

# HyperBIBTEX

A HyperCard bibliographic database manager for BIBTEX

Evan L. Antworth  
Summer Institute of Linguistics  
E-mail: [evan.antworth@sil.org](mailto:evan.antworth@sil.org)

## H

yperBIBTEX is a HyperCard application for managing bibliography databases in a format compatible with BIBTEX,<sup>1</sup> the bibliography formatting program commonly used with the LATEX document processing system<sup>2</sup>. LATEX is based on T<sub>E</sub>X typesetting language.<sup>3</sup> While T<sub>E</sub>X is widely used in the general academic community, it remains little known in SIL circles. Besides its usefulness in general document processing, T<sub>E</sub>X can also be used for specialized linguistic purposes. For example, Kew and McConnel's Interlinear Text Formatter (*ITF*)<sup>4</sup> is based on T<sub>E</sub>X. While HyperBIBTEX will be immediately useful only to LATEX users, it demonstrates a number of features of general interest to HyperCard users and developers and to those interested in bibliography management. With this wider audience in mind, we will start with a brief overview of BIBTEX.

## About BIBTEX

BIBTEX uses files of bibliographic entries in a special database format. Figure 1 shows a sample entry.

---

<sup>1</sup>See Patashnik 1988.

<sup>2</sup>See Lamport 1986.

<sup>3</sup>See Knuth 1986.

<sup>4</sup>See Kew and McConnel 1991.

\

**Figure 1** A sample BIBTEX entry

```
@BOOK{kew-mcconnel90,  
AUTHOR = "Kew, Jonathan and McConnel, Stephen R.",  
TITLE = "Formatting interlinear text",  
YEAR = "1990",  
SERIES = OPAC,  
NUMBER = "17",  
ADDRESS = "Dallas, TX",  
PUBLISHER = SIL}
```

\

The string `@BOOK` signals the beginning of a Book entry. `BIBTEX` recognizes thirteen standard entry types: Article, Book, Booklet, InBook, InCollection, InProceedings, Proceedings, Manual, TechReport, MastersThesis, PhdThesis, Misc, and Unpublished. Each entry type specifies a set of required and optional fields, such as Author, Title, Publisher, and so on. In figure 1, the contents of fields such as Author and Title are enclosed by quote delimiters (curly braces can also be used). The contents of the Series and Publisher fields however are not so delimited; these fields are filled by abbreviations which are expanded by `@STRING` definitions:

```
@STRING{OPAC = "Occasional Publications in Academic Computing"}
@STRING{SIL = "Summer Institute of Linguistics"}
```

In figure 1 the string `kew-mcconne190` in the first line is the entry key (similar to the notion of a record key in a database). The entry key is not a true field, and is not delimited by quotes or braces (nor is it an abbreviation). The entry key is used in a `LATEX` document to refer to this bibliographic entry.

`BIBTEX` takes as input a database file, applies a selected style, and produces a formatted output. This output file is in turn fed into `LATEX`, which attaches it to the main document and applies a final layer of styling. The point to notice here is that the database format is totally separated from the style of the printed bibliography.

While you can use a text editor to maintain files of bibliographic entries such as shown in figure 1, the chances of making errors are great. Just one mistyped field name or missing quote mark and `BIBTEX` will choke. Obviously the better solution is to use a front-end data entry program to insulate you from `BIBTEX` format. Several such programs have been developed for various computing platforms. `HyperBIBTEX` is a new entry into this market.

## Design goals

The following points summarize `HyperBIBTEX`'s most important design goals and how they were implemented.

- *HyperBIBTEX should be fully compatible with BIBTEX.*

`HyperBIBTEX` supports all standard `BIBTEX` entry types and fields and can exchange data with `BIBTEX` by means of importing and exporting `bib` files.

- *HyperBIBTEX should provide general database management facilities for*

\

*data entry, navigation, sorting, searching, and retrieval.*

\

HyperBIBTEX uses HyperCard's built-in database facilities augmented by various XCMDs.

- *A HyperBIBTEX stack should safeguard its own structural integrity.*

When a HyperBIBTEX stack is opened, it goes through a series of checks to ensure that the stack has not been corrupted.

- *HyperBIBTEX should assist the user to maintain the consistency, accuracy, and security of the bibliographic data.*

HyperBIBTEX maintains authority lists of repetitive data such as authors, journal names, publishers, and keywords. Fields can be filled in simply by choosing items from the appropriate authority list. Data that is manually typed into fields is automatically verified against the appropriate authority list. As for security, commands to cut or delete a card require confirmation, and the user is prompted to do a daily back-up of the stack. Also, exporting all the data from a stack is another form of back-up.

- *HyperBIBTEX should support a variety of hardware and software configurations.*

HyperBIBTEX supports 9 inch monitors, System 6 and 7, HyperCard versions 2 and 2.1, HyperCard Player, and non-English versions of HyperCard. There was no attempt to support WorldScript.

- *HyperBIBTEX should provide enough documentation and on-line help for anyone to use it with relative ease.*

HyperBIBTEX has an on-line Help stack and Balloon Help (under System 7) for buttons and fields.

The remainder of this article will describe HyperBIBTEX's features in detail.

## **General architecture**

A HyperBIBTEX installation minimally consists of three stacks: a main stack of bibliography cards, a Utilities stack, and a Help stack. The user can have as many bibliography stacks as desired, but only one copy each of the Utilities and Help stacks need be maintained on the user's system. HyperBIBTEX is supplied with an empty bibliography stack containing only several utility cards. To make a new bibliography stack, the user makes a copy of the empty stack. A new bibliography stack can be developed either by manually making new cards and

\  
typing in data or by

\

importing an existing `bib` file (containing entries of the type shown in figure 1). When you want to make a bibliography for a document, you simply export selected cards from the stack to a `bib` file and process it as usual with `BIBTEX`.

All of `BIBTEX`'s thirteen entry types (Article, Book, etc.) are supported as well as all required and optional fields for each entry type. Each entry type is implemented as a distinct background. A template card for each background is kept in the Utilities stack. When you make a new card for an entry type (background) not yet used in the stack, `HyperBIBTEX` copies the template card from the Utilities stack to the main stack. If the entry type (background) does exist, then the new card is created in the usual way. This scheme makes a `HyperBIBTEX` stack somewhat unusual compared to most `HyperCard` stacks since there is no fixed number of backgrounds in a stack.

A sample Book card is shown in figure 2. Required fields have field names in bold.

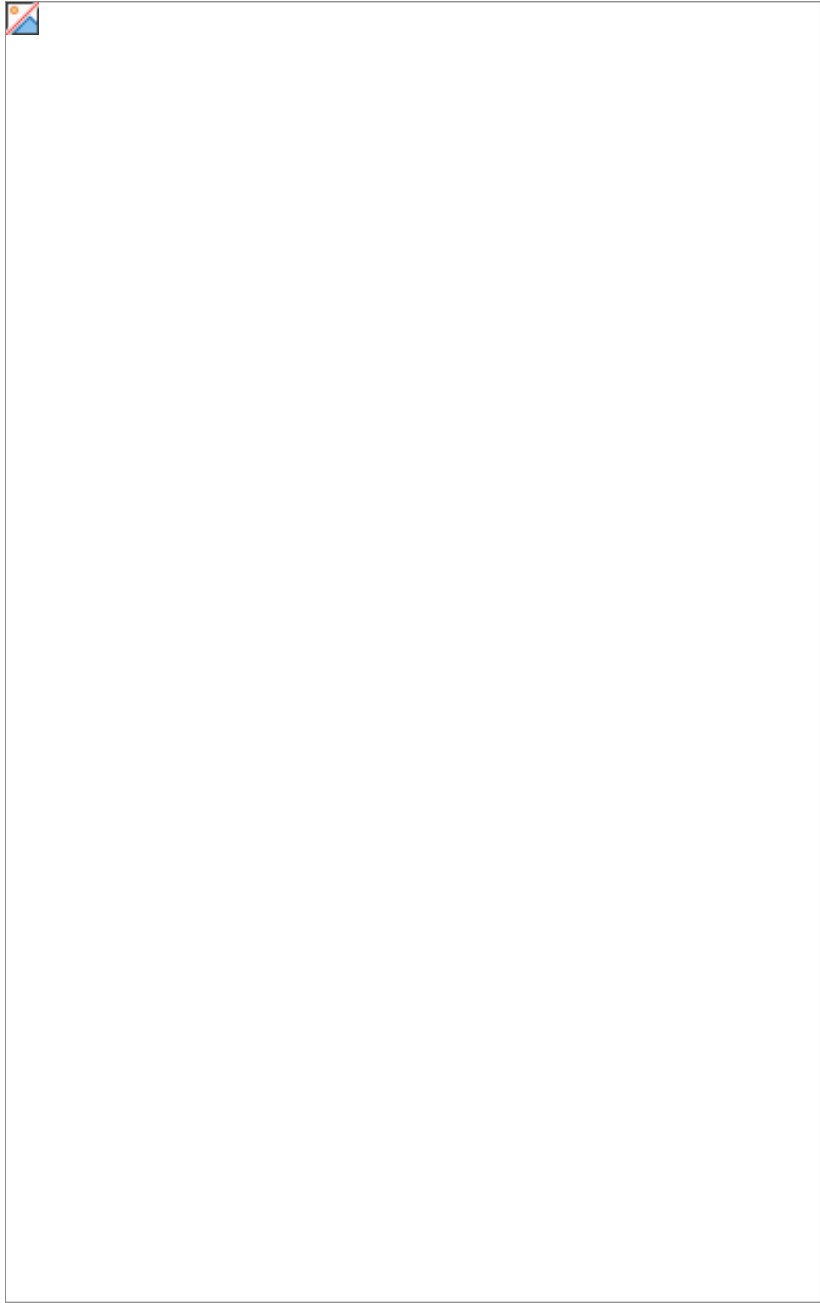
`HyperBIBTEX` includes several useful fields not normally supported by `BIBTEX`; these are ISBN number, Location, Keywords, and Annotation. The Annotation field, accessed with the Annotation button, is implemented as a scrollable, pop-up windoid that can be moved and resized on the screen (see figure 3).

Next to the Annotation button is the User Fields button, which also brings up a windoid into which can be typed user-defined fields coded directly in `BIBTEX` format (see figure 4). This permits the inclusion of fields that are supported only by customized `BIBTEX` styles or simply extra fields that the user wants to maintain. (Both the Annotation and User Fields windoids are created with Rinaldi's `Textoid XCMD`.)

`HyperBIBTEX` uses the standard `HyperCard` command interface of buttons and menus. Many commands also have command-key or function-key aliases. These key assignments can be customized by adjusting a couple handlers in the stack script.

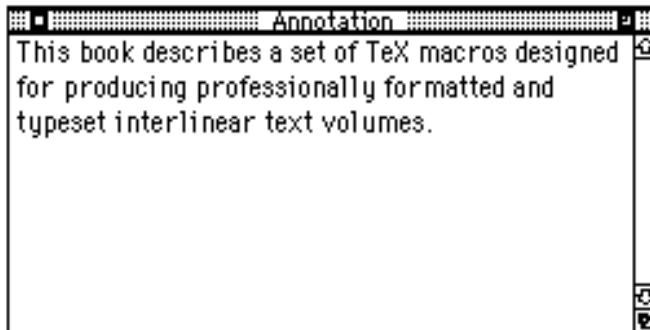
\

**Figure 2 A Book card**

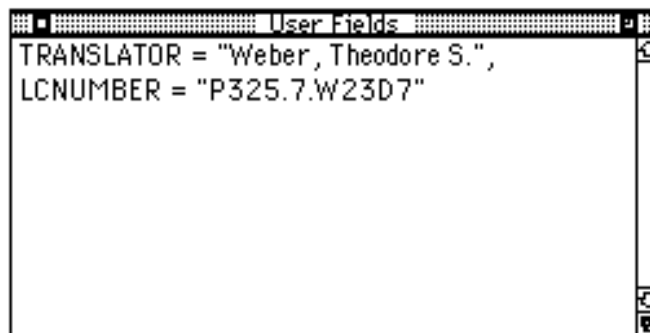




**Figure 3** The Annotation windoid



**Figure 4** The User Fields windoid



## Ensuring stack integrity

When HyperBIBTEX starts up, it performs a number of integrity checks on a bibliography stack.

- It verifies that the main stack and the Utilities stack have the same version number. (ResEdit is used to set the "vers" resource and Rinaldi's ResText XFCN to get the version number.)
- It verifies that only valid backgrounds are present in the stack.
- It verifies that no duplicate backgrounds are present. (Unfortunately, HyperCard will happily permit a stack to have two virtually identical backgrounds with the same name. The programmer must compare names with background IDs to discover duplicates.)
- It verifies that all the utility cards are present and in their proper locations.
- It verifies that all required fonts are available.

- It verifies that there is sufficient disk space; if there is not, then modification of the stack is disallowed.

Here are some other stack and data safeguards.

- A stack can become corrupted when the user pastes in a card copied from some other stack. HyperBIBTEX will permit you to paste a bibliography card cut/copied from another HyperBIBTEX stack, but will check the pasted card to be sure it both belongs to a valid background and does not create a duplicate background; if it does, then it is immediately deleted from the stack.
- The utility cards cannot be deleted or renamed.
- HyperCard normally permits you to interrupt an operation by pressing ⌘. (Command-period). But if this done during some operations, the stack could be left in an unstable state. HyperBIBTEX avoids this problem by setting the abort property to false at the start of most handlers and setting it back to true at the end.
- HyperCard commands that could modify the stack are disabled, such as creating, deleting, cutting, or pasting buttons and fields.
- HyperBIBTEX uses HyperCard version 2.1's superior error handling facility to ensure that if an unexpected error occurs during sensitive operations, the stack will be restored to a stable state (globals are restored to normal values, windoids are closed, open files are closed, and so on).
- When you cut or delete a card, you are asked to confirm the operation.
- When you close a stack, you are prompted to make a back-up copy if more than one day has passed since the last back-up.

One powerful HyperCard function has not been disabled: the Message Box. You can execute any HyperCard command from the Message Box. While you could do damage this way, I feel that the usefulness of the Message Box outweighs its dangers.

The user has full access to HyperBIBTEX's scripts. Since one can hardly modify a script accidentally, this is not much of a threat to a stack's security. In the spirit of HyperCard freeware, I prefer users to be able to examine the scripts and even modify them.

## Authority lists

HyperBIBTEX maintains four authority lists for bibliographic data, each on its own card in a large scrolling field. Three of the lists serve more than one field. The lists and the fields they serve are as follows:

Author/Editor

Journal/Series/Type

Publisher/Organization/Institution/School/HowPublished

Keywords

The authority lists are automatically updated when new data is entered into any of these fields or when the Verify Field command is used. However, a list can also be manually edited, normally to remove incorrect or obsolete entries. The list will be automatically cleaned up (extra white space removed) and sorted when you finish editing it.

## Data entry

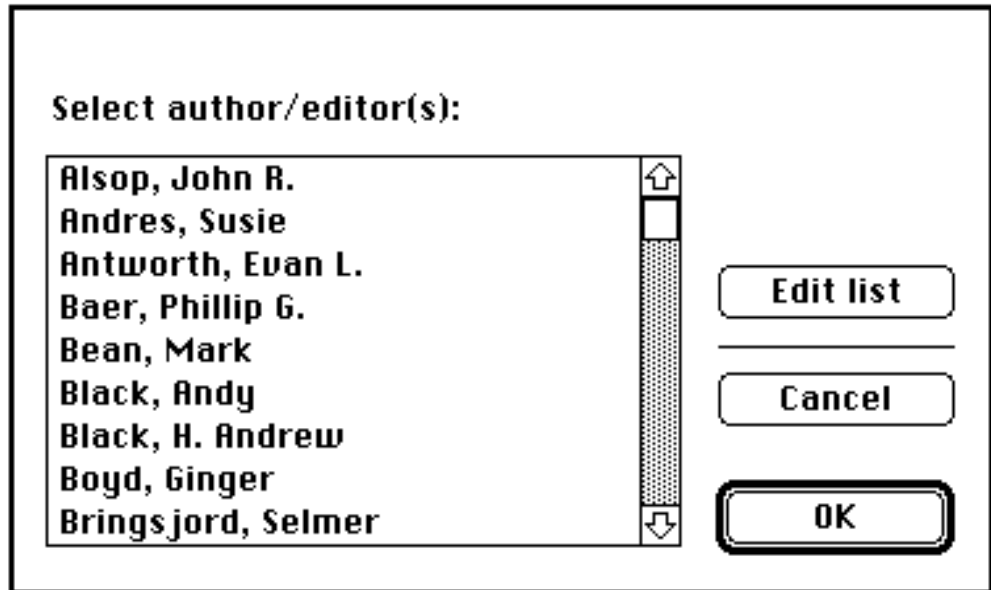
HyperBIBTEX offers several features to facilitate data entry and to ensure its accuracy and consistency.

- To make a new card, you choose an entry type from the New menu. Or, HyperCard's New Card command will make a new card of the same entry type as the current card.
- If you type data into a new card and then discover that it is the wrong entry type, you can select a different entry type from the Change menu.
- To move from one field to another on a card, use the Tab key (or Return and Enter keys in nonscrolling fields) or click in a field.
- When you finish typing data into a field, all extra spaces and blank lines are removed. (These functions are provided by a set of handlers from Apple called Valid Entry Package.)
- The Year and Month fields can be filled in by choosing from a pop-up list of years or month names. (These buttons use Dartmouth's PopList XFCN.)
- If you command-click on a word in the Title field, it is copied to the Keywords field.

- Authority lists are kept for fields with repetitive information, including Author, Journal, Publisher, and Keywords. To enter data in one of these fields, click on the button next to it and choose from the list shown in a dialog box (see figure 5). If an entry has

more than one author, you can select them from the list using multiple contiguous and noncontiguous selection; they are put into the Author field with “and” separating them. (The dialog box shown in figure 5 is created with Apple’s ShowList XFCN.)

**Figure 5** Dialog box to choose from the Author list



- When you type data into a field which has an authority list, the data is checked against the authority list. If it is not found in the list, a dialog box prompts you either to change the data you just typed (in case it is incorrect) or to add the new data to the appropriate authority list (see figure 6). Data verification can be turned off with an option on the Preferences card.

**Figure 6** Dialog box to verify data in Author field



- Abbreviations are supported. An abbreviation consists of a single word with a period prefixed to it. In figure 2, the Series field contains the abbreviation “.OPAC” and the Publisher field contains the abbreviation “.SIL”. The definition of an abbreviation is kept in the authority list for

\

the field. Abbreviations can be inserted in fields using the field buttons,  
and

new abbreviations are placed in the authority list after prompting the user for a definition. Abbreviations can also be used in fields which have no authority list, but no definitions of them can be supplied when the cards are exported.

- You can use Macintosh accented characters (such as *é*). These are automatically converted to T<sub>E</sub>X macros (such as `{\e}`) when the data is exported.
- All fields use Geneva font and plain style. Because choice of font and typestyle is done when a bibliography is formatted for printing, B<sub>I</sub>B<sub>T</sub>E<sub>X</sub> database files do not require (or permit) font or typestyle information. So neither does HyperB<sub>I</sub>B<sub>T</sub>E<sub>X</sub>. HyperCard's Font and Style menus are normally unavailable in a HyperB<sub>I</sub>B<sub>T</sub>E<sub>X</sub> stack, but if you do succeed in using a different font or style when typing data into a field, it will revert to Geneva font and plain style when you close the field.

**Figure 7** Dialog boxes to verify and define an abbreviation



## Navigation

A stack can be navigated using the usual HyperCard commands. The Navigator palette is also available. At the bottom of each card are two arrow buttons that go to the previous or next card. If you hold down the option or command key when clicking these buttons, they go to the previous or next marked card in the stack.

One special navigation feature is the Index windoid (see figure 8). The Index windoid displays a list of entry keys of either all the cards in the stack or just the

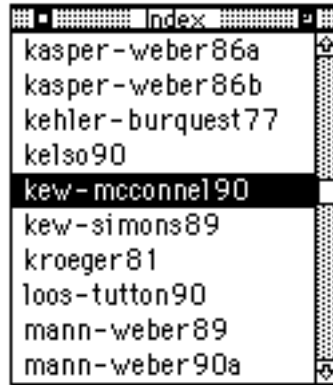
\  
marked cards. To go directly to a card, you simply



\

double-click on the card's key in the Index windoid. (The Index windoid is created with Rinaldi's Listoid XCMD.)

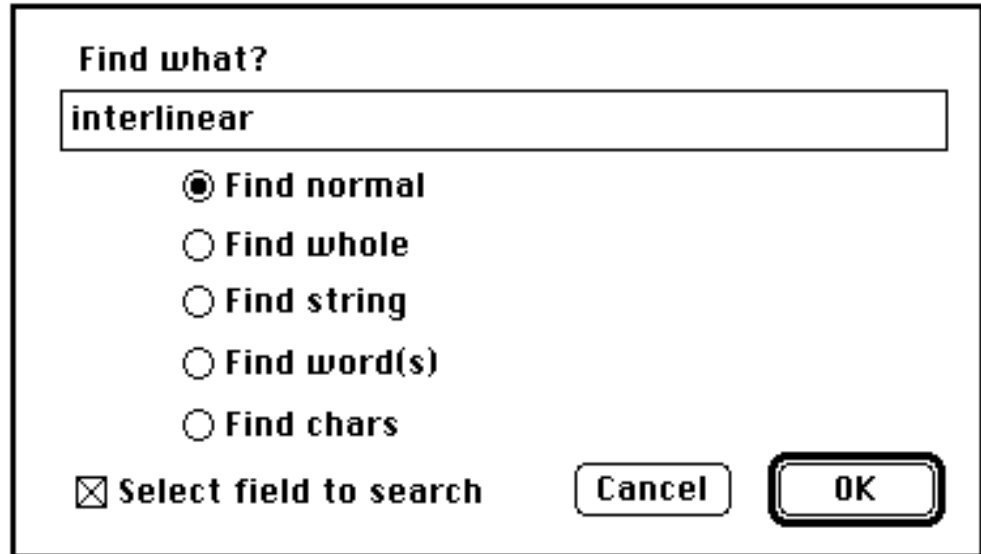
**Figure 8** The Index windoid



## Searching

HyperBIBTEX uses HyperCard's built-in Find function with an improved Find dialog box, shown in figure 9 (created with Dartmouth's ModalDialog XFCN). The Find dialog box permits the user to type in a search string, to choose the method of finding such as normal or whole (these are all built into HyperCard), and an option to limit the search to a selected field. If the "Select field to search" box is checked, then another dialog appears with a list of fields to select from. Unfortunately, HyperBIBTEX has no Find and Replace function. This is because HyperCard itself has none, and I have not found a suitable (and free) Find and Replace XCMD.

Figure 9 The Find dialog box



## Marking and unmarking cards

HyperBIBTEX uses HyperCard's built-in facility for marking cards. You can mark or unmark any bibliography card by checking or unchecking its Marked check box. You can also mark or unmark groups of cards that are selected using the Find function. For example, you could mark all cards whose Keywords field contains "text analysis" and then navigate among just those cards either by option-clicking the arrow buttons or by building an Index of just the marked cards. You can also export just the marked cards or delete marked cards. By doing successive mark and unmark operations, you can refine your selection. For example, to select all the cards in a stack except those published by SIL, you would first mark all cards (with a single command) and then unmark just those cards whose Publisher field contains "SIL".

You can also mark/unmark cards by directly running the HyperTalk mark and unmark commands from the Message Box. A very useful form of these commands is this:

```
mark/unmark cards where expression
```

*Expression* can be any valid HyperTalk expression. For example:

```
mark cards where field "year" > 1980
```

```
mark cards where "linguistics" is not in field "keywords"
```

(Obviously it would be preferable to have a dialog box interface for doing this.)

## Sorting

HyperBIBTEX uses HyperCard's built-in facility for sorting the stack. HyperBIBTEX's Sort command uses the International option, which causes all accented variants of a letter to sort together (for example, *a*, *á*, *à*, and *ä*). You can choose to sort the cards on any significant field such as Key, Entry Type, Author, Title, Year, Journal, and Publisher (fields such as Pages and Volume do not permit sorting). Note however that a given field may not occur on every card; for example, the Journal field occurs only on Article cards. If you sort the stack on the Journal field, then HyperCard will place all cards without that field first in the stack.

You can achieve the effect of sorting on a primary and secondary field simply by sorting on those fields successively in reverse order. For example, if you want the stack sorted by author but cards with the same author to be subsorted by date, then first sort on the Year field and then sort again on the Author field.

\

Sorting a HyperBIBTEX stack proved to be a challenge in HyperCard design. In addition to the bibliography cards, every stack contains a Title card, four authority list cards, and a Preferences card. The Title card must always be the first card of the stack, the Preferences card must be last, and the authority list cards must immediately precede the Preferences card in a fixed order. It is not possible to exclude certain cards or backgrounds when sorting a stack. Thus there has to be a mechanism for ensuring that the utility cards don't get misplaced when the stack is sorted. The Title card is easy: it has no sortable fields on it so it always ends up first. The authority list and Preferences cards, however, must sort last. To ensure this, each of these cards has all sortable fields on it, hidden from view. In each field is a string of the form *XX#*, where *X* is the last character of the sorting order and *#* is the relative number of the five cards. By default *X* is set to the character corresponding to ASCII 255, since when you "sort international" with the English (or at least American) version of HyperCard, that is the last character in the sorting order. Unfortunately, this is not the case with some non-English versions of HyperCard; ASCII 255 might be occupied by an alphabetic character right in the middle of the sorting order, which would cause the utility cards to be misplaced when the stack is sorted. Until someone can offer a better solution, here is mine. There is a button on the Preferences card named "Fix sort order". When you click on it, HyperBIBTEX build a list of characters up to ASCII 255, sorts it, picks out the last character, and substitutes it for *X* in all the fields of all the utility cards. This strategy should work no matter what language version of HyperCard you use. But if it fails, HyperBIBTEX checks the location of the utility cards after sorting and on start-up, and if they are out of order they are cut and pasted into their proper place. (This is called the belt-and-suspenders philosophy of programming!)

## Data verification

HyperBIBTEX has several commands that go through a stack card-by-card, checking the consistency and accuracy of the bibliographic data.

- The Check Keys command looks at the entry Key for each card and alerts the user if the Key is empty or if the same Key is used for more than one card in the stack.
- The Find Duplicates command looks at each pair of consecutive cards in the stack, comparing their entry Keys and the field that was last used for sorting. If they partially share identical data, then the user is alerted that the cards might be duplicates.
- The Verify Field command checks each card to see if the data in a

\

specified field is found in the appropriate authority list. The command can be run either in batch mode or confirm mode. In

batch mode, field data not found in the authority list is automatically added to the list. This method is good to use if you have imported a `bib` file and want to automatically build the authority lists. In confirm mode, you are prompted to decide what to do when field data is not found in the authority list. This is done with the same dialog as is shown in figure 6. One refinement of this process is that if you click the Continue button for, say, a certain author's name which for some reason you don't want added to the authority list, you will not be asked to consider that same name if it occurs again on a subsequent card.

## Importing and exporting

HyperBIBTEX exchanges data with BIBTEX by importing and exporting `bib` files. Because most BIBTEX users already have `bib` files that they have developed by hand, it is vitally important for HyperBIBTEX to be able to import `bib` files. From a programmer's point of view, this was perhaps the trickiest part of writing HyperBIBTEX. Even though `bib` files have a well-defined syntax, they permit a considerable amount of etic variation. HyperTalk is not powerful enough to write a really robust routine for reading and parsing files of data. Also, importing runs very slowly, though this is mostly due to the time it takes to make new cards. Ultimately, the better solution would be to write an importing routine in C or Pascal and use it as an XCMD. In spite of these disclaimers, the only major deficiency in HyperBIBTEX's importing facility is that it does not support concatenated abbreviations (an advanced feature added to BIBTEX version 0.99b, see Patashnik 1988).

HyperBIBTEX's importing function offers these features:

- Either quotes or curly braces are recognized as field delimiters.
- Abbreviations (except concatenated abbreviations) are supported. `@STRING` definitions are imported and stored in an authority list.
- Non-standard (user-defined) fields can be imported by listing their field names in a box on the Preferences card.
- TEX macros for characters with diacritics (such as `{\e}`) can be automatically translated into Macintosh accented characters (such as `é`).
- When imported data is inserted into a field on a card, all extra white space is removed.

HyperBIBTEX's exporting function offers these features:

- The user can export either all the cards in a stack or just the marked cards. Cards can be marked either manually or by using the Mark and Unmark commands.
- You can choose which non-BIBTEX fields to export (these are ISBN, Location, Keywords, Annotation, and User Fields).
- Abbreviations used in fields can be exported as abbreviations along with their @STRING definitions or they can be expanded into their full form before they are exported.
- Either quotes or curly braces can be chosen as field delimiters.
- All entries of the types InBook, InCollection, and InProceedings can be exported first. BIBTEX requires this if the entries contain cross-references to other entries.
- Macintosh accented characters (such as é) can be automatically translated into TEX macros (such as {\'e}).

## How to get HyperBIBTEX

HyperBIBTEX version 0.9.8 was released on 27 September 1993. (This version number is in keeping with BIBTEX, which is still version 0.99c!) If you are presently using an older version of HyperBIBTEX, I strongly recommend that you upgrade to version 0.9.8. HyperBIBTEX is freeware and is available by anonymous FTP from:

```
mac.archive.umich.edu
/mac/hypercard/organization/hyperbibtex0.98.sit.hqx
```

Macintosh versions of TEX, LATEX, and BIBTEX are available by anonymous FTP from [midway.uchicago.edu](http://midway.uchicago.edu).

You can order HyperBIBTEX on diskette (\$5 to addresses in the U.S., \$7 elsewhere) from:

```
International Academic Bookstore
7500 W. Camp Wisdom Road
Dallas, TX 75236

phone: 214/709-2404
fax: 214/709-2433
e-mail: academic.books@sil.org
```

\

Send all correspondence about HyperBIBTEX to:

Evan Antworth  
Academic Computing Department  
Summer Institute of Linguistics



\  
7500 W. Camp Wisdom Road  
Dallas, TX 75236  
phone: 214/709-3346  
fax: 214/709-2433  
e-mail: evan.antworth@sil.org

## References

- Kew, Jonathan and Stephen R. McConnel. 1991. *Formatting interlinear text*. Occasional Publications in Academic Computing No. 17. Dallas, TX: Summer Institute of Linguistics.
- Knuth, Donald E. 1986. *The T<sub>E</sub>Xbook*. Reading, MA: Addison Wesley Publishing Company.
- Lamport, Leslie. 1986. *L<sub>A</sub>T<sub>E</sub>X: A Document Preparation System*. Reading, MA: Addison Wesley Publishing Company.
- Patashnik,, Oren. 1988. *BIBT<sub>E</sub>Xing*. Unpublished manuscript.