XBibTEX
and Friends

Nickolas J. Kelly*
kilroy@naitc.naitc.com

Christian H. Bischof†
bischof@mcs.anl.gov

Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, Illinois 60439-4801

Preprint MCS–P193–1190

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 "annote" 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**. The first allows one to merge the contents of user-defined fields into a .bbl file produced by BibTEX; the second allows one to retrieve the LATEX key of publications that contain user-specified keywords. This report contains a users guide as well as installation and customization instructions for XBibTEX, **bibprocess**, and **bibsearch**.

1 An Example of How to Use XBibTEX

BibTEX is a bibliographical database utility described in [2]. 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 syntactially correct.

*Participant in the Spring 1990 Science and Engineering Research Semester Program at Argonne. This program is coordinated by the Division of Educational Programs. Kelly's current address is Nielsen Advanced Information Technology Center, Bannockburn Lake Office Plaza, 2201 Waukegan Rd., Ste. S-200, Bannockburn, IL 60015.

†The work of this author 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.

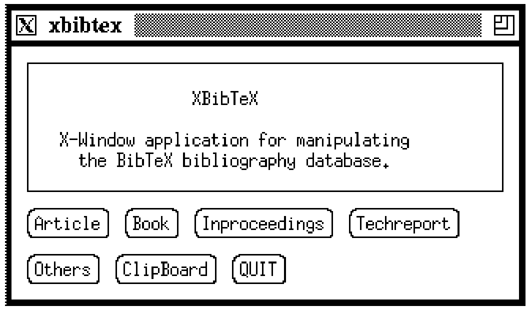

Figure 1: Application Window

To demonstrate the convenience provided by XBibTeX, let us go through an example. We'll assume that XBibTeX has been properly installed (see Section 4). Start it up by 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 XBibTeX pops up when the user presses the *Others* button.

Let us assume that you wish to enter an article into your bibliography database. Press the *Article* button. The Article **Entry Window** displayed in Figure 2 will pop up, but with empty text fields. You may notice that all of the buttons in the **Application Window** have been grayed except for the *ClipBoard* button. This "grey-out" protects you from exiting the application while entering an entry. (The *ClipBoard* is discussed in Section 2.)

There are two kinds of fields: "required" and "optional" fields. 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 are Vol. 1, No. 3, *Sunny Side Up.* Then you would enter the text as displayed in Figure 2. Note that you do not type the double quotes that surround information in BibTeX entries. XBibTeX will automatically generate those.

In addition, you now must specify a "key"; we'll enter "Ronnie87." You also have to specify a directory where your 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, you can set the environment variable XBIBTEX_DATABASE_PATH to specify this path so it need not be reentered every time (see Section 2).

When all the information has been entered, press the *Save* button. The **Article Window** will 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 you should use from now on in a LaTeX `\cite` command to refer to this entry. You will notice that the LaTeX key is not the same as the XBibTeX key that you entered—the LaTeX key is the XBibTeX key with an "*a*" appended to it. XBibTeX automatically

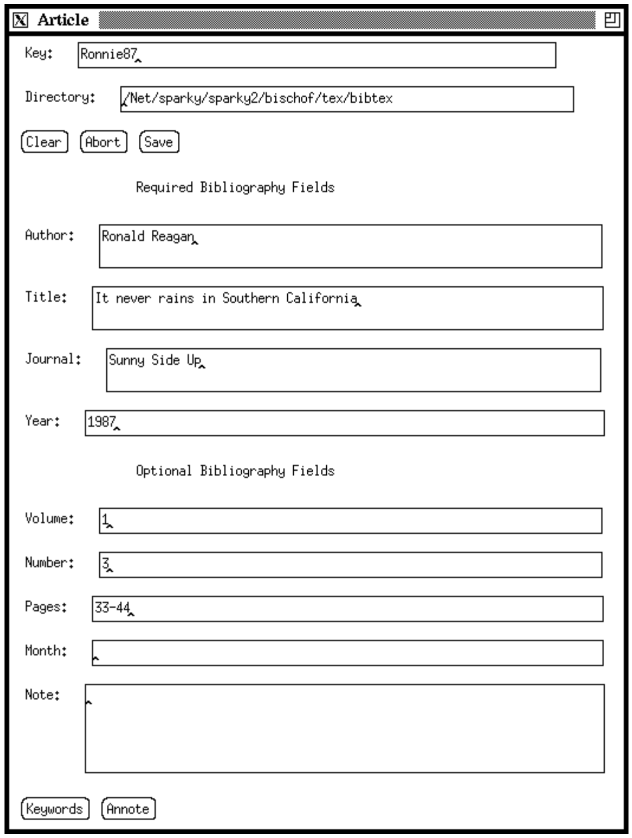

Figure 2: Entry Window for an *Article*

Figure 3: The Confirmation Window

Figure 4: A Sample Message Window

indexes entries by appending a lower-case letter. This procedure allows for multiple entries to have the same XBibTeXkey, but has the restriction that you can create only twenty-six entries with the same "base" XBibTeX key. The **Message Window** also shows you 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.

Now let us repeat the same procedure, except this time we'll enter a book, say "Hollywood Dreams," again by Ronald Reagan, published by Beverley 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, if you have a declaration like
`@string{elpresidente = "Ronald Reagan"}`
, then this entry will be expanded correctly by BibTeX. The WARNING! message at the bottom of the template tells you that if you have an editor, but no author field (BibTeX requires one of the two), you'll have to edit the .bib file by hand after storing this entry, since XBibTeX assumes that the "author" field has been entered.

Now go ahead and press the "annote" button. Another window pops up into which you can enter information that will be stored as an additional entry labeled "annote". BibTeX will ignore this entry since it is not one of those that BibTeX knows about. (The "bibprocess" utility described in Section 3 will allow you to merge this information with output generated by standard BibTeX.

After you have saved 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 that you entered "What great stuff" in the "annote" window.

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

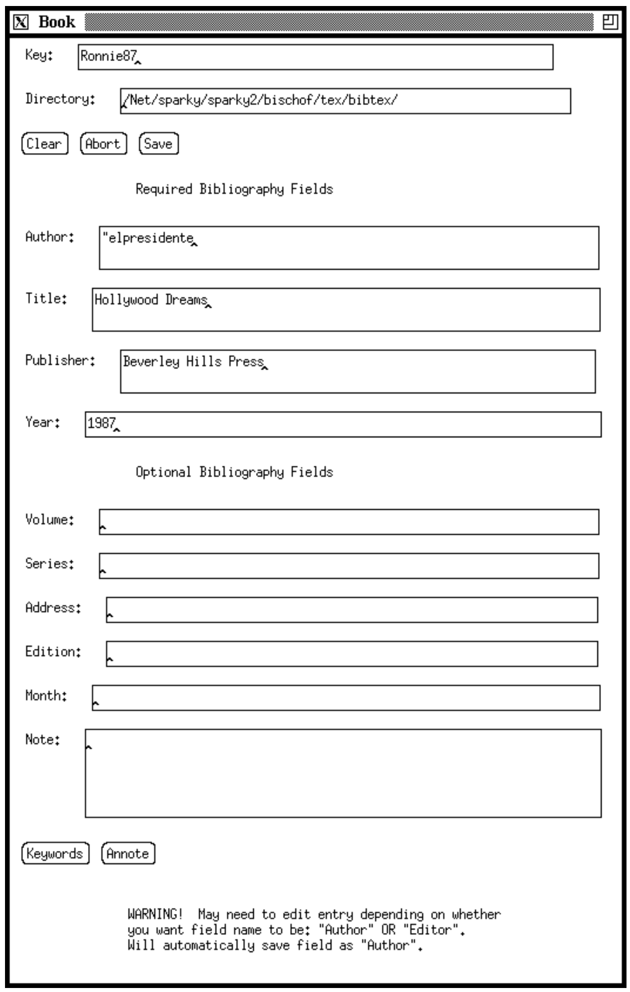

Figure 5: Entry Window for a *book*.

```
<**  XBIBTEX MESSAGE #3  **>

ERROR!  Can not save entry to bibliography database.

Press any button while in window to dismiss message...
```

Figure 6: Sample error message

```
  pages = "33--44"
  }

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

To exit XBibTEX, now press the *Quit* button.

2 More Detail

The example in the preceding section 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 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, "annote" and "keywords," that are useful for maintaining an annotated bibliography.

Before discussing XBibTEX's default settings, environment variables, X window resources, and input restrictions, let use say a few words about XBibTEX's messages. We've already shown the confirmation message that is displayed when an application is saved. Figure 6 shows a sample error message when not enough information is provided. When such a message is displayed, access to the rest of the application is suspended. This pause allows you to read the message and determine the

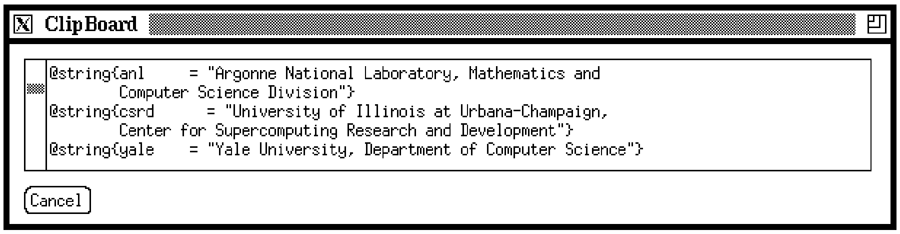

Figure 7: Clipboard Window

action to be taken. Once you have read the message, you can eliminate it by placing the cursor within the **Message Window** and pressing any mouse button. The Appendix contains a listing of all of XBibTeX's messages; we will refer to messages by the numbers listed there.

Backup File Directory: The first and most important thing to do before starting up XBibTeX is to make sure that there is a directory for XBibTeX to store its temporary files into. The default directory is "/tmp." If this directory does not exist, Message #8 will be displayed at the startup of the application. You should exit XBibTeX and set the environment variable XBIBTEX_BACKUP_PATH to a directory to which you have write access.

If your database ever becomes corrupted because of a system shutdown while you are modifying the file, you can go to the backup directory to retrieve the version since your last save. The backup file name is *xbib[pid].bkp* , where *pid* is the process ID of this XBibTeX application.

Database Path and Files: Once you have decided where your database file(s) will be located, you may wish to set the environment variable XBIBTEX_DATABASE_PATHto this directory path. This step will save you the time of typing the database path for every entry. The path can be specified either as an absolute path or as a path relative to the directory from which XBibTeX is started up.

BibTeX requires that bibliography database files have the ".bib" extension. XBibTeX extends this by splitting the database file into twenty-six files (a.bib through z.bib). For instance, all entries with keys beginning with "*b*" will be placed in the file "b.bib". In addition, entries are stored alphabetically in each file.

If you do not have write access to the directory or database file, XBibTeX will display Message #17 or Message #18 respectively. If the directory path does not exist, Message #5 will be displayed.

ClipBoard: If you wish to make the ClipBoard operational, you must set the environment variable XBIBTEX_CLIPBOARD. By setting XBIBTEX_CLIPBOARD to the name of a file that contains frequently used information, you can pop up (by pressing the *ClipBoard* button), a read-only window containing the information stored in this file, and cut-and-paste from this window into others. A sample **ClipBoard Window** is shown in Figure 7.

If XBIBTEX_CLIPBOARD has not been set, XBibTeX will display Message #12 when you press the *ClipBoard* button. If this environment variable has an invalid path or filename, Message #9 will be displayed. When the clipboard file is displayed, the *ClipBoard* button, is grayed to make it inoperative. That is, you can display only one **ClipBoard Window** at a time.

Application Resources: You may wish to load the resources file ".xbibrc" to the resource database using the command "xrdb -merge .xbibrc." This will give your application color (if you have a color monitor).

Clear/Abort Buttons: Practically all text for an entry can be lost by pressing either the *Clear* or the *Abort* buttons, so caution must be taken. The *Clear* button clears all text fields except the directory and key field. The *Abort* button pops down the **Entry Window** (e.g., **Article Window**), in effect killing all text entered in text fields.

Key: A valid XBibTEX key must begin with a letter (a–z). If this is not the case, Message #10 or Message #11 will be displayed. The key may be entered in uppercase or lowercase letters; but when it is saved, the first character is converted to upper case and the remaining ones to lower case.

XBibTEX appends an index letter from *a* through *z* to every XBibTEX key to generate the LATEX key. This way you may use the same XBibTEX key more than once, and XBibTEX will generate different LATEX keys. This method does have the restriction that you cannot reuse any given XBibTEX key more than twenty-six times.

Required *vs* **Optional Fields:** Required fields are those that must contain text for XBibTEX to save an entry to the BibTEX database. If you fail to enter text into these fields, Message #1 will be displayed.

To simplify programming for less frequently used entries, we have made the following fields mandatory, even though they are optional in BibTEX:

Entry Type	Text Field
Booklet	Author Year
Manual	Year
Misc	Title
Unpublished	Year

In the Book and Inbook entry types, BibTEX requires either an "Author" or an "Editor" field. XBibTEX allows you to specify an author; if you wish to store an editor, you should store it as author and then edit the BibTEX database file generated by XBibTEX. Similarly, in the Inbook entry type, BibTEX provides the option of "Chapter" and/or "Pages" field type. Here XBibTEX provides for "Chapter"only.

Quotes: Double quotes entered in a text field are removed unless they appear at the beginning of the text, in which case they signify that the text is to be written to the database file *without* double quotes around it. Such a field may not contain any spaces, tabs, or newlines; it must be a constant string of characters. If you violate these rules, Message #13 or Message #14 will be displayed.

Keywords and Annote Fields: XBibTEX provides the *keywords* and *annote* fields for the maintenance of an annotated bibliography database. Depending on the field selected, an item
`keywords = "<information you entered>"`
or
`annote = "<information you  entered>"`
is added to the bibliography entry. If an "annote" or "keywords" window was opened, it must either be cancelled or be nonempty. Otherwise Message #1 will be displayed.

3 bibprocess and bibsearch

In this section we describe two related utilities:

bibprocess: which allows you to merge the context of additional fields added to .bib files into the .bbl files produced by BibTEX, and

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

3.1 bibprocess

By now, you may be wondering how to merge the information provided in the "annote" and "keywords" fields into the output created by BibTEX. As we mentioned before, BibTEX will ignore these two fields.

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 Beverley 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 "annote" is a keyword that is not known to BibTEX, the information in the "annote" field (i.e., "what great stuff!") was ignored.

The **bibprocess** utility allows you to merge this information into the .bbl file. Its 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, things are easiest to understand with an example. Assume that we wish to merge the "keywords" and "annote" information that XBibTEX allowed us to specify. The original BibTEX output is in `test.bbl`, the bibliography file was `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 annote
           -o testmerged.bbl
```

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

```
annote
before: @\newblock {{\bf annote:} @
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>` ("annote" and "keywords" in our example) must be specified with lower-case characters. `testmerged.bbl` then contains the following information:

```
\begin{thebibliography}{1}

\bibitem{Ronnie87b}
Ronald Reagan.
\newblock {\em Hollywood Dreams}.
\newblock Beverley Hills Press, 1987.
\newblock {{\bf annote:} What great stuff!}

\bibitem{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 "annote" information has been appended to the "Ronnie87a" entry using the format specified in `merge.fmt`. There was no "keywords" information, so no fields were created for it.

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

3.2 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 annote -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.

4 Installation Guide

This section describes how to install XBibTEX, **bibprocess,** and **bibsearch.**

4.1 XBibTEX Installation

Below is a list of files needed for XBibTEX :

Makefile:	make.xbibtex		
X-Resource file:	.xbibrc		
C-Header file:	xbibtex.h		
C-Source files:	xbibtex.c	levels.c	confirm.c
	field.c	entry.c	opt.c
	save.c		

For XBibTEX to compile and execute properly, you must check that

- you are running X11 release 4;

- you have access to the X-Toolkit library (see [1]);
- you have the X-Athena Widget Set (see [3]);
- the makefile, header file, and all source files are contained in the same directory;
- the macro X_DIR in "makefile" is set to the directory where X stores its header files; and
- you have access to the libX11, libXt, libXaw, libXmu, and libXext libraries.

If these conditions are satisfied, typing

```
make -f make.xbibtex
```

will create "xbibtex" in the current directory.

4.2 bibprocess and bibsearch Installation

Below is a list of the files needed for **bibprocess** and **bibsearch**:

Makefile:	make.bibutils		
C-Header file:	globalbib.h		
C-Source files:	bibprocess.c	bibsearch.c	
	hash.c	init.c	input.c
	parse.c	save.c	

To compile **bibsearch** and **bibprocess** in the current directoy, issue the command

```
make -f make.bibutils <target>
```

where <target> is either bibsearch or bibprocess, or missing, in which case both programs are compiled.

Acknowledgments

We thank Bill Gropp and Remy Evard for their friendly help with X Windows programming.

Appendix: XBibTEX Messages

Each message has a severity level associated with it indicating the seriousness of the information reported. A summary is given in Table A-1.

Table A-1: **Message Severity Levels and Their Interpretation**

1 ADD_REQ_MSG (Severity Level 3)
ERROR! Text not entered in [field name] field.
Either enter text to this field or eliminate this field with the *Cancel* button.

Severity Level	Interpretation
1	Information— no action required
2	Possible modifications in BibTEX file needed
3	Required information for saving an entry missing
5	XBibTEX error

2 AUT_EDI_MSG (Severity Level 2)

WARNING! May need to edit entry depending on whether
you want field name to be "Author" or "Editor."
Will automatically save field as "Author."

3 CANT_SAVE_MSG (Severity Level 3)

ERROR! Cannot save entry to bibliography database.

4 CHA_PAG_MSG (Severity Level 2)

WARNING! May need to edit entry depending on whether
you wish the field name to be "Chapter" and/or "Pages."
Will automatically save field as "Chapter."

5 DIR_EXIST_MSG (Severity Level 3)

ERROR! Could not find Directory "[path]."

6 ENTRY_INFO_MSG (Severity Level 1)

Bibliography Entry Information
Type: "[entry type]"
Key: "[reference key]"
DB File Name: "[database path & file name]"

7 GET_LEV_MSG (Severity Level 5)

Program ERROR!
Check all calls to function GetWidget_from_Level().
Trying to access widget from invalid level.

8 INV_BKPATH_MSG (Severity Level 5)

The environment variable "XBIBTEX_BACKUP_PATH" was set to an invalid path.
Exit XBibTeX and set XBIBTEX_BACKUP_PATH to a valid path.

9 INV_CLIPFILE_MSG (Severity Level 2)

Cannot access the file specified by the environment
variable "XBIBTEX_CLIPBOARD".
The "ClipBoard" button will be disabled.

10 KEY_FMT_MSG (Severity Level 3)

The key "[Key]" is incorrectly formatted.
The first character must be a letter.

11 KEY_REQ_MSG (Severity Level 3)
ERROR! No key entered! Enter a valid key, and re-save.

12 NO_CLIPENV_MSG (Severity Level 2)
The environment variable: "XBIBTEX_CLIPBOARD" has not been set.
The "ClipBoard" button will be disabled.

13 QUOTE_FMT_MSG (Severity Level 3)
Text field "[field name]" is not in correct format for an abbreviation.

14 RM_QUOTES_MSG (Severity Level 1)
[num of] double quote(s) removed from Entry field "[field name]."

15 STORE_LEV_MSG (Severity Level 5)
Program ERROR!
Check all calls to the function StoreWidget_to_Level().
Trying to store widget to invalid level.

16 TXT_REQ_MSG (Severity Level 3)
ERROR! Text not entered in "[field name]" field.

17 WRT_FILE_MSG (Severity Level 3)
You do not have write access to file
"[file name]."

18 WRT_PATH_MSG (Severity Level 3)
You do not have write access to directory
"[path name]."

19 default (Severity Level 5)
Program ERROR!
Check all calls to function Popup_Messageshell().
Unknown error type passed to function.

References

[1] Joel McCormack, Paul Asente, and Ralph R. Swick, *X Toolkit Intrinsics – C Language Interface, X Window System, X Version 11, Release 4*, December 1989.

[2] Leslie Lamport, LaTeX *User's Guide and Reference Manual*, Addison-Wesley, Reading, Mass., 1986.

[3] Chris D. Peterson, *Athena Widget Set – C Language Interface, X Window System, X Version 11, Release 4*, December 1989.