.

When journal table-of-contents data are converted to BibTeX format, care is needed in determining page ranges, because usually only the starting page is listed. If the journal does not permit empty pages, and always starts each article on a new page, then it is possible to interpolate the missing final page numbers of all but the last article. However, if articles always start on an odd-numbered (right-hand, for left-to-right printing) page, then such interpolated ending pages could be off by one. Similarly, if a new article can start on the same page as the previous one ends, page-range interpolation is unreliable. Well-designed publisher Web sites provide correct page ranges, but sadly, many others still do not.

Broad access to the Internet has encouraged some journal publishers to move to a new model. Instead of delaying release until a suitable number of articles have been collected into an issue of convenient size (sometimes limited by postal regulations), accepted articles are made available online in advance of print publication.

If there is no particular ordering of articles within their issue, then page numbering can continue to increase monotonically through the issue, and often, the volume. Page numbers then might reflect the order of acceptance of articles.

Some journals in physics, and a few other areas, group articles by subject, and online publication as soon as an article is accepted means that uniformly increasing page ranges in an issue or volume are no longer possible. For example, in 2005, the ***Journal of Mathematical Physics*** changed to a new style of page numbering:

```
  pages =        "012101",
  pagecount =    "18",
```

Instead of a page range, the **pages** field now has a six-digit number, with leading zeros. Examination of successive articles shows that the first four are fixed for short intervals, then increase, but not uniformly: `012101, 012102, ..., 012107, 012301, 012302, ..., 012305, 012501, 012502, ..., 012504, 012701, ..., 012703, 012901, 013301, 013302, 013501, ..., 013506, 019901, 019902, 022101, 022102, ...`. The first two digits correspond to the issue number, but otherwise, there is no obvious connection of the four-digit prefixes to acceptance or publication dates, so the six-digit values just have to be considered arbitrary identifiers.

Comment: Sadly, the lack of monotonicity caused by the new page-numbering system makes it impossible for software to check for missing articles. Such checks have been routinely used in the journal-specific bibliographies in the archives, and have many times uncovered problems, even in the original publisher Web pages. For an online-only journal, loss of a database entry for an article generally means the entire loss of that article to journal readers. If a page gap could be detected, the loss could be reported to the publisher, but that is not possible when gaps cannot even be diagnosed.

The **pagecount** value records the number of pages in the article. Standard BibTeX styles do not recognize that field, so its value would normally be ignored. Neverthess, the Utah archives record it in thousands of entries in more than 100 different bibliography files.

In 2007, the ACM began a publish-on-acceptance model for some of its journals, but chose a different approach than that of the physics publishers. Each ACM article has its own page range, but that range always starts with **1**, and the article is assigned a sequential number within the issue. We then have BibTeX data that might look like this:

```
  pages =        "1--25",
  articleno =    "17",
```

However, that format now means that page numbers overlap, and page-gap checks for missing articles are no longer possible. After discussion with ACM journal production staff, the new ACM styles developed by this author and others instead use this encoding of the data:

```
  pages =        "17:1--17:25",
  articleno =    "17",
  numpages =     "25"
```

The article number can optionally appear in both parts of the page range, separated by a colon from the page range. If it does, then all previous bibliography style files will automatically include the article number in the output. If it does not, the new ACM styles use the `articleno` and `numpages` values to reconstruct a range. Depending on the style, the reference list output could then look like any of these:

```
... , J. ACM 53, 6, 978--1012 (Nov. 2006).            % traditional style

... , J. ACM 54, 3, Article 11 (June 2007), 38 pages. % new style

... , J. ACM 54, 3, 11:1--11:38 (June 2007).          % alternate traditional style
```

Here is what this author wrote in his documentation of the new styles:

> **The ACM Transactions and ACM plain bibliography styles use `numpages` if provided, but can also produce its value from `pages` values of the forms `1--37` and `17:1--17:37`. The condition for extracting the page count from those ranges is that the first page number must be `1`, and in the second case, that the same article number is prefixed to starting and ending pages. The range parsing is simplistic: the field is expected to contain exactly two or exactly four numbers, separated by nondigit characters. Thus, `17/1--17/37` and `17(1)--17(37)` are also recognized, although the latter produces a warning about unexpected text following the final page number.**
>
> > ***There is therefore no particular need to introduce the new field name `numpages` for the new ACM styles, and this author recommends that it be withdrawn in order to keep the database files usable with all existing bibliography styles.***
>
> **Similarly, the `articleno` field is also extraneous, and could be withdrawn, because the ACM styles can extract it from the four-number page range. Those new styles typeset the new article pagination in the form `Article 17, 37 pages`, whereas existing styles typeset it in the traditional form `17:1--17:37`.**

As more journals move to online publication, we may expect to see further variations in article page numbering. If hundreds of existing BibTeX bibliography styles are to be kept stable and unchanged, then bibliographers should always prefer page ranges of the form `17:1--17:25`. The many bibliography files for ACM journals in the archives adopt that practice, but supply `articleno` values, and omit `numpages` data.

The `day`, `month`, and `year` fields

As we noted earlier in the description of standard BibTeX abbreviations , for consistency, convenience, and ease of translation to other languages than English, month values are normally abbreviated to *three-letter* lowercase prefixes, `jan`, `feb`, ..., `sep`, `oct`, `nov`, and `dec`. If an issue is assigned to a range of months, then one can use BibTeX string concatenation , like this:

```
  month =        jan # "--" # mar,
  month =        oct # "\slash " # dec,
```

Regrettably, some article databases fail to record *month* data, even though such values are commonly assigned to journals and magazines. Two of the leading databases for mathematical publications have that serious flaw.

It cannot be assumed that *month* values are merely synonyms for issue numbers. While that is true for many journals, it is not true for all. The ***Smithsonian*** begins new monthly volumes in April, so issue number **1** corresponds to that month, not to January. For magazines, the issue month and year may be prominently displayed on the cover and spine, while the volume and issue number are hidden inside the issue front matter, often in tiny text in a footnote.

Year values should almost always be a *four-digit number*, but rarely, such as for a multivolume work issued over several years, a list of years, or a year range, might be appropriate:

```
  year =         "1905",

  year =         "1905--1909",

  year =         "2001 (vol. 1), 2003 (vol. 2), 2007 (vol. 3)",
```

However, a multi-volume work should normally be split into separate entries for each volume, so that it is possible to cite any particular volume unambiguously.

The original BibTeX styles made no provision for a day number, even though that is essential information for daily and weekly periodicals. That deficiency led some bibliographers to encode that information with string concatenation, like this:

```
  month =        jan # " 25",
  month =        "14 " # jul,
```

Strong recommendation: Please avoid that ugly practice! It illogically mixes independent data in a single field, and interferes with use of bibliographic data in other languages.

A few extended bibliography styles, notably the `is-*.bst`, `acm*.bst`, and `xchicago.bst` style files, recognize the `day` field and format it appropriately. You can then have entry data like these fragments from `einstein.bib`:

```
  journal =      "{Die Vossische Zeitung}",
  pages =        "33--34",
  day =          "26",
  month =        apr,
  year =         "1914",

  journal =      j-NY-TIMES,
  day =          "3",
  month =        feb,
  year =         "1929",
```

Comment: Although some newspapers and magazines assign volume and issue numbers, many give only *day*, *month*, and *year* data, so those three values are essential information in identifying when an article was published. Addition of support for a `day` field is therefore the *most pressing issue* for maintenance of existing BibTeX styles. If you are forced to use a style that lacks support for those fields when you need

them, you might consider creating a temporary entry that records the day number in the `note` field that all styles output when it is present.

The `publisher` and `address` fields

Publisher and address data pose no particular problem, and many entries in the archives use abbreviations to ensure compact and consistent coding of such data, such as these fragments from `einstein.bib`:

```
  publisher =    pub-GAUTHIER,
  address =      pub-GAUTHIER:adr,

  publisher =    pub-VIEWEG,
  address =      pub-VIEWEG:adr,
```

Entries with smaller publishers may just use explicit string values, like these:

```
  publisher =    "Verlag Teubner",
  address =      "Leipzig, Germany",

  publisher =    "Deutsche Verlags-Anstalt",
  address =      "Stuttgart and Berlin, Germany",
```

Traditional publishing of reference lists usually includes only the city, but that practice may be ambiguous, and reflects economy, and perhaps laziness, and even arrogance. The bibliography archives include the country, and for less well-known publishers, sometimes more detailed location information. As examples of ambiguity, consider *Paris*, which is likely to mean the one in France, but could also be cities in the US states of Idaho, Texas, or 11 others, and *London*, which is probably the one in England, but could be a university city in the Canadian province of Ontario, or a city in one of six US states. Because emigrants often took their favorite city and town names with them, such ambiguities are common in place names all over the world.

If you and/or your publisher want to have only a city of publication, or prefer compact publisher names, then when BibTeX string abbreviations are used, it is simple to provide following alternative definitions in a small document-specific bibliography containing data copied from larger bibliographies:

```
@String{pub-GAUTHIER            = "Gauthier"}
@String{pub-GAUTHIER:adr        = "Paris, France"}

@String{pub-GAUTHIER            = "Gauthier"}
@String{pub-GAUTHIER:adr        = "Paris"}

@String{pub-VIEWEG              = "Friedrich Vieweg und Sohn"}
@String{pub-VIEWEG:adr          = "Braunschweig, Germany"}

@String{pub-VIEWEG              = "Vieweg"}
@String{pub-VIEWEG:adr          = "Braunschweig"}
```

However, this bibliographer feels strongly that it is the job of the bibliographic database to provide *maximal information*. Any abbreviations employed in a formatted reference list should be handled by a style file, or if necessary, by manual tweaks to copies of the BibTeX data, or, in rare cases, to the reference list itself. Because that list is produced automatically when BibTeX is run, manual changes are highly inadvisable. If you use the *Unix* `make` utility for controlling BibTeX and (La)TeX runs, then it is a simple matter to insert additional commands to run the *Unix* stream-editor utility, `sed`, after each invocation of BibTeX.

The bibliography archives generally encode publisher addresses in their English-language form. However, for reference lists in other languages, you may wish to provide alternate string definitions, such as these

examples for Danish, French, and German text:

```
@String{pub-VIEWEG              = "Friedrich Vieweg og S{\o}n"}
@String{pub-VIEWEG:adr          = "Braunschweig, Tyskland"}

@String{pub-VIEWEG              = "Friedrich Vieweg et Fils"}
@String{pub-VIEWEG:adr          = "Braunschweig, Allemagne"}

@String{pub-VIEWEG              = "Friedrich Vieweg und Sohn"}
@String{pub-VIEWEG:adr          = "Braunschweig, Deutschland"}
```

That flexibility is a strong reason to prefer use of abbreviations for such often-repeated data, and one of the software tools described later, `publisher.sh` makes it easy to standardize entries to use abbreviations.

Publishers sometimes change cities or expand offices to additional cites, such as Prentice-Hall's move from Englewood Cliffs, NJ, to Upper Saddle River, NJ, Wiley's move from New York City, NY, to Hoboken, NJ, and Oxford University Press' opening of offices in New York City. For large well-known publishers, it is unlikely to matter which of those cities is referenced, and certainly, library catalog data often disagree on the city for the same publication. Here is how some of the bibliography archive files deal with such problems:

```
@String{pub-MORGAN-KAUFMANN       = "Morgan Kaufmann Publishers"}
@String{pub-MORGAN-KAUFMANN:adr   = "Los Altos, CA 94022, USA"}
@String{pub-MORGAN-KAUFMANN:adrsf = "San Francisco, CA, USA"}
@String{pub-MORGAN-KAUFMANN:adrbo = "Boston, MA, USA"}

@String{pub-OLIVER-BOYD         = "Oliver and Boyd"}
@String{pub-OLIVER-BOYD:adr     = "Edinburgh, UK; London, UK"}

@String{pub-PENGUIN             = "Penguin"}
@String{pub-PENGUIN:adr         = "London, UK and New York, NY, USA"}

@String{pub-SV                  = "Spring{\-}er-Ver{\-}lag"}
@String{pub-SV:adr              = "Berlin, Germany~/ Heidelberg,
                                  Germany~/ London, UK~/ etc."}

@String{pub-U-CHICAGO           = "University of Chicago Press"}
@String{pub-U-CHICAGO:adr       = "Chicago, IL, USA and London, UK"}

@String{pub-WILEY-VCH           = "Wiley-VCH"}
@String{pub-WILEY-VCH:adr       = "Berlin, Germany, Weinheim, Germany,
                                  and New York, NY, USA"}
```

Springer-Verlag and Elsevier are large publishers that have absorbed many smaller ones, and as a result, have publication offices in several cities. The `pub-SV:adr` definition contains `etc.` to indicate the omission of other addresses.

The practice in the Utah bibliography archives has been, from time to time, to produce sorted lists of quoted-string publisher names and addresses, ordered by frequency of occurrence. When sufficient repetitions of such data are found, new definitions are added to the tool that supplies abbreviations. Ideally, all **`publisher`** and **`address`** values in BibTeX entries should only use names of BibTeX string abbreviations, but there is often a limit to human energy, enthusiasm, and time for doing so.

There has been confusing documentation in the past that suggests that for **`@Proceedings{...}`** entries, the **`address`** value should hold the location of a conference. Ignore that nonsensical advice! The **`address`** value is that of the *institution*, *organization*, or *publisher*, just as it is for all other `DocumentTypes`. The conference location is commonly recorded in the **`booktitle`** and **`title`** values, like this:

```
  booktitle =    "{Trends and Topics in Computer Vision: ECCV 2010
                 Workshops, Heraklion, Crete, Greece, September 10--11,
                 2010, Revised Selected Papers, Part II}",
  title =        "{Trends and Topics in Computer Vision: ECCV 2010
                 Workshops, Heraklion, Crete, Greece, September 10--11,
                 2010, Revised Selected Papers, Part II}",
```

For those less common cases where the proceedings volume omits the conference location from the title, you can include it in a `note` field, like this:

```
  booktitle =    "{Sixty-two years of uncertainty: historical,
                 philosophical, and physical inquiries into the
                 foundations of quantum mechanics}",
  title =        "{Sixty-two years of uncertainty: historical,
                 philosophical, and physical inquiries into the
                 foundations of quantum mechanics}",
  note =         "Proceedings of a NATO Advanced Study Institute held
                 August 5--15, 1989, in Erice, Sicily, Italy.",
```

At least for conference series that move to new locations each time, the city name, rather than the year or conference number, is likely to be the tag by which human attendees remember the conference. It is therefore desirable to include the conference location in the BibTeX entry.

The `note`, `annote`, and `remark` fields

All standard BibTeX styles support a `note` field that is output near the end of the reference-list item. It is a catch-all for useful information about the publication that should always appear in the reference list. Examples of note material include additional credits, such as to an illustrator, translator, or the writer of a foreword or introduction, as well as cross references to related papers with comments, corrigenda, errata, notes, and remarks.

Many bibliographies found on the Web use an `annote` field to record additional remarks that are *not* intended to be copied to output reference lists. This bibliographer has never been comfortable with the use of that obsolete English word, so he instead employs `remark` fields for that purpose.

In some cases, it may be desirable to record multiple comments in a BibTeX entry. To improve readability, it is best to record them separately, like this:

```
  remark-1 =     "...",
  remark-2 =     "...",
  remark-3 =     "...",
```

The `CODEN` field

Partly because of the problem of journal-name ambiguity, in the 1950s and 1960s, researchers at the *University of Buffalo* and the *American Society for Testing of Material (ASTM)* introduced compact unique journal *code names* that are called `CODEN` values.

Originally, CODENs were four-letter strings derived from abbreviations of journal names, but they now classify more than 100,000 periodicals, and have been lengthened to six-character uppercase strings, where the first five are letters, and the sixth is an alphanumeric check digit that can be used to validate the `CODEN` value. In 1975, assignment and management of CODEN data was moved to the *International CODEN Service* of the Chemical Abstracts Service (CAS) of the *American Chemical Society*.

The CAS Source Index (CASSI) Search Tool provides a convenient way to convert between journal names, `CODEN` values, and `ISSN` values (discussed later).

CASSI searches return considerable useful information about journals, including full and recommended abbreviated names, publication history, language, publisher, and alternate titles. For current journals, there is also often a URL that provides a link to the publisher site for that journal.

You can see the CASSI results for the journal in our sample entry by following this link . Select **CODEN** in the pull-down menu above the search box, enter **ANPYA2** in the search box, and select the **Search** button.

There does not appear to be a freely available list of CODEN values and their corresponding journal names. The CASSI service gives only one-at-a-time access. The ASTM Web site offers access only to expensive old CODEN documents from the 1960s. It has taken this bibliographer much effort over several years to collect the CODEN data that are automatically provided to BibTeX entries by the `journal.sh` utility.

The bibliography archives at Utah supply CODEN values when they are available; they are assigned to journals in the fields of chemistry and physics, and sometimes mathematics, but are unlikely to be available for journals in other subject areas. The CODEN can also be found in numeric field **030** of the United States *Library of Congress MARC record* that is used to describe publications in library catalogs. MARC records are widely used in the US, Canada, and some other countries for interlibrary catalog-data exchange. Many libraries use the Library of Congress call numbers (LCCNs) in catalogs to identify the positions of publications on library shelves.

Web sites for library catalogs that use Library of Congress call numbers always have a way of displaying the raw MARC record data, but the normal default in catalog lookups shows only a subset of that data, reformatted for human readability, because the numeric fields of MARC records are distinctly unfriendly for people to deal with.

A few bibliography styles include CODEN values in the output reference list, but most do not. Nevertheless, the CODEN is a useful value to record in a BibTeX entry because it provides a unique handle, and because it provides convenient access to publication information and often, the journal Web location, via the CASSI search tool , and some library catalogs.

The DOI field

The **DOI** field is an extremely important new addition that did not exist when Scribe and BibTeX were designed in the 1970s and 1980s. Its value is a ***Digital Object Identifier*** that provides a *unique handle* by which the document may be found.

Officially, a DOI begins **10.**, followed by a publisher number and a slash. What follows the slash was originally completely up to the publisher, but that turned out to be unwise, and there is now less variation in that part. Typically, it is a document number in a publisher database; that is effectively a random alphanumeric string that is unlikely to related to the **volume**, **number**, and **pages** values, even though those values, together with a suitable journal identifier, could frequently make a predictable, and necessarily unique, DOI value.

Because most humans are unlikely to know how to lookup a document by DOI, it is better to include the prefix shown in our examples that makes it a valid Internet *Uniform Resource Locator* (URL) that is acceptable to any Web browser. Typically, following that link leads you to a publisher Web page for the document, offering additional publication information, possibly an abstract, and links to HTML and/or PDF forms of the document. Following those links to the full text may require a personal or institutional digital-library subscription, or payment of a fee.

Here are some sample DOI values:

```
  DOI =          "https://doi.org/10.1002/andp.19013090306",

  DOI =          "https://doi.org/10.1002/(SICI)1097-4571(1999)50:3<233::AID-ASI6>3.0.CO;2-8",
```

```
  DOI =          "https://doi.org/10.1073/pnas.18.3.213",
```

The first DOI encodes a journal name, and a database record number. The second DOI illustrates the silliness of some early DOI conventions that have since been dropped by that particular publisher, but must remain in use because DOI values are immutable. The last one, from the ***Proceedings of the National Academy of Sciences (PNAS)***, encodes the volume, issue number, and starting page, making DOI values for that journal predictable. Regrettably, few publishers follow that practice, which otherwise could make it possible to automatically retrofit valid DOI values into existing BibTeX entries for journal articles.

Some recently developed bibliography styles know how to supply, and strip, the `https://doi.org/` prefix in a DOI value, so a publisher that uses those styles can choose to display either form of a DOI in formatted reference lists.

Before 2017, the DOI agency used the prefix `http://dx.doi.org/` on the URL form of Digital Object Identifiers; their current recommendation is the slightly shorter `https://doi.org/` that also changes to an encrypted communications channel. The Utah bibliography archives, and most of the associated software tools, have all been converted to prefer the new prefix, but the old one is still recognized, and remains valid.

In those rare cases where spaces occur in DOI values, represent them in hexadecimal as `%20`, so as to avoid unwanted line breaks that cause confusion about where the DOI ends.

Sometimes DOI values are excessively long, or contain characters that are problematic for use in TeX macros. An alternate service from the DOI organization at **http://shortdoi.org** provides a reliable way to obtain a short form that is as persistent as the original DOI value. For example, when given the long DOI `10.1002/(SICI)1097-461X(1996)57:1<3::AID-QUA1>3.0.CO;2-1`, that service reports

```
  shortDOI®
  Your request was processed. The previously created shortcut for
  10.1002/(SICI)1097-461X(1996)57:1<3::AID-QUA1>3.0.CO;2-1 is the
  handle:

      10/b8sr3k

  The shortcut HTTP URI is:

      http://doi.org/b8sr3k

  This shortcut will return the same results as
      https://doi.org/10.1002/(SICI)1097-461X(1996)57:1<3::AID-QUA1>3.0.CO;2-1,
  and
      doi:10/b8sr3k
  can be used in place of
      doi:10.1002/(SICI)1097-461X(1996)57:1<3::AID-QUA1>3.0.CO;2-1.
```

There are similar services for URLs, though possibly not reliable long into the future: **http://bitly.com/** and **http://tinyurl.com/**.

For our long DOI, they return **http://bit.ly/2mH3Zl3** and **http://tinyurl.com/k4y4pz7**, respectively.

The `ISBN` and `ISBN-13` fields

The *International Standard Book Number (ISBN)* system was introduced in 1972 to provide a uniform, and worldwide, unique identifier for published books. Post-1972 reprints of older books generally get ISBN values assigned to them. The oldest such entry found in the Utah bibliography archives dates from 1888, and more than 3650 pre-1972 entries with ISBN values are present in the archives.

The ISBN is a ten-character value that is divided into four hyphen-separated groups. In early years, spaces could be used instead of hyphens, but that practice is now strongly deprecated. The first nine characters of an ISBN are digits, and the last may be a digit or the letters *X* or *x*. Those two letters are equivalent, and the uppercase one is conventional. The first group identifies the language or country of publication. The second group identifies the publisher. The third group identifies a book from that publisher. The fourth and last group always has a single character, and serves as a check digit to allow validation of the ISBN. Here are some examples:

```
ISBN =          "0-8065-0711-X",

ISBN =          "0-517-02530-2 (paperback), 0-517-02961-8 (hardcover)",

ISBN =          "0-585-11047-6 (e-book), 0-19-280086-8 (paperback)",
```

Each form of a book has its own distinct ISBN, as the second and third examples illustrate.

The ISBN is always printed on the back cover of a book, along with a machine-readable bar code. It also appears on the copyright page following the title page.

Because the number of data digits is fixed at nine, large countries, language groups, and publishers have small numbers. Here are some country/language-group values: **0** and **1** (English), **2** (French), **3** (German), **4** (Japanese), **5** (Russian, and former members of the USSR), **6** (unused), **7** (Chinese), ..., **99971** (Myanmar (Burma)), and **99972** (Faroes). Among the largest publishers in the English-language group are **00** (Collins, Harper--Collins, Fontana, and subsidiaries), **01** (unassigned), **02** (Collier, Macmillan, Prentice-Hall, and subsidiaries), **03** (Holt, Rinehart and Winston, and affiliates), **04** (Unwin), **05** (unassigned), **06** (Harper--Collins family), **07** (McGraw-Hill family), and **08** (Elsevier and Pergamon family). At the small end, at least in its early years, is **937175** (O'Reilly & Associates), which had space for only 100 books in its initial ISBN range. It has now grown enough to qualify for the numbers **596** and **4493**. There are even small publishers with a seven-digit number and space for only ten books.

When a publisher has used up its quota of book numbers, it must apply for a new publisher number. Eventually, numbers began to run out for some ISBN ranges, and in 2007, the ISBN system was extended to 13 digits in the form used by the *European (now International) Article Number (EAN)* system. The new values have a fixed prefix of **978-** or **979-**. The second prefix will come into use only after the first is exhausted for a particular publisher. The next 9 characters are from the 10-digit ISBN, and the final character is a new check digit that in general differs from that of the 10-digit ISBN. That convention makes it possible to convert from 10-digit to 13-digit forms, but eventually, newer 13-digit ISBN values with the **979-** prefix will not map to unique old 10-digit values. Here are some examples that correspond to our 10-digit ISBN samples:

```
ISBN-13 =       "978-0-8065-0711-8",

ISBN-13 =       "978-0-517-02530-7 (paperback), 978-0-517-02961-9
                (hardcover)",

ISBN-13 =       "978-0-585-11047-9 (e-book), 978-0-19-280086-2
                (paperback)",
```

Two of the bibliography tools described later can convert ISBN 10-digit values to their 13-digit forms, so the bibliography archives now supply both forms in every entry for which such data are available.

Although supplying both 10-digit and 13-digits forms is redundant, library catalogs will take many years to convert, so both forms may be needed for searching in different catalogs.

Because the ISBN-10 and ISBN-13 values are compact and unique identifiers for a book, it would be

sensible to use them in Web addresses. One scientific publisher, Springer-Verlag, has commendably done so, with DOI and URL values like these for a paper in one of the ***Lecture Notes in Computer Science*** series volumes that publish papers from the majority of conferences in computer science:

```
  DOI =          "http://dx.doi.org/10.1007/978-3-642-25704-9_1",
  URL =          "http://link.springer.com/accesspage/chapter/10.1007/978-3-642-25704-9_1",

  book-DOI =     "http://dx.doi.org/10.1007/978-3-642-25704-9",
  book-URL =     "http://www.springerlink.com/content/978-3-642-25704-9",
```

In the first two values, the paper number within the volume is separated from the ISBN-13 value by an underscore. The second pair of values provide a useful link to the Web pages for the entire volume.

That practice often makes it possible to guess a Web address for a book from its ISBN values, and if the Web page is found, its address can be recorded in the BibTeX entry. A small *Unix* shell script with some simple `awk` pattern matching, and a text-only Web browser (e.g., `elinks`, `links`, `lynx`, `netrik`, and `w3m`), or the `curl`, `gnurl` or `wget` programs, could then easily recover additional useful BibTeX data.

The ISSN field

The ***ISSN*** field is the ***International Standard Serial Number*** , a unique 8-digit value, where the last digit may also be the letters *X* or *x*. That last digit is a *check digit* that can be used to validate the ISSN value. The ISSN is a unique identifier for a periodical, and each form of the periodical has a unique ISSN. Many journals now have both a print ISSN and an electronic ISSN, for which we might use a value like this:

```
  ISSN =         "0001-0782 (print), 1557-7317 (electronic)",
```

Unlike ISBNs, the digit groups do not encode any meaning, such as a language or publisher. Indeed, the ISSN values of apparently similar journals from the same publisher tend to be completely different, as do those for different publication forms of the same journal.

A few bibliography styles include ISSN values in the formatted reference list, but most do not. Nevertheless, the ISSN is a useful value to record in a BibTeX entry because it provides a unique handle, and because it provides convenient access to publication information and often, the journal Web location, via the CASSI search tool , and the search facilities of most library catalogs.

Because it is compact and unique, the ISSN can sensibly be incorporated into Web addresses, and several publishers do just that. One of the software tools described later, `find-journal.sh` , suggests possible Web addresses for specified ISSN values.

The ISSN-L field

The ***ISSN-L*** field is a new field that is called the ***linking ISSN*** . For almost all journals with multiple ISSNs, it is identical to the print ISSN. It is intended to be the unique identifier that represents *all* publication forms of the periodical. For bibliography styles that include ISSN data in the output reference lists, it is recommended that the shorter ISSN-L value be used in preference to the possibly longer ISSN value.

The LCCN field

This field holds the US Library of Congress *call number* for the document. The call number is based on a human librarian's subject classification, and on what else is already catalogued in the subject, so it is not necessarily unique across libraries. However, many libraries share Library of Congress catalog

information, and shelve documents by their call numbers, so a call number should bring you close to the document's location in your library, and importantly, also help you to identify related documents. The call number usually, but not always, ends with the document's copyright year. Here are two examples from Einstein's books:

```
  LCCN =         "QC6 .E45",

  LCCN =         "QC6 .E5 1979; QC173.55",
```

The second example supplies two call numbers, separated by the preferred semicolon. The two locations are likely to be several shelves apart, but probably in the same shelf aisle.

Sometimes, a document appears to cover widely separated subject areas. Here are three examples from other bibliographies:

```
  LCCN =         "QA267 .S95 1979; TK7885.A1 S92 1979",

  LCCN =         "QA76.24 .F76 1986, T185.F76 1986, TK7828 .S48 1986",

  LCCN =         "QC371 .S67 v. 8157; TK5102.92 .S3745 2011",
```

The third of those examples suggests a long-running book series that is shelved together, and our entry is at volume 8157.

Warning: Library of Congress catalog data also include a *catalog number* that could be confused with LCCN (*call number*) values. The catalog number has meaning only within that single library, so it is not useful to record it in a BibTeX entry, and none of the entries in the files in the Utah archives does so.

The `MRclass`, `MRnumber`, and `MRreviewer` fields

Our sample entry for Einstein's first paper is not in the American Mathematical Society's MathSciNet Mathematical Reviews database, so none of the fields in this section's heading is present. Here is what some of them might look like for another famous paper, Andrew Wiles' 1995 triumphant proof of a famous 300-year old conjecture in mathematics, namely, that the generalization of Pythagoras' theorem about right triangles, $a^n + b^n = c^n$, has no positive integer solutions a, b, and c, for fixed integer $n > 2$:

```
  author =       "Andrew Wiles",
  title =        "Modular elliptic curves and {Fermat's Last Theorem}",
...
  MRclass =      "11G05 (11D41 11F11 11F80 11G18)",
  MRnumber =     "1333035 (96d:11071)",
  MRreviewer =   "Karl Rubin",
```

The `MRclass` values are five-character strings that the AMS uses for subject classifications that correspond to a detailed list of topics in mathematics. The `MRclass` values provide a convenient way to find related publications in the ***MathSciNet*** database. If you have access to that database, you can find subject codes in the *Mathematics Subject Classification* service. Even if you lack access to that database, you can obtain the free 47-page booklet *MSC2010* that is produced by the joint efforts of the AMS ***Mathematical Review (MR)*** and EMS ***Zentralblatt für Mathematik (Zbl)*** staff. From that booklet or service, here is an explanation of the classification codes in our sample:

- **11G05**: Elliptic curves over global fields;
- **11D41**: Higher degree equations; Fermat's equation;
- **11F11**: Holomorphic modular forms of integral weight;
- **11F80**: Galois representations;

- **11G18**: Arithmetic aspects of modular and Shimura varieties.

Such detailed subject classification is rare outside the mathematical community, and clearly has considerable value for researchers who want to find publications in specialized areas of mathematics.

The `MRnumber` value is seven-digit value that is a unique identifier for a publication recorded in the ***MathSciNet*** database. If you have access to that database, you can enter the number `1333035` into a search box, choose `MR number` in the pulldown menu for that box, and select the `Search` button. A successful search leads to a Web page that contains detailed publication information, links to other papers by the author(s), links to journal information, and often, a detailed review of the article.

Our example contains an additional parenthesized number, `96d:11071`. That number is a relic of one of *seven* older numbering systems used by the ***MathSciNet*** system, and it can still be used to locate the article. However, the seven-digit form is now the preferred identifier.

The `MRreviewer` field is supplied in ***MathSciNet*** search results in BibTeX format. Its value could be useful in identifying experts in the field of the reviewed publication, or additional publications by the reviewer, but no standard BibTeX style makes use of it.

The `onlinedate` and related fields

There is a long tradition in journal publishing of recording the dates of first receipt of a manuscript and those of any further revisions made in response to referee comments, and the date of final acceptance. Those dates are important for determining scientific priority and awarding credit to the first discoverers of an important new idea, product, or theorem.

For a few important historical papers, the bibliography archives include `accepted` and `received` fields values. Neither is used by current bibliography styles.

Now that some journals are published online in advance of printed issues, or are only published electronically, publishers report an additional date that records when an article was first made available to subscribers and paying customers. Their use of that date suggests the obvious field name `onlinedate`. Here is an example:

```
  onlinedate =   "23 December 2004",
```

So far, there is no evidence that a more precise time stamp is available from publishers, but there should eventually be such a value accurate to at least a microsecond. There is an interesting scholarly account in the book ***The Telephone Gambit*** (ISBN 0-393-33368-X (paperback), 0-393-06206-6 (hardcover)) of the patenting of the telephone within a few hours of the same day by Elisha Gray and Alexander Graham Bell. That patent has been called the most valuable patent ever, and at the time, applications were only stamped to the nearest day. The book discusses the probable priority of the patent applications: the winner was likely not the earlier one that day.

The `URL` field

Although computer networks certainly existed when Scribe, TeX, LaTeX, and BibTeX were developed, the World-Wide Web did not. Consequently, none of those programs anticipated the need for support of field values that could lead directly to a document on a network. The now-widely familiar convention is that the Internet *Uniform Resource Locator* (URL) supplies such a value.

About 60% of the entries in the bibliography archives at Utah now contain a `URL` field that records document handles.

The `URL` field provides a useful adjunct to an even newer field, the `DOI` field that we described earlier in this report. While each document is intended to have only a single DOI value, there may be multiple independent sources of the document on the Internet, and the `URL` field can record them. If there are multiple URL values, separate them with semicolons, because commas are sometimes seen in URLs. In the rare case that a URL contains semicolons, replace them with their hexadecimal equivalent in the ASCII character set, **%25**. Any problematic ASCII character can be represented in a URL as a percent character followed by two hexadecimal characters.

The intent of the designers of the URL system was that the handles should be *permanent*: once a document is placed on the Web and its address becomes known to others, it should *never* be removed. Sadly, that is often not the case. Newspapers are frequent offenders, but employers are too when they delete user accounts, and Web sites, of past employees. The *Wayback Machine* project at `http://www.archive.org/` attempts to archive the entire Web, but is not always successful, partly because it only finds out about documents by scans at intervals of days, weeks, or months, so it cannot discover Web documents that exist for only a short time.

While many journals now encourage use of `DOI` values in reference lists, they may require that a `URL` value be accompanied by a date recorded in a `lastaccessed` field to give the reader some indication of when the document was known to exist.

In some entries, `DOI` and `URL` values resolve to the same Web pages, even though the values may look quite different. That redundancy is nevertheless useful in those rare cases that the redirection service at `https://doi.org` is not reachable. An error in a `DOI` value returns a report that the document does not exist, but gives the user no clue as to which journal publisher holds the intended document. By contrast, the `URL` value contains the publisher Web address, and it may then be possible to find the document directly that way, or else use the publisher's search facility to track it down.

All BibTeX styles need to be extended to report `URL` and `lastaccessed` values in output reference lists, but only a few recent styles do so.

Including `URL` and URL-like `DOI` values in reference lists poses a problem for typesetters, especially for output in multicolumn journals and newspapers: the values are often long, and interfere with line breaking. Member of the TeX community have developed the `path` and `url` packages to apply special line-breaking rules on such data: because hyphens are common in the data, they cannot be used to indicate line breaking without introducing ambiguity in a value that must be known precisely if it is to be usable. The few BibTeX styles that recognize `DOI` and `URL` fields assume one of those packages, and wrap the values in macros that provide hyphenless line breaking. There is consequently *no need* to use those macros in BibTeX entries.

Warning: Be aware that a URL that contains a user login name prefixed with a tilde may no longer be accessible if the account is disabled or deleted. Such personal URLs are therefore likely to have a shorter shelf life than URLs in an organizational Web tree.

Warning: There is a second important caution that must be raised about URL data: they may contain personal or location information that should not be revealed. Sometimes, such extra data are obvious. For example, the ProQuest thesis database includes visible account information in URLs in its search reports. More commonly, such data are concealed in long alphanumeric or hexadecimal strings in the URL. If you use such a URL in a Web browser to reach a document, the URL is likely to be rewritten in the browser's

address box once the document is found. By that time, the remote site has decoded the hidden information, recorded something about you, and then remapped the URL to an innocuous one that you can safely record in a BibTeX entry.

The `ZMnumber` field

The European Mathematical Society has an extensive database of publications in mathematics and theoretical physics and theoretical computer science that goes even further back in time than does the AMS ***MathSciNet*** database. Like the latter, the EMS database, which is known by its German-language name ***Zentralblatt MATH***, assigns a unique number to each publication in its database, which it, and we, record in the `ZMnumber` field value.

The `ZMreviewer` field

Like the AMS ***MathSciNet*** `MRreviewer` field, the `ZMreviewer` field value records the name of the ***Zentralblatt MATH*** reviewer of the publication, and can be used for similar purposes. Its value is not used by any standard BibTeX style.

The `acknowledgement` field

This field's value is always a string abbreviation that provides credit to the creator of a BibTeX entry. It is not used by any standard BibTeX style, but is present in almost 99% of the entries in the Utah archives.

The `ajournal` and `fjournal` fields

These fields are used in the ***MathSciNet*** database to record abbreviated and full journal names. They are not used by any standard BibTeX style, but recording them in the entry is a convenience to users who may wish to switch between short and long journal names in their reference lists.

The *American Mathematical Society* publishes a free (online) booklet that contains recommended abbreviations of journal names, and their ISSN values, for use in reference lists. However, other publishers, and many authors, use different abbreviations. That variation confuses readers, and computer software. If the abbreviation is that of a foreign-language title, it may be quite difficult for identify the true name of the journal in order to find it in a library catalog, or on library shelves, or at its own Web site.

For example, when the author of this text first encountered a journal abbreviated as ***Fiz. Sz.***, he was unable to find it, despite searching in several major library catalogs. Eventually, he identified it as the journal ***Fizikai Szemle (Budapest)***, which means ***Physical Review*** in Hungarian. That confusion is one important reason why we have adopted the practice of recording full journal names in their string-abbreviation definitions.

Modern journals generally have short names, or nicknames that are widely understood in their user community (e.g., ***ACM Transactions of Mathematical Software*** is known as ***TOMS*** to its readers, and may be abbreviated ***ACM Trans. Math. Softw.*** in reference lists).

Past journal-naming practice was quite different: consider the famous journal known officially by the 161-character name ***Proceedings of the Royal Society of London. Series A. Mathematical and physical sciences (1905--1989), Containing papers of a mathematical and physical character***, for which we

use a string abbreviation of ***j-PROC-R-SOC-LOND-SER-A-MATH-PHYS-SCI***.

The `bibdate` field

Journal publications carry information about when articles were received, accepted, and published, and modern computer filesystems record timestamps for various kinds of access to files (*backup*, *read*, *modify*, *write*, ...). Such temporal data are important for ordering publications, and for managing filesystems.

The **`bibdate`** field, which is used in more than 98% of the entries in the Utah archives, records the date of last change to the record, in the format produced by the *Unix* (and *POSIX*) `date` command. It is not used by any BibTeX styles, but has proved exceedingly useful when BibTeX data are stored in a real database system, because it can be used to restrict database searches to subsets of the data. For example, maintenance and updating of author-specific and subject-specific bibliographies requires extracting entries from other bibliography files that have been added to the database since those specialized bibliographies were last updated.

When new entries are created, the **`bibdate`** field value is automatically supplied by all of the software developed in support of the Utah archives. In the `emacs` editor, when an existing entry is substantially modified, its **`bibdate`** value can be updated with just two keystrokes.

The `bibsource` field

Works of art are accompanied by letters of provenance that document their history and origin, in order to reassure future buyers that their purchases are authentic works, rather than clever forgeries. For similar reasons of end-user confidence, publications generally carry author, institution, and publisher names and addresses. The legal mechanisms of copyrights and trademarks are part of human society's controls against possibly substandard forgeries.

The **`bibsource`** field that is used in more than 99% of the entries in the Utah archives records information about Web locations of bibliography files, and names of library catalogs and publisher Web sites from which at least part of the data were originally taken. When the field value names multiple bibliographies, that may be a helpful clue to the user of the entry that there are other bibliographies, possibly not yet known to the user, that could contain relevant related information. Marketers often use that technique in advertisements: *"If you liked our product, you'll love this related product!"*. However, there is no commercial interest in the bibliography archives, and the information in **`bibsource`** fields is intended only to be helpful to the user.

Other field names in our sample entry

Our sample Einstein paper entry contains a few other field names, some of which are unique to the Einstein bibliography file . The `Calaprice-number` value records the index of this paper in a list of Einstein's publications given, with commentary about them, in Alice Calaprice's useful book ***The Einstein Almanac***. Her descriptions are a good way to understand the relations between, and the importance of, Albert Einstein's many works.

The `Schilpp-number` and `Whittaker-number` values are similar indexes into other published bibliographies of Einstein's works that are recorded in **`einstein.bib`** , and discussed in the comment preamble of that bibliography. Although the three numbers here agree in our sample entry, they differ in later entries, because the different bibliographers did not always identify the same set of publications. Einstein wrote a

lot of material, and sent hundreds of letters to newspapers. Editing of his collected works remains an ongoing, and still incomplete, project more than sixty years after his death on 18 April 1955 at the age of 76.

Our sample entry contains two additional entries that we have not assigned to subsections of this report.

The `language` field records the language of the publication, because that can be useful for later searches. It is used primarily for entries published in languages other than English, although it might name that language in multilingual publications:

```
  language =     "English; French; German",

  language =     "Italian, German, English, Spanish or French",
```

If you need to record Cyrillic-language text in a BibTeX entry, you can do so with Unicode UTF-8 encoding, but then your entire document-processing tool chain must be capable of handling Unicode. You can stick to plain portable ASCII if you transliterate the text to the Latin alphabet. The transliteration service at http://www.translit.ru/ may be helpful in guaranteeing consistency. Here are some examples from the Russian/English journal, ***Reliable Computing*** = ***Nadezhnye vychisleniia***:

```
  title =        "Uvaszaemije kollegi! ({Russian}) [{Dear} colleagues]",

  title =        "Interval'nye vychisleniia -- predmet isledovani{\u\i}
                 i polesnij{\u\i} instrument. ({Russian}) [{Interval}
                 Computations --- Subject of research and useful tool]",

  title =        "Predislovie. ({Russian}) [{Foreword}]",
```

The `xxvolume` field in our sample entry looked like this for Einstein's first paper:

```
  xxvolume =     "4",
```

It identifies a value that may be incorrect, but has been encountered in a literature citation of that paper. The prefix `xx` causes it to be sorted near the end of the entry by the `biborder` utility that is used to standardize field/value order in all BibTeX entries in the Utah archives. A few journals, including the one for our sample entry, restarted volume numbering from **1** when the journal was renamed with a new series number. That confusing old practice has fortunately almost disappeared, except for the Italian journal, ***Il Nuovo Cimento*** [The New Chemistry], which has had 10 different series since 1855. Journal issues may give both the volume number since the beginning of journal publication (here, `309`), and since the beginning of the current series (here, `4`). It is unclear which volume number should be preferred, so both are preserved in the entry, but no bibliography style should ever use the `xxvolume` value.

It is quite common for literature-citation sources to disagree about the values of some parts of the publication data: humans are often careless and error-prone, and whenever data are converted from one format to another, mistakes are likely. Our practice has been to preserve conflicting values in fields prefixed with `xx`, at least until the original journal paper can be found to resolve the discrepancies.

The `crossref` field

BibTeX provides several ways to describe a publication that appears inside another publication. There are three distinct `DocumentTypes`: `InBook`, `InCollection`, and `InProceedings`. The first is used to describe a chapter of a book that is written by the same author(s), and would usually have a `chapter` field value. The second is used to describe a portion of a volume that contains an edited collection of papers by authors who may differ from the volume editors. The third is used for papers in conference proceedings.

In each of those three `DocumentTypes`, you can supply all of the information in a single BibTeX entry, such as this example, found as the first entry of the *in-something* type in `einstein.bib`:

```
@InCollection{Einstein:1914:GSP,
  author =       "Albert Einstein",
  editor =       "Arnold Eucken",
  booktitle =    "{Die Theorie der Strahlung und der Quanten}.
                 ({German}) [{The} Theory of Radiation and the
                 Quantum]",
  title =        "{Zum gegenw{\"a}rtigen Stande des Problems der
                 spezifischen W{\"a}rme}. ({German}) [{On} the present
                 state of the problem of specific heat]",
  publisher =    "Knapp",
  address =      "Halle a.s., Germany",
  pages =        "330--352",
  year =         "1914",
  bibdate =      "Mon Oct 30 07:43:26 2006",
  bibsource =    "http://www.math.utah.edu/pub/tex/bib/einstein.bib",
  note =         "Republication of \cite{Einstein:1913:GSP}.",
  acknowledgement = ack-nhfb,
  bookpages =    "xii + 405",
  Calaprice-number = "34b",
  language =     "German",
  Schilpp-number = "63",
}
```

That entry contains the obvious standard `publisher` and `address` field values. It also contains two titles, and their corresponding page counts.

In most cases, however, there is a better way to encode an *in-something* entry. Convert it to *two* entries, with a cross-reference that connects one to the other:

```
@InCollection{Einstein:1914:GSP,
  author =       "Albert Einstein",
  title =        "{Zum gegenw{\"a}rtigen Stande des Problems der
                 spezifischen W{\"a}rme}. ({German}) [{On} the present
                 state of the problem of specific heat]",
  crossref =     "Eucken:1914:TSQ",
  pages =        "330--352",
  year =         "1914",
  bibdate =      "Mon Oct 30 07:43:26 2006",
  bibsource =    "http://www.math.utah.edu/pub/tex/bib/einstein.bib",
  note =         "Republication of \cite{Einstein:1913:GSP}.",
  acknowledgement = ack-nhfb,
  Calaprice-number = "34b",
  language =     "German",
  Schilpp-number = "63",
}

@Book{Eucken:1914:TSQ,
  editor =       "Arnold Eucken",
  booktitle =    "{Die Theorie der Strahlung und der Quanten}.
                 ({German}) [{The} Theory of Radiation and the
                 Quantum]",
  title =        "{Die Theorie der Strahlung und der Quanten}.
                 ({German}) [{The} Theory of Radiation and the
                 Quantum]",
  publisher =    "Knapp",
  address =      "Halle a.s., Germany",
  pages =        "xii + 405",
  year =         "1914",
  bibdate =      "Mon Oct 30 07:43:26 2006",
  bibsource =    "http://www.math.utah.edu/pub/tex/bib/einstein.bib",
  acknowledgement = ack-nhfb,
  bookpages =    "xii + 405",
  language =     "German",
}
```

The *in-something* entry is now shorter, and importantly, the book itself is now citable as a separate document. Also, new entries for other portions of that book can now be added to the bibliography without

unnecessary replication of information. Although the **year** value is the same in both entries, it is needed in other bibliographic software that processes BibTeX entries.

BibTeX makes only a *single pass* over the bibliography file, saving information in memory only for those entries that it has found citations for. Thus, when it reads the entry with citation label `Einstein:1914:GSP`, and has that on its list of *to-be-found* entries, its data collection includes the value of the **crossref** field, and it adds that value to its *to-be-found* list. Thus, the referenced entry must always *follow* the referencing entry in the bibliography file, usually with other intervening entries. The `bibsort` utility knows about that requirement, and automatically sorts all referenced entries in a separate group at the end of the output bibliography file.

If you fail to ensure correct ordering of cross-referenced entries, then you must either run BibTeX twice, or else you must list the database file twice in the `\bibliography{...}` command. Both of those workarounds are confusing and error prone, so it is best to be a regular user of `bibsort`.

Dissertations and theses

BibTeX provides only two kinds of entries for recording advanced academic degrees, as shown in these recommended minimal templates:

```
@MastersThesis{,
  author =       "",
  title =        "",
  type =         "",
  school =       "",
  address =      "",
  pages =        "",
  day =          "",
  month =        "",
  year =         "",
  advisor =      "",
}

@PhdThesis{,
  author =       "",
  title =        "",
  type =         "",
  school =       "",
  address =      "",
  pages =        "",
  day =          "",
  month =        "",
  year =         "",
  advisor =      "",
}
```

Notice that the field names are identical, and most styles set them identically as well. Such documents are really just a special kind of book, with an extra **type** field to record the kind of degree, and a **school** field replacing the **publisher** field. Internally, the publications usually include a special title page that records the degree, department, institution, and address. The front matter usually has some additional institution-specific pages, such as a record of the names of advisory-committee members. Apart from those special features, the rest of a thesis or dissertation should look like a normal book, with a table of contents, possibly lists of figures and tables, chapters containing sections, subsections, subsubsections, ..., followed at the end by possible appendices, a bibliography, and desirably (but sadly, rarely seen), a subject index. In some fields or institutions, chapters may be reproductions of published papers, possibly prefixed with a short introduction.

Although books are normally assigned only a copyright year, with no day or month data, all three values

are important to record for theses and dissertations.

The **type** field value is where the degree is recorded, and that is normally the only place where there is a distinction between degrees. Unfortunately, the value is subject to downcasing in some styles, so it *always* requires brace protection. Here are some examples that illustrate several variations in degree types:

```
  type =          "Diplomarbeit",
  type =          "Doctorate de 3{\`e}me cycle",
  type =          "Inaugural dissertation",
  type =          "Tesi di Abilitazione",
  type =          "Tesi di Laurea",
  type =          "Thesis (doctoral)",
  type =          "Thesis ({A.B., Honors})",
  type =          "Thesis ({Ph.D.})",
  type =          "Thesis ({S.M.} in Science Writing)",
  type =          "Thesis",
  type =          "{Docteur d'{\'E}tat {\`e}s Sciences}",
  type =          "{Ed.D.} thesis",
  type =          "{Habilitationsschrift}",
  type =          "{Honors B.S.} thesis",
  type =          "{M.S.} dissertation",
  type =          "{M.Sc.} thesis",
  type =          "{Master of Science}",
  type =          "{Masters of Science in Aeronautics}",
  type =          "{Masters} essay",
  type =          "{Ph.D.} dissertation",
  type =          "{Ph.D.} thesis",
  type =          "{Undergraduate} thesis",
```

The field whose value is merely `Thesis` is a deficient one, from a library catalog that did not record the name of the degree; the value should eventually be corrected.

Although some people contend that a *dissertation* corresponds to a higher degree than a *thesis*, that is not uniformly true. Naming conventions vary substantially across institutions, even among those within the same country. Notice that some of the values correspond to undergraduate bachelor's-level degrees, even though they reside in an entry with a different `DocumentType`.

Many countries have different degree traditions that do not correspond to the traditional bachelor's, master's, and doctoral degrees awarded in much of the English-speaking world. Germany, for example, has a *Habilitation* degree that follows a doctoral degree, and is required to hold a permanent teaching position at a German university. Scandinavian universities also have degrees that follow those that would be considered equivalent to doctoral degrees, but they are much less common than in Germany.

The **advisor** field in our templates is not output by current styles, but it is nevertheless useful, because it allows the links between teachers and students to be traced back in time. The ***Mathematics Genealogy Project*** records those links for more than 326_500 people in mathematics and related sciences.

BibTeX and online documents

We observed earlier that the development of BibTeX predated the World-Wide Web, and consequently, there is often confusion among new users of BibTeX about how to create BibTeX entries for online documents that they can then cite.

There are several candidate `DocumentType`s for such entries: `Manual`, `Misc`, `TechReport`, and `Unpublished`. For a particular online document, one of those may be a more obvious choice than the others. In the absence of a clear choice, the practice in the Utah archives has been to use `@Misc{...}` entries. Here are some examples:

```
@Misc{Einstein:1939:AEL,
  author =       "Albert Einstein",
  title =        "{Albert Einstein}'s Letter to {President Franklin
                 Delano Roosevelt}",
  howpublished = "World-Wide Web document",
  day =          "2",
  month =        aug,
  year =         "1939",
  bibdate =      "Mon Jun 18 17:52:21 2012",
  bibsource =    "http://www.math.utah.edu/pub/tex/bib/einstein.bib",
  note =         "Reprinted in \cite{Einstein:1963:EL}.",
  URL =          "http://hypertextbook.com/eworld/einstein.shtml;
                 http://www.anl.gov/Science_and_Technology/History/Anniversary_Frontiers/aetofdr.html;
                 http://www.lanl.gov/history/road/pdf/Einstein.pdf",
  acknowledgement = ack-nhfb,
  remark-1 =     "The letter, written at the urging of Leo Szilard,
                 Eugene Wigner, and Edward Teller, begins: ``Some recent
                 work by E. Fermi and L. Szilard, which has been
                 communicated to me in manuscript, leads me to expect
                 that the element uranium may be turned into a new and
                 important source of energy in the immediate future.''",
  remark-2 =     "Reprinted in \cite[pages 113--114]{Segre:1970:EFPb}.",
}

@Misc{Roosevelt:1939:LPA,
  author =       "Franklin D. Roosevelt",
  title =        "Letter to {Professor Albert Einstein}",
  howpublished = "Web document",
  day =          "10",
  month =        oct,
  year =         "1939",
  bibdate =      "Fri Aug 03 19:04:49 2012",
  bibsource =    "http://www.math.utah.edu/pub/tex/bib/einstein.bib",
  URL =          "http://www.pugetsound.edu/files/resources/7579_Roosevelt-einstein-letter.png",
  acknowledgement = ack-nhfb,
}
```

BibTeX and patent documents

There is no special BibTeX `DocumentType` for a patent, which is a specialized kind of document with important values that need to be recorded consistently, and therefore deserve to have standardized field names. Until an official `@Patent{...}` `DocumentType` is introduced and widely supported in BibTeX styles, the most suitable stop-gap entry type appears to be a `@Misc{...}` entry, formatted like these examples for important historical patents:

```
@Misc{Bell:1876:IT,
  author =       "Alexander Graham Bell",
  title =        "Improvement in Telegraphy",
  howpublished = "US Patent 174,465.",
  day =          "7",
  month =        mar,
  year =         "1876",
  bibdate =      "Thu Feb 14 10:56:14 2013",
  note =         "Application filed 14 February 1876. This first patent
                 award for the telephone has been called the most
                 lucrative patent in history.",
  URL =          "http://www.google.com/patents/US174465",
  acknowledgement = ack-nhfb,
  author-dates = "Alexander Graham Bell (3 March 1847--2 August
                 1922)",
}

@Misc{Einstein:1930:R,
  author =       "Albert Einstein and Leo Szilard",
  title =        "Refrigeration",
  howpublished = "US Patent 1,781,541.",
  pages =        "4",
  day =          "11",
  month =        nov,
```

```
  year =         "1930",
  bibdate =      "Tue Sep 13 15:09:21 2011",
  bibsource =    "http://www.math.utah.edu/pub/bibnet/authors/s/szilard-leo.bib;
                 http://www.math.utah.edu/pub/tex/bib/einstein.bib",
  note =         "Application filed December 16, 1927 (serial number
                 240,566) and in Germany, December 16, 1926. See
                 \cite{Dannen:1997:ESR,Dannen:1997:SRD} for accounts of
                 this invention.",
  URL =          "http://www.google.com/patents?id=t0BRAAAAEBAJ",
  acknowledgement = ack-nhfb,
  author-dates = "Albert Einstein (14 March 1879--18 April 1955);
                 Leo Szilard (11 February 1898--30 May 1964)",
}

@Misc{Szilard:1935:PSN,
  author =       "Leo Szilard",
  title =        "Provisional Specification No. 7840 1934.
                 {Improvements} in or relating to the Transmutation of
                 Chemical Elements",
  howpublished = "British Patent 440,023.",
  day =          "12",
  month =        dec,
  year =         "1935",
  bibdate =      "Thu Sep 15 10:31:19 2011",
  bibsource =    "http://www.math.utah.edu/pub/bibnet/authors/s/szilard-leo.bib",
  note =         "Patent applied for on March 12, 1934.",
  URL =          "http://nuclearhistory.wordpress.com/2011/02/28/szilards-chain-reaction-patents-440023-630726-loc
  acknowledgement = ack-nhfb,
  author-dates = "Leo Szilard (11 February 1898--30 May 1964)",
  remark =       "This is believed to be the world's first patent on
                 nuclear fusion, with applications to both power
                 plants, and to atomic weapons.",
}

@Misc{Fermi:1940:PPR,
  author =       "Enrico Fermi and Edoardo Amaldi and Bruno Pontecorvo
                 and Franco Rasetti and Emilio Segr{\'e}",
  title =        "Process for the Production of Radioactive Substances",
  howpublished = "US Patent 2,206,634.",
  day =          "2",
  month =        jul,
  year =         "1940",
  bibdate =      "Mon Jun 18 08:11:59 2012",
  bibsource =    "http://www.math.utah.edu/pub/bibnet/authors/f/fermi-enrico.bib",
  note =         "US Patent Application 43,462, filed October 3, 1935.",
  URL =          "http://www.google.com/patents/US2206634",
  abstract =     "The process, for production of isotopes including
                 transuranic elements by reaction of neutrons, employs
                 means for generating neutrons having a high average
                 energy, slowing down and scattering the neutrons by
                 projecting them through a medium of an element of a
                 class including H, He, Be, C, Si, and Pb, and then
                 passing the neutrons into a mass of material containing
                 an element capable of forming a radioactive isotope by
                 neutron capture, including radioactive isotopes capable
                 of emitting beta rays.",
  acknowledgement = ack-nhfb,
  author-dates = "Enrico Fermi (29 September 1901--28 November 1954);
                 Edoardo Amaldi (1908--1989); Bruno Pontecorvo
                 (1913--1993); Franco Rasetti (1901--2001); Emilio
                 Segr{\'e} (1905--1989),"
  remark =       "Turchetti \cite{Turchetti:2006:SNS} provides an
                 interesting historical study of Fermi's patent
                 involvement. He reports that in mid 1953, about 17
                 years after this patent was filed, the inventors
                 received US\$300,000 from the Atomic Energy Commission
                 (AEC) for their patent rights, an amount representing
                 only 0.005 percent of the revenues on their invention,
                 compared to the normal 5 percent given to inventors
                 before the War.",
}

@Misc{Eastman:1984:AMC,
  author =       "Willard L. Eastman and Abraham Lempel and Jacob Ziv
                 and Martin Cohn",
```

```
  title =        "Apparatus and method for compressing data signals and
                 restoring the signal",
  howpublished = "US Patent 4,558,302.",
  day =          "7",
  month =        aug,
  year =         "1984",
  bibdate =      "Thu Feb 14 10:33:02 2013",
  note =         "US Patent Application 291,870 filed 10 August 1981.
                 See also the related patent \cite{Welch:1984:HSD}
                 issued on the same day. This patent was held by Sperry
                 and its successor, Unisys, and vigorously enforced
                 until its expiry on 7 August 2001. It led to
                 multimillion-dollar lawsuit settlements, and the move
                 away from LZW compression to several other methods that
                 are patent free.",
  URL =          "http://www.google.com/patents/US4464650;
                 https://en.wikipedia.org/wiki/Graphics_Interchange_Format",
  acknowledgement = ack-nhfb,
  author-dates = "Abraham Lempel (10 February 1936--4 February 2023);
                 Jacob Ziv (27 November 1931--25 March 2023)",
}
```

Although patents are issued by national governments, they have been slow to make their data available online, so we instead use Google URLs for patents. Eventually, there may be official DOI data for patents.

BibTeX software tools

It is evident from the earlier examples in this report that BibTeX entries are verbose. An earlier bibliographic system in the *Unix* world, with the `bib` and `refer` programs, used a more compact encoding that might look like this for a subset of the data in our BibTeX entry for Einstein's first paper:

```
%A Albert Einstein
%T Folgerungen aus den Capillaritätserscheinungen
%J Annalen der Physik (1900) (series 4)
%V 309
%N 3
%P 513-523
%D 1914
```

Blank lines separate entries, and if there are multiple authors, there is one `%A` line for each. At least one commercial desktop-publishing bibliographic database system uses a very similar format, and some Web sites and online databases can return search results in a comparable format. The `reftobib` tool converts such data to rough BibTeX form.

While that format is easy to type, its use of single-letter field names makes it impossible to extend in any serious way. By contrast, entries in the Utah bibliography archives use several hundred different field names that have proven convenient for recording additional data that might not be used in a formatted reference-list item, but are nevertheless useful in searches.

Another nit is that single-letter field names quickly lose mnemonic significance: while `%A` is an *author* field, and `%T` is the `title` field, would you have guessed that a field named `%Q` holds a *corporate author*, and a field named `%X` holds an *abstract*?

Because BibTeX entries are extensible with new field names, and their format is defined by a rigorous grammar , BibTeX files can be used for much more than just BibTeX itself. They can serve as a master repository of rich bibliographic data that are easily searchable with common tools such as text editors, filesystem search utilities (like *Unix* `agrep`, `grep`, `egrep`, and `fgrep`), and Web search engines. They can be exchanged between different files and programs with cut-and-paste operations, sent in e-mail messages, posted on Web sites, grammatically validated with a BibTeX parser, heuristically checked for possible

errors, spell checked, doubled-word checked, delimiter-balanced checked, prettyprinted, ordered by fields, and have their entries sorted by several different dynamically constructed search keys. Entries matching specified search criteria can be extracted from, or deleted from, BibTeX files. Citation labels can be automatically generated by software, ensuring consistent, understandable, and easy-to-use labels.

Even more importantly, because of the use of extensible field names, software tools can automatically join entries for the same document, even when those entries differ in many details. That feature makes it possible to combine entries culled from many different source, merging their common data transparently to produce maximal information, and revealing discrepancies that indicate errors in one or more of the original sources. That technique has been exploited to a high degree in the production of the archives at Utah, allowing users to have considerable confidence that the data are substantially correct.

Searching with simple tools is often insufficient to quickly isolate desired bibliographic entries. Most searches, whether by string searches, or in library catalogs, or in online databases, produce far too many results, and the poor researcher then has to spend time throwing away most of them, or be so overwhelmed that she simplify declares defeat and abandons further searching. A good way to solve that problem is to install the BibTeX data inside a powerful SQL database, because that provides Boolean search operations (such as *this string AND that string AND NOT this other string AND year > 2000*), and importantly, field extraction and display (e.g., find all the article titles that contain the phrase *Maxwell–Boltmann distribution*).

No other bibliographic system provides such a rich array of possibilities. Although the software tools were developed in a *Unix* environment (a category that includes Apple Mac OS X, GNU Hurd, and GNU/Linux), they can be easily added to the free `cygwin` software layer that can be installed on most versions of *Microsoft Windows*. The *Cygnus Project* that supplies `cygwin` was founded to provide a *Unix*-like layer on top of the GUI-rich, but tool-poor, *Windows* operating system. The work of the *Cygnus Project* has made it possible to port many thousands of *Unix* programs to the *Windows* platform. More information about that topic can be found here . The inverse problem of making *Windows* software available on *Unix* platforms is discussed here .

More recently, Microsoft has introduced the *Windows Subsystem for Linux (WSL)* that makes it easy to install one or more entire Linux systems on top of Windows. The WSL subsystems behave like normal Unix-family machines, with the exception that package installations can be done by ordinary users, rather than requiring a Unix `root` administrator account. The WSL GUI windows behave like those in the Windows operating system, and text can therefore be cut and pasted between the different operating system environments. Files inside a WSL subsystem are also visible from Windows itself, and vice versa.

Here is a list of standalone command-line BibTeX tools that are freely available;

BibTeX software tools

`bibcheck`	Apply heuristic checks to BibTeX files to find possible, or probable, errors in field values and other data.
`bibclean`	Prettyprint and syntax-check BibTeX and Scribe bibliography database files, or alternatively, produce a rigorously parsed simple token list that can be used as input to other software tools that then need not deal with the syntax of BibTeX entries.
`bibdestringify`	Replace string substitutions in a BibTeX database, so the output entries no longer depend on prior `@String{...}` definitions. Such an operation would normally be discouraged, but could be useful in preparing selections of entries that are to be

sent to a publisher.

`bibdup` Check for duplicate abbreviations and entries in BibTeX files.

`bibextract` Extract BibTeX entries from a list of `.bib` files. This is particularly useful for creating small bibliography files to accompany articles sent to a publisher.

`bibisbn` Validate, and correctly hyphenate, ISBN values.

`bibjoin` Join duplicate or similar adjacent entries in a BibTeX file.

`biblabel` Generate standardized BibTeX citation labels defined in a substitution file that the companion `citesub` program can use to convert (La)TeX and BibTeX files to the new label values.

`biblex` Lexically analyze BibTeX files, producing a token stream that can be much easier to process by other software tools. The companion `bibunlex` program can convert the possibly modified token stream back to a correct and neatly formatted BibTeX entry.

`biborder` Rearrange field/value pairs in a BibTeX bibliography file into a standard order, and optionally, do extra checks, including converting 10-digit ISBN (International Standard Book Number) values to new ISBN-13 format, thereby supplying both in output field values.
Although in the absence of duplicate fields, field order does not matter to BibTeX, entries can be hard to read for a human unless they are neatly formatted, with fields in a consistent order.

`bibparse` Verify a bibclean or biblex lexical token stream, or BibTeX files. This tool is useful for quickly checking for basic syntax errors in a BibTeX file. It does not attempt to inspect field values for errors. One common use is to validate a BibTeX file before applying any other tool to that file, and before installing it in a Web archive or an SQL database.

`bibsearch` Search BibTeX bibliography files that have been converted to a special binary format for entries matching Boolean search queries. This tool is extremely fast (submillisecond searches even in a collection of a million entries), but the searches are against the entire entry. It is not possible to select only those entries with a certain author and a specified title string, possibly further restricted to a certain journal or publication year range.

`bibsort` Sort entries in a BibTeX bibliography file by any of several criteria: common choices are publication order for journal and series bibliographies, and citation-label order or year-then-citation-label for author-specific and subject-specific bibliographies. Sorting can be either ascending or descending. Most nonjournal bibliographies benefit by year ordering, because that makes it easy to identify the earliest and latest publications. However, this author's bibliography of books in his personal library is ordered by citation label, because that matches their shelf order.

`bibsplit` Split large BibTeX bibliography files into independent parts. The splitting can be according to any of several criteria, similar to those used by `bibsort`. Both BibTeX and TeX have internal memory limits that can be exceeded by large bibliographies: in practice, 4000 entries is a reasonable upper bound, with half that a recommended maximum. Many of the journal-specific and subject-specific bibliographies in the archives are split into files by decade or pentad. Huge files expand opportunities for an editing disaster that might go unnoticed until much

later when it could be very difficult or impossible to repair the mistake. Uppercasing, lowercasing, or deleting large chunks of data, are examples of such disasters that this bibliographer has had to recover from on multiple occasions.

`bibsql` Search an SQL (Structured Query Language) database of BibTeX data. This is the most powerful and flexible way of searching, because you can select output of all or part of the data, in several different formats.

`bibtex` This is BibTeX itself, and it should *always* be a standard part of any installation of TeX. It reads auxiliary files produced in prior (La)TeX runs, locates the requested bibliography style file and all of the requested BiBTeX database files, and then extracts and formats the cited entries in an output file of the same name as the input file, but with extension **`.bbl`**. Any warnings or errors appear on the screen, and in a companion log file with extension **`.blg`**.

`bibtosql` Convert BibTeX files to input acceptable to any SQL database supported by the companion program, `bibsql`. This is how BibTeX data get installed in, or updated in, an SQL database. At the bibliographer's site, regularly scheduled `cron` jobs run a small wrapper script that finds files changed since the last update, and invokes `bibtosql` to update the database with the contents of the changed files.

`bibunlex` Reconstruct a BibTeX bibliography data base file from `bibclean` or `biblex` lexical analysis output. This is typically used in a *Unix* shell pipeline with one of those two at the front, and intermediate pipeline stages that process, and possibly alter, the tokenized data.

`cattobib` This tool provides a convenient interface to scores of major library catalogs around the world that support the librarians' Z39.50 protocol for exchange of catalog information. That protocol is a distinctly user-unfriendly one that is impractical to use directly or interactively. Instead, that protocol is used by the **`yaz`** system that has proven to be highly portable across *Unix* platform. The `expect` program manages the communication with the selected remote Z93.50 server through the `yaz-client` program.
`cattobib` allows searching library catalogs by *author*, *editor*, *title*, *CODEN*, *ISBN* (both 10-digit and 13-digit forms), and *ISSN*. The results of the search are most commonly in US Library of Congress MARC record format, and that format is then converted to rough BibTeX form that can optionally be sorted internally with the help of `bibsort` . That is *much* better than creating entries by hand, but long experience shows that such entries always need further editing by a competent human to make them acceptable for literature citations.

`citesub` Replace old citation labels with new standardized BibTeX citation labels in one or more (La)TeX and BibTeX files. The substitution file can be prepared by hand, or automatically by the companion program `biblabel`.

`checksum` Create, or validate, an in-file checksum that is independent of the three common line-terminator conventions for text files. The tool supplies the internal checksums used early in the comment preamble of each of the BibTeX files and their LaTeX wrapper files in the bibliography archives. That makes it possible to verify that no corruption has occurred when the file has been moved between machines. The checksum field includes counts of lines, words, and characters, so even if the program is not installed locally, limited validation is still possible, with, for example, the standard *Unix* (and *POSIX*) `wc` utility.

`chkdelim`	Check delimiter balance, and report mismatches. Command-line options select special handling for different kinds of files, including TeX and BiBTeX files.
`detex`	Strip TeX commands from a file. This program is useful for recovering raw text from (La)TeX files, perhaps for input to other software tools.
`dw`	Find duplicate words. Such errors are surprisingly hard for human proofreaders to spot.
`doi-to-bibtex`	Convert one or more DOI values on the command line to rough BibTeX entries.
`find-journal.sh`	Given one or more ISSN values, suggest possible publisher URLs for them. This is sometimes a fast way to locate a publisher Web site for a journal.
`journal.sh`	Filter a BibTeX file and replace **`journal`** strings with standard abbreviations, and supply **`CODEN`**, **`ISSN`**, and **`fjournal`** values. The program knows about almost 4000 journals, and more than 9500 variations of their full and abbreviated names. It has ensured that more than 98% of the journal-article entries in the Utah archives have such additional, and highly reliable, data.
`publisher.sh`	Filter a BibTeX file and replace **`publisher`** and **`address`** strings with standard abbreviations. The program recognizes almost 1800 publisher names and abbreviations.
`reftobib`	Convert bibliographic data in *Unix* `bib` and `refer` format to rough BibTeX form.
`scholar`	Search the Google Scholar corpus for bibliographic data matching author or title words. BibTeX is one of its output formats.
`spell`	This standard *Unix* program reports input words that are not found in its built-in dictionary, or in any optional user-provided dictionaries. The resulting *exception list* can be inspected by a competent human for errors that can then be repaired in the input. All remaining words that are known to be correctly spelled can then be added to a private dictionary, so they never again appear in exception lists for that document. Because technical documents contain many specialized words, and also personal names from many foreign languages, document files should always be accompanied by a file-specific dictionary. In the Utah bibliography archives, every **`.bib`** file has a companion **`.sok`** file of correctly spelled words. Some systems may offer instead the `aspell` or `ispell`, and the book ***Classic Shell Scripting*** presents a compact portable multilingual spell checker as a literate program. All four of those programs are routinely used for spell-checking files in the Utah bibliography archives.

BibTeX editing tools

The previous section discusses standalone tools for dealing with existing BibTeX data. For manual creation of such data, it is best to choose a powerful text editor with specialized editing support for BibTeX entries. If your current text editor cannot do that, you might save considerable time in the future by changing editors. This report's author has long preferred the `emacs` editor, because it is by far the most powerful text editor ever developed, and because it has been ported to all common desktop platforms, so a comfortable and uniform editing environment is available on any computer. If you cannot use such an editor, you can save time by creating a small read-only file with suitable empty BibTeX templates that you can copy into your current file with cut-and-paste or insert-file operations. A minimal sample template for a book entry might look like this:

```
@Book{,
  author =       "",
  title =        "",
  publisher =    "",
  address =      "",
  pages =        "",
  year =         "",
  ISBN =         "",
  LCCN =         "",
}
```

In `emacs`, such templates can be inserted into the editor buffer with just three simple keystrokes, and a ***TAB*** character moves to the next empty field. Once the string fields are filled in, two more keystrokes magically generate a standard citation label after the brace on the first line. Four more keystrokes filter the newly created entry through a script that runs several other tools to further cleanup the entry and supply standard journal and publisher abbreviations. Creating a final clean entry thus takes fewer than ten keystrokes beyond the cut-and-paste or manual typing needed to supply the raw data. You can find nearly 50 `emacs` library files with more than 1400 BibTeX-related editing functions here.

Two features of `emacs` have been particularly helpful in bibliographic work: *dynamic word expansion*, and *filtering regions of text with command-line tools*.

The first is bound to a convenient two-key sequence that expands the word fragment at the current point to the closest word that begins with that prefix. If that is not the desired expansion, repeating the same keystrokes reexpands the prefix to words that are further away. The search for matching words extends into other editor buffers as needed, so if you like that feature, you could have a spelling dictionary in one buffer (e.g., `/usr/dict/words`), and then have an easy way to type long common words in other buffers. That simple feature has proved enormously helpful in dealing with long technical words and complex human names.

The second allows application of numerous programs, including most of the bibliographic tools, to restricted regions of text, such as a single BibTeX entry, or a single string value.

Other software tools

Conventional programming languages like Ada, C, C#, C++, Fortran, Java, Lisp, and Pascal are of little practical use for writing software for processing the many sources of data that can be mined to obtain data for new BibTeX entries, because programs in those languages are too complex, and too hard to modify, and most of those languages lack adequate support for powerful string-processing operations.

Instead, what is needed is a programming language that is specialized for string processing, and designed for compact and easy-to-modify program code. Two of the oldest languages for that job, `snobol` and `icon`, have largely fallen into disuse. The next on that scene was `awk`, developed between 1977 and 1985 at Bell Laboratories by members of the same team that created the influential *Unix* operating system. The `awk` language is still in wide use, and enjoys at least five independent implementations, four of them free and highly portable, and one commercial. Since its development, several other languages with somewhat similar goals have appeared, notably `javascript`, `perl`, `php`, `python`, and `ruby`. All of them are much more complex to learn and use than `awk`, and most have only a single implementation. Some have introduced incompatible versions that destroy portability of software written in those languages.

The `awk` language is the *only one* of the scripting languages mentioned in the preceding paragraph that is defined in an International Standard (*POSIX*). That standardization largely guarantees stability, and uniform behavior across different implementations. One caveat that should be mentioned is that a few

vendors provide a very early version of the language as `/bin/awk`, with the newer `/bin/nawk` corresponding to the *POSIX* version described in the 1987 book ***The AWK Programming Language***. A second caveat is that on GNU/Linux systems, `/bin/awk` may instead be identical to the greatly extended GNU version, `/bin/gawk`. Using language extensions could be a barrier to porting your software to other systems, so if you really need them, ensure that you reference the program as `gawk` instead of as `awk`.

`awk` is simple enough that this author has several times taught a one-hour class that covers most of the language, and more than 35 years of its use in the writing of many different software tools have not changed his view that it is exactly the right tool for string-processing jobs, if it is augmented by a few other programs that can be used to make the job easier.

The *Unix* (and *POSIX*) shell is another essential tool; its portable use is described in the book ***Classic Shell Scripting***, and in several other textbooks. The original Bourne shell, known just as `sh`, is the program that users interact with by default in a terminal window on any *POSIX*-compliant system. It is a full programming language that is rigorously defined in an International Standard (*POSIX*). Enhanced versions such as `ash`, `bash`, `dash`, `ksh`, `mksh`, `pdksh`, `yash`, and `zsh` make it even more convenient by providing job control (for switching between foreground and background processes), parallel processing, command editing, and command history and recall. Few of those features are needed for batch use, so you can achieve maximal portability if you stick to the original shell language in your shell scripts.

If you want to make an `awk` program available for general use by others, you can hide irrelevant details, such as where the program is stored in your filesystem, by wrapping its use in a shell script. That script can do other useful things, like creating temporary files, and ensuring that they are removed on both successful and failing runs of the script. It can also link multiple programs together, often with *Unix* command pipelines that direct output from one program to the input of another program, with data flowing rapidly through central memory without ever having to be written to, and read from, a filesystem.

As an extreme example of the shell-wrapper approach, at this author's site, a single 2800-line shell script, `journal-to-bibtex.sh`, supervises the processing of publisher Web data for about 580 different journals, automatically finding the correct input Web pages, and sending them through a 19-stage pipeline that transforms them from raw HTML in radically different markup and layout to clean, consistent, correct, and labeled BiBTeX entries that require only minor additional editing to supply protecting braces in title words that were not already recognized and braced by earlier stages in the pipeline. The tools invoked in that lengthy pipeline correspond to many hundreds of thousands of lines of code in several languages, yet each tool does only one, or a few jobs, and the great power of the conversion comes from their union in a shell pipeline. No monolithic program would be likely to do all of those jobs successfully, and still be easily modifiable the next time a publisher radically revises a journal Web site.

Many of the software tools described earlier are written in the `awk` language, with most of the remainder written in the *Unix* shell, or in the C language. The design goal has always been maximal cross-platform portability, because hardware changes much more rapidly than software does.

Of the C-language programs, two have been of supreme importance for the development and maintenance of the bibliography archives: `html-pretty` and `bibclean`. The first turns Web pages in arbitrarily formatted HTML and XML markup into consistently formatted files that are often amenable to simple regular-expression pattern matching in an `awk` program that can extract useful data, or just filter it for later processing by other tools. The second polishes and standardizes BibTeX files, and can also do numerous field validations.

Conclusions

The experiences, recommendations, samples, and warnings in this report have arisen over many years of use of BibTeX, and the archives cited earlier now contain millions of entries that provide ample data for understanding the needs of bibliographic databases and markup. It is quite likely that no other bibliographic-software system in wide use has been exercised as heavily as BibTeX has been since its development in the 1980s. It is a great tribute to the many members of the TeX research group under the leadership of Donald Knuth at Stanford University that their software has proved to be so robust, so portable, so long-lived, and so admired by its many users. Even though users of software sometimes run into design limitations that produce temporary frustration, and require workarounds, it is truly remarkable in the computing industry that TeXware has provided such a reliable platform on which documents and bibliographic databases can be expected to survive for decades, and perhaps even for centuries.

Many of the software tools that we described have been applied many thousands of times to large volumes of real BibTeX data, and we have considerable confidence that they behave as documented. From time to time, new kinds of BibTeX data turn up that suggest the need for tool extensions. This author's practice has been to add those ideas to a `TO-DO` list that guides development of the next version of the software. After a suitable local test period during which the tool is exercised many times on varied data, if no further problems, or ideas, turn up, the version is frozen and the code is released on the Web. Thus, the software evolves with the BibTeX bibliography archives, and the bibliographer's job becomes easier after each evolutionary advance in software.

Two of the most important lessons of using BibTeX have been that markup is *important*, and markup must be *flexible*. While common things, like the names of most authors and most titles, can be largely free of markup, the clear identification of the different logical parts of a reference-list item gives software, and humans, great flexibility in dealing with it.

BibTeX markup allows straightforward translation of BibTeX entries to input acceptable to SQL databases, and then even more powerful, and fast, search facilities become available. Once you are comfortable with SQL queries, it is possible to create intricate queries that can be saved and reapplied routinely, and can answer questions about a large corpus of data that would be completely impractical to answer with simpler search tools, or worse, unreliable visual inspection at human speeds. You also come to understand how impoverished are the GUI interfaces to online library catalogs and databases: they may make simple queries *easy*, but they make many possible queries completely ***impossible***.

Because SQL is defined by several ISO Standards, and is mission critical for many of the world's industries and governments, it is in use worldwide. Knowledge of how to query an SQL database in one area largely carries over to SQL databases in other areas: all that you have to do is learn the new table and field names. You may even discover that you yourself have data-management needs or problems that could be well-served by getting your own data into an SQL database. As long as you consider that database to be a *derivative* of your original data, you never even have to give up control of your data to the database.

If your data volume is large, but still fits in the memory or filesystem of a personal computer, you can use the `SQLite3` program to create and query the database. That program is so simple to use that any competent computer user can quickly become comfortable with it, and its files can be moved unchanged from the smallest hand-held computers to the largest supercomputers. The `bibtosql` and `bibsql` programs, and perhaps a few of the other software tools listed earlier, particularly, `bibclean`, provide all the software that is needed for SQL access to bibliographic data.