Figure D.1: Registration Form

Please answer all 5 of the following questions:

1. Your name: ____________________

2. Your mailing address: ____________________
____
____

3. Your e-mail address(es): ____________________
____
____

4. Which operating system are you using: MS-DOS ☐ OSK ☐

5. Name of BBS or information serivce you downloaded the *Home Librarian* from: ____________________

The remaining questions are optional. You don't have to answer any of the, but please try to answer as many as you feel like. These answers will help me to make improvements in the *Home Librarian* package.

6. Percentages of items in your library are:
Printed materials (books and magazines):
0 to 20 ☐ 21 to 40 ☐ 41 to 60 ☐ 61 to 80 ☐ 81 to 100 ☐
Audio materials (albums, CDs, DATs, audio cassettes, 8-tracks):
0 to 20 ☐ 21 to 40 ☐ 41 to 60 ☐ 61 to 80 ☐ 81 to 100 ☐
Video materials (VHS, Beta, 8mm, laser disks):
0 to 20 ☐ 21 to 40 ☐ 41 to 60 ☐ 61 to 80 ☐ 81 to 100 ☐

7. What do you *like* about the user interface of `EditLibr` and `Librarian`? ____________________

8. What do you *not like* about the user interface of `EditLibr` and `Librarian`? ____________________

---

lems you might have. It also means I will really listen to your suggestions and comments.

C.3 If You Decide Not To Register.

This is ok. You are stuck with the plain ASCII text version of the manual that came with the package and are entirely on your own if you have problems or questions. I will also "turn a deaf ear" to any suggestion or comments.

C.4 Why Not Simply Make It a Commercial Package?

Producing a slick, shrink-wrapped commercial package requires a certain amount of investment in time and resources. It would make the package more costly to the user (much more than the registration fee) and would put me in debt. If it did not sell I could be stuck with a large debt. It is posible, if this package is extremly popular, that a commercial version might be produced. As a shareware package, the *Home Librarian* package can be distributed widely at low cost. A lot of people can get a chance to try it out without having to pay. If the package proves to be usefull, people can choose to pay the modest registration fee and get a nice manual and customer support.

demanded for a copy of this package[1].

B.2.3 No Support.

Unless the package is registered, no customer support of any sort is provided.

B.3 No Warranty.

BECAUSE THE *Home Librarian* PACKAGE IS LICENSED FREE OF CHARGE, THERE IS NO WARRENTY FOR THE PACKAGE, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PACKAGE "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PACKAGE IS WITH YOU, THE USER. SHOULD THE PACKAGE PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY REDISTRIBUTE THE PACKAGE AS PERMITED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF USE OF THE USE OR INABILITY TO USE THE PACKAGE (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED

[1] A modest media or transmission charge or fee is allowed. This charge or fee should not exceed the cost of the media or the cost of the transmission.

A.2 Invoking Ascii2Libr.

`Ascii2Libr` is invoked from the command line like this:

```
Ascii2Libr -outfile <filename>
```

or like this to specify a new minimum page count:

```
Ascii2Libr -outfile <filename> -minpages <minpages>
```

Where `<filename>` is the name of the file to be created and `<minpages>` is the (optional) minimum number of index pages to allocate[1].

The input to `Ascii2Libr` is from the standard input stream.

A.3 When You Need to Use These Programs.

These programs convert a binary format card catalog data base file to an easily transferable text file. All modern computer systems support plain ASCII text files. The binary format contains processor-dependent binary representations of the values use.

These two programs can also be used to "compress" a data base file. `EditLibr` makes no attempt to re-cycle data records. This is due in part because these records are variable length. Maintaining a table of "deleted" records, along with their sizes would involve considerable overhead. It is simpler to waste some disk space during the editing process and reclaim it later with `Libr2Ascii` and `Ascii2Libr`.

[1]If `<minpages>` is not specified, the number of index pages to pre-allocate is taken from the input stream and is just enough to hold the indices.

Table 5.2: `PrintLabels` Template File Field Replacement Formats.

Field	Replaced by
`%%`	`%`
`%id`	The id string.
`%type`	The type.
`%author`	The author.
`%title`	The title.
`%publisher`	The publisher.
`%city`	The city.
`%volume`	The volume.
`%year`	The year.
`%description`	The description.

Where `<filename>` is the file to print labels from, `<tfile>` is a template file and `<options>` consist of zero or more of the options listed in Table 5.1.

The expression can contain the operators listed in Table 4.2. Each relational expression needs to specify a field name and a value. The value must be of the correct type as shown in Table 4.3.

5.2 The Template File.

The template file is a text file that is copied to the standard output stream for each card record processed. The file is copied literally, except when a `%` character is encounted. This character indicates a field replacement as shown in Table 5.2.

5.3 `PrintLabels` Output.

The output of `PrintLabels` is to the standard output stream. This can be redirected with the output redirection command line character (`>`) or piped with the pipe character (`!` under OSK or `|` under MS-DOS).

Table 4.1: `PrintCards` options

Syntax	Description
`-largep true|false`	Selects between large (5x8) or small (3x5) cards. If `true` large cards are generated and if `false` small cards are generated.
`-by id|title|author|subject`	Selects the type of card to generate: shelf list cards (`id`), title cards (`title`), author cards (`author`), or subject cards (`subject`).
`-only <expression>`	Used to select a subset of cards to generate. `<expression>` is a relational expression comparing fields to selected values.

Table 4.2: Expression operators

Operator	Meaning
`<`	Less than
`>`	Greater than
`=`	Equal to
`!=`	Not Equal to
`<=`	Less than or Equal to
`>=`	Greater than or Equal to
`&`	And
`|`	Or
`!`	Not
`(...)`	Expression grouping
`"..."`	String constant

Figure 3.1: Main command menu of `Librarian`.

```
                 Main Command Menu

Quit
List Cards
List Titles
List Authors
List Subjects
```

Table 2.1: Card Editor Key Bindings.

Special Key	Generic Key	Meaning
`Home`	`Esc`	Exit
← (left arrow)	`Ctrl-B`	Previous field
→ (right arrow)	`Ctrl-F`	Next field
↑ (up arrow)	`Ctrl-P`	Previous field
↓ (down arrow)	`Ctrl-N`	Next field
	`Tab` or `Ctrl-I`	Next field
	`i` or `I`	Modify current field
	`Ctrl-L`	Repaint screen
	`?`	Display help screen
	`Ctrl-S`	Save card
	`h` or `H`	Give help for the current field

2.2.2 Card Editor.

When the card editor is invoked, it checks to see if the card to be edited exists in the data base or not. If the card does not exist, you are asked if you want to create a new card. The card editor then puts up a screen as in Figure 2.2. The card editor defines the key bindings shown in Table 2.1 The `i` (or `I`) key is used to modify fields. The `id` field cannot be modified. When the displayed card is different from the stored card, an asterisk (`*`) is displayed in the uppper left corner of the screen. Most of the fields are updated by simply typing the `i` (or `I`) key and then re-typing the field and pressing the `Return` or `Enter` key. Two fields are different. The `description` and `subject` fields are modified using an external text editor[9]. Entering the `i` (or `I`) key spawns the external editor with a temp file containing the field to be modified.

[7] If only one subject matches, it is automatically selected

[8] If the subject only refers to a single card, that card is placed in the card editor directly

[9] Defined by the `EDITOR` environment variable or `umacs` by default under OSK and `???` by default under MS-DOS.

The `Delete by Subject` Menu Item.

The `Delete by Subject` menu item is used to delete some or all cards for items with a selected subject. You are prompted for an subject and then the card ids for that subject are listed one by one and you are asked if you want to delete each card.

The `List Cards` Menu Item.

The `List Cards` menu item lists cards with their titles which have a common selected prefix for their id strings. You are prompted for a search prefix and those cards with a matching prefix string in their id string are listed. You have the option of editing a selected card.

The `List Titles` Menu Item.

The `List Titles` menu item lists titles with a common selected prefix string. You are prompted for a search prefix and those titles with a matching prefix string are listed. You have the option of selecting one of the listed titles[3], in which case the ids for the selected title are listed (much as in the `List Cards` menu item)[4].

The `List Authors` Menu Item.

The `List Authors` menu item lists authors with a common selected prefix string. You are prompted for a search prefix and those authors with a matching prefix string are listed. You have the option of selecting one of the listed authors[5], in which case the ids for the selected author are listed (much as in the `List Cards` menu item)[6].

---

[3] If only one title matches, it is automatically selected

[4] If the title only refers to a single card, that card is placed in the card editor directly

[5] If only one author matches, it is automatically selected

[6] If the author only refers to a single card, that card is placed in the card editor directly

Figure 2.1: Main command menu of `EditLibr`.

```
                 Main Command Menu

Quit
Edit a Card
Delete a Card
Delete by Author
Delete by Title
Delete by Subject
List Cards
List Titles
List Authors
List Subjects
Spawn Shell
```

a new file. Each index page stores upto 21 index keys. Remember, there are four indexes. If while editing your data base, you run out of pre-allocated index pages, more will be allocated. The difference will be that new index pages are allocated from the end of the file on an as needed basis. If pages get allocated this way the index table will be fragmented. Pre-allocating the pages puts them all new at the beginning of the file, for faster access[2].

input editing disabled), much like a screen-based text editor such as Emacs.

The input handling code expects a terminal that has arrow keys on its keyboard, but will work with terminals whose keyboards do not have arrow keys, by accepting "generic" control characters instead. These "generic" control characters were taken from Emacs's positioning bindings. The common (used in all screens, except as noted) key bindings are listed in Table 1.4.

1.3 How This Manual is Organized

This manual is organized by functions – each chapter describes the program(s) that perform a particular function.

1.3.1 Creating and Modifing a Data Base.

A data base file can be created and modified with the EditLibr program, which is described in chapter 2. This program creates and modifies card catalog data bases.

1.3.2 Searching a Data Base.

The Librarian program, which is described in chapter 3, is a program meant to be used to search card catalog data bases.

1.3.3 Getting Hard Copy of a Data Base.

A "hard copy" of the data base can be created with the PrintCards program which is described in chapter 4. This program is used to generate either 3x5 or 5x8 cards. These cards are much like old-fashioned paper card catalogs.

[1] If you are using OSK, you need to set the TERM environment variable to the terminal type. Under MS-DOS, the terminal type is presumed to be an ANSI terminal and you need to be running ANSI.SYS or something simular.

Figure 1.1: Structure of a “Card Catalog Data Base”.

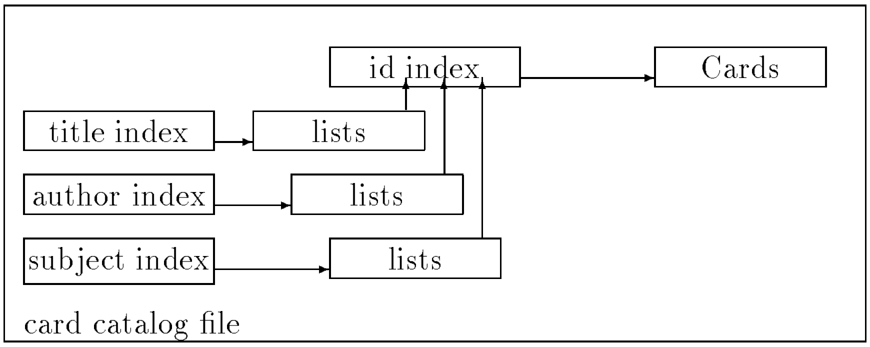

Table 1.2: Fields in a “card”.

Name	Description
`id`	A unique identifier.
`type`	The type of item the card describes.
`author`	For the author (or artist).
`title`	For the title.
`publisher`	For the name of the publisher.
`city`	For the city where the item was published.
`description`	For a long description.
`volume`	For the volume number.
`year`	For the year published.

to the cards *indirectly*, as shown in Figure 1.1.

1.1.2 What is a “card”?

A card has a number of fields as shown in Table 1.2. The `id` field could be an ISBN number, a LC call number, a Deuy-decimal call number, or any other unique identification code. The `type` field must be one of the types listed in Table 1.3. The `author` and `title` fields are used for the author and title cross-references.

have.

I'd like to thank my friend Ephraim St. George Robbins for helping me with this manual.

Robert Heller
Deepwoods Software
Wendell, MA, USA

This documentation was prepared with LaTeX.

This document describes version 1.0β of the *Home Librarian* package.

The author, Robert Heller, may be contacted electronically (E-Mail) via the following:

FidoNet 1:321/153, Locks Hill BBS.

CompuServe 71450,3432

BIX locks.hill.bbs

InterNet heller@cs.umass.edu