

## Software

### XBibTeX and Friends\*

Nickolas J. Kelly and Christian H. Bischof

#### Abstract

The XBibTeX utility provides an X Window interface for inserting entries into a bibliography database in BibTeX format. XBibTeX provides a template of the required and optional fields for each entry type defined by BibTeX. In addition, XBibTeX accommodates two additional entries named “keyword” and “annotate” for maintaining an annotated bibliography. The user enters the information to be stored in the database; XBibTeX automatically inserts the entry in the bibliography file in a format that conforms to BibTeX’s specifications. We also introduce two related utilities, **bibprocess** and **bibsearch**. **bibprocess** allows one to merge the contents of user-defined fields into a .bbl file produced by BibTeX; **bibsearch** allows one to retrieve the L<sup>A</sup>T<sub>E</sub>X key of publications that contain user-specified keywords. This article describes the features of these programs.

#### 1 How to Use XBibTeX

While BibTeX provides a convenient mechanism for storing and retrieving bibliographic records, it lacks a convenient means for entering new items. The user can easily make mistakes — for example, forgetting the required fields, the double quotes, or the commas surrounding the information in an entry.

XBibTeX minimizes these errors by

- displaying a template with all the fields defined for a particular type of bibliographic entry,
- making sure that information for required fields is indeed provided, and
- storing the information in a format that is guaranteed to be syntactically correct.

The ‘look and feel’ of XBibTeX is best conveyed through an example. After typing “xbibtex”, the **Application Window** in Figure 1 will appear. This window contains a menu of options which the user may select by pressing the left mouse button over the appropriate option.

The *Article*, *Book*, *Inproceedings*, and *Techreport* buttons are bibliography entry types defined by BibTeX. A menu containing the other (in our experience, less common) entry types known to

\* This work was supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U. S. Department of Energy, under Contract W-31-109-Eng-38.

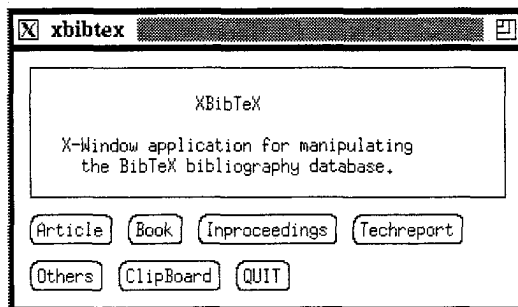


Figure 1: Application Window

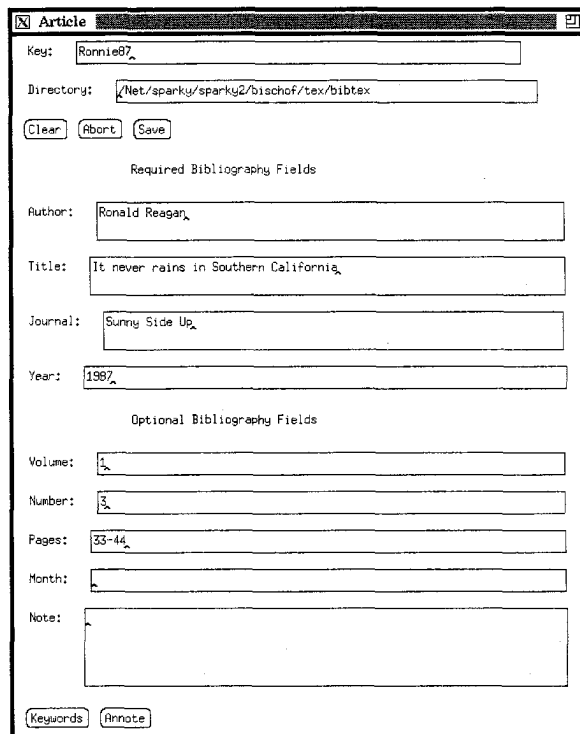


Figure 2: Entry Window for an Article

XBibTeX pops up when the user presses the *Others* button.

Let us assume we enter an article into our bibliography database. After pressing the *Article* button, the **Article Entry Window** displayed in Figure 2 will pop up, but with empty text fields. Notice that all of the buttons in the **Application Window** have been grayed except for the *ClipBoard* button. This “grey-out” prevents one from exiting the application while entering an entry.

There are two kinds of fields: “required” and “optional”. The “required” fields must be filled in for XBibTeX not to complain. For example, let us assume that the author is Ronald Reagan; the title of the article is “It never rains in Southern California”; the date of publication is 1987; the page range is 33–44; and the volume, issue, and journal title

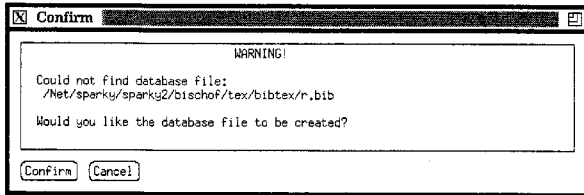


Figure 3: The Confirmation Window

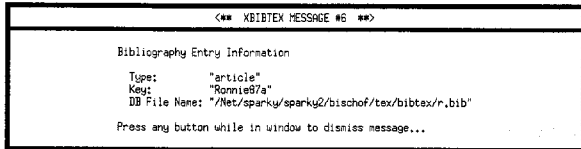
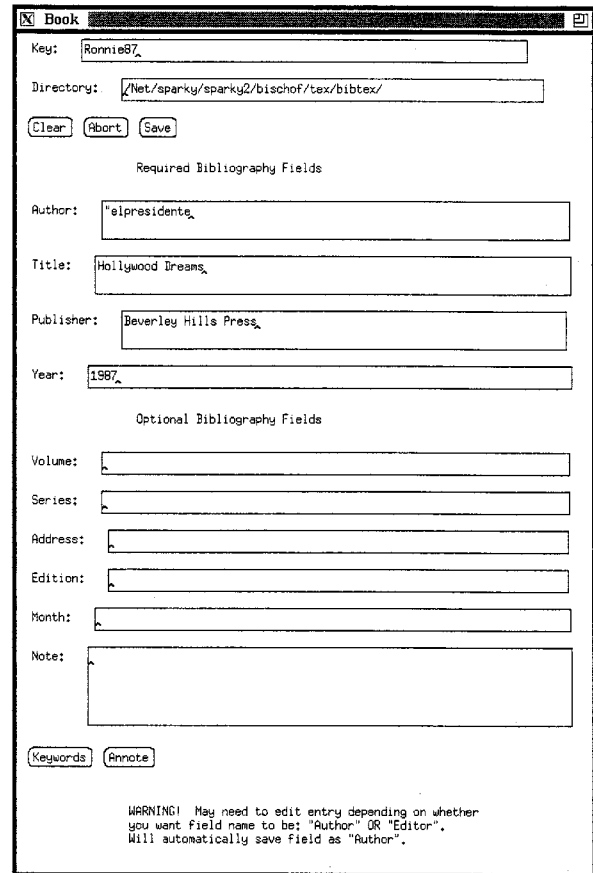


Figure 4: A Sample Message Window

are Vol. 1, No. 3, *Sunny Side Up*. Then one would enter the text as displayed in Figure 2. Note that there is no need to type the double quotes that surround information in `BIBTEX` entries. `XBIBTEX` will automatically generate them.

In addition, we have to specify a “key”, and we enter “Ronnie87” (say). One has to specify a directory where the database (`.bib`) files are kept. The path can be an absolute path or can be relative to the directory from which `XBIBTEX` was started. The path entered is not expanded by the shell, so the tilde (`~`) notation does not work, for example, to specify your home directory. Since database files are usually kept in one directory, the environment variable `XBIBTEX_DATABASE_PATH` can be set to specify this path so it need not be reentered every time.

When all the information has been entered, pressing the *Save* button causes the **Article Window** to disappear, and the **Confirmation Window** (see Figure 3) will appear since the database file `r.bib` does not yet exist. Upon confirmation, the **Message Window** (see Figure 4) will be displayed. The **Message Window** identifies the key that should be used from now on in a `LATEX \cite` command to refer to this entry. Notice that the `LATEX` key is not the same as the `XBIBTEX` key that was entered — the `LATEX` key is the `XBIBTEX` key with an “a” appended to it. `XBIBTEX` automatically indexes entries by appending a lower-case letter. This procedure allows for multiple entries to have the same `XBIBTEX` key, but has the restriction that one can create only twenty-six entries with the same “base” `XBIBTEX` key. The **Message Window** also shows in which file the new entry was stored. The database file name is determined by the first character of the `XBIBTEX` key with the “`.bib`” extension appended.

Figure 5: Entry Window for a *Book*

Now let us repeat the same procedure, except this time a book will be entered into the database. Let us assume the book is, say, “Hollywood Dreams”, again by Ronald Reagan, published by Beverly Hills Press in 1987. After opening the “book” template by pressing the *Book* button in the **Applications Window**, we enter the information as shown in Figure 5. Note that we entered the same `XBIBTEX` key as before, namely, “Ronnie87”. Since there is already an entry that used the same `XBIBTEX` key (and was stored with the `LATEX` key “Ronnie87a”), the new entry will be assigned the `LATEX` key “Ronnie87b”. Also note that we entered “elpresidente” in the author field. If an entry starts with a leading quote and has no spaces, newlines, or tabs in it, it will be saved without quotes surrounding it. Therefore, a declaration like `@string{elpresidente = "Ronald Reagan"}` causes this entry to be expanded correctly by `BIBTEX`. The **WARNING!** message at the bottom of the template indicates that if one wanted to enter an ‘editor’ field instead of an ‘author’ field (`BIBTEX` requires one of the two), one has to edit the `.bib` file by hand after storing this entry, since `XBIBTEX`

assumes that the “author” field has been entered.

If we press the “annotate” button, another window pops up into which additional information can be entered, which will be stored as an additional entry labeled “annotate”. `BIBTEX` will ignore this entry since it is not one of those that `BIBTEX` knows about. (The “`bibprocess`” utility described in Section 2 makes it possible to merge this information with output generated by standard `BIBTEX`.)

After saving this entry, the contents of the `BIBTEX` database file will look as follows, assuming that the file “`r.bib`” did not exist or was empty before, and “What great stuff!” was entered in the “annotate” window.

```
@article{Ronnie87a,
  author = "Ronald Reagan",
  title = "It never rains in Southern California",
  journal = "Sunny Side Up",
  year = "1987",
  volume = "1",
  number = "3",
  pages = "33--44"
}

@book{Ronnie87b,
  author = elpresidente,
  title = "Hollywood Dreams",
  publisher = "Beverly Hills Press",
  year = "1987",
  annote = "What great stuff!"
}
```

This example provided a flavor of the functionality that `XBibTEX` affords. In summary, `XBibTEX` has the following features:

- Automatically provides a template of the required and optional fields associated with each entry type as defined by `BIBTEX`.
- Ensures that the required fields of a specific entry are filled.
- Produces a unique `LATEX` key by which entries are referenced.
- Automatically inserts entries in alphabetical order (by key) into the database.
- Provides a clipboard that can be initialized to display a specified file. This feature can be useful for displaying an abbreviations file from which to cut and paste.
- Provides two new fields, “annotate” and “keywords”, that are useful for maintaining an annotated bibliography.

## 2 `bibprocess` and `bibsearch`

In this section we describe two related utilities:

**`bibprocess`**: which allows one to merge the content of additional fields added to `.bib` files into the `.bbl` files produced by `BIBTEX`, and

**`bibsearch`**: which allows one to search for the `LATEX` keys of bibliography entries containing the desired information.

### 2.1 `bibprocess`

As mentioned above, `XBibTEX` allows one to enter two additional fields (“annotate” and “keywords”) which are ignored by `BIBTEX`. **`bibprocess`** provides the functionality to merge the contents of those fields (as well as others) with the `BIBTEX` output. Consider the output produced by `BIBTEX` when we cite “Ronnie87a” and “Ronnie87b”. (For future reference, assume that it was written to `test.bbl`.)

```
\begin{thebibliography}{1}

\bibitem{Ronnie87b}
Ronald Reagan.
\newblock {\em Hollywood Dreams}.
\newblock Beverly Hills Press, 1987.

\bibitem{Ronnie87a}
Ronald Reagan.
\newblock It never rains in Southern California.
\newblock {\em Sunny Side Up}, 1(3):33--44, 1987.

\end{thebibliography}
```

Since “annotate” is a keyword that is not known to `BIBTEX`, the information in the “annotate” field (i.e., “What great stuff!”) was ignored.

The **`bibprocess`** utility makes it possible to merge this information into the `.bbl` file. The syntax is

```
bibprocess
<bibfile> [ <bibfile> ... ] <bblfile>
-f <formatfile>
-k <keyword> [ -k <keyword> ... ]
[ -o <outputfile> ]
```

Given a `<bblfile>` (a file with a `.bbl` extension) produced by `BIBTEX` from a set of `<bibfiles>` (files with a `.bib` extension), **`bibprocess`** appends the contents of the fields specified by `<keyword>` to the corresponding entries in `<bblfile>`. The output is written to `<outputfile>` if the `-o` option is present, and to `stdout` otherwise. The `<formatfile>` specifies how the new information is to be formatted.

Again, it’s easiest to understand this process with an example. Assume that we wish to merge the “keywords” and “annotate” information that `XBibTEX` allowed us to specify. The original `BIBTEX` output is in `test.bbl`, the bibliography

file is `r.bib`, and the merged output should go to `testmerged.bbl` (say). We issue the command

```
bibprocess r.bib test.bbl -f merge.fmt
-k keywords -k annotate
-o testmerged.bbl
```

The file `merge.fmt` contains the following information:

```
annotate
before: @\newblock {\bf annotate:} @
after: @}@
```

```
keywords
before: @\newblock {\bf keywords:} @
after: @}@
```

We see that we have an entry for each keyword that is to be merged, and for each keyword we specify what string to place *before* and *after* the information to be merged. The “at” sign `@` is used as a delimiter, the entries must be separated by blank lines, and the `<keywords>` (“`annotate`” and “`keywords`” in our example) must be specified with lower-case characters. `testmerged.bbl` then contains the following information:

```
\begin{thebibliography}{1}

\bbitem{Ronnie87b}
Ronald Reagan.
\newblock {\em Hollywood Dreams}.
\newblock Beverly Hills Press, 1987.
\newblock {\bf annotate:} What great stuff!}

\bbitem{Ronnie87a}
Ronald Reagan.
\newblock It never rains in Southern California.
\newblock {\em Sunny Side Up}, 1(3):33--44, 1987.

\end{thebibliography}
```

We see that the “`annotate`” information has been appended to the “`Ronnie87a`” entry using the format specified in `merge.fmt`. There was no “`keywords`” information, so no field was created for it.

Using `bibprocess` you can thus augment the functionality of `BIBTEX` and augment its output with any number of user-defined fields.

## 2.2 bibsearch

The syntax of `bibsearch` is as follows:

```
bibsearch <bibfile> -k <keyword>
[-k <keyword>] [-o <outputfile>]
-s <searchword>
```

`bibsearch` searches the title field and the fields specified by `<keyword>` in a bibliography file `<bibfile>` for an occurrence of `<searchword>` and prints the `LATEX` keys of all entries either to `<outputfile>` or `stdout`. For example,

```
bibsearch r.bib -k annotate -s "great stuff"
will produce
```

```
Ronnie87b
```

on the standard output. Note that one must enclose the search string in quotes if it contains more than one word.

In summary, `bibsearch` is useful for retrieving citations containing information on specified words—in particular, if one maintains a “`keywords`” entry in the `BIBTEX` database.

## 3 How to Obtain XBibTEX and Friends

The source for `XBibTEX`, `bibprocess`, and `bibsearch` as well as a more detailed user and installation guide can be retrieved via anonymous ftp from `info.mcs.anl.gov` (current Internet address: 140.221.10.1). The `pub/xbibtex` subdirectory contains the following files:

**userguide.ps.Z:** A more complete description of `XBibTEX`, `bibprocess`, and `bibsearch`, as well as installation instructions.

**xbibtex.shar.Z:** Source code, man page, and makefile for `XBibTEX`.

**bibutils.shar.Z:** Source code, man pages and makefiles for `bibprocess` and `bibsearch`.

All files must be transferred in ‘binary’ mode.

- ◊ Nickolas J. Kelly  
Nielsen Advanced Information  
Technology Center  
Bannockburn Lake Office Plaza  
2345 Waukegan Rd.  
Bannockburn, IL 60015
- ◊ Christian H. Bischof  
Mathematics and Computer  
Science Division  
Argonne National Laboratory  
Argonne, Illinois 60439-4801  
bischof@mcs.anl.gov