# Home Librarian
# A Computerized Library Card Catalog System
# User Manual

Robert Heller
Deepwoods Software
Wendell, MA, USA

May 27, 1998

This documentation was prepared with LaTeX.

This document describes version 2.0 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.

**InterNet** heller@deepsoft.com

Web site URL: `http://mall.netmar.com/mall/shops/heller/`.

# Contents

# List of Figures

# List of Tables

# Changes From Version 1.1

I have made a number of changes from version 1.1. These include:

- Added support for up to ten user defined card types.
- Added 2 fields for item location information. These two fields are a "location type" field, a coded field much like the card type field, which includes support for up to ten user defined location types, and a text field for the "location detail". These two fields all for recording the exact location of items in the database.
- Added a "category type" field. This allows for up to 256 user defined category types.
- A completely new GUI that is common across all platforms, UNIX, MS-Windows, and MacOS. This interface is colorful and easy to use and combines all functions into a single GUI application. The package no longer consists of 6 command-line oriented programs.

## User Defined Types

There is now support for upto ten user defined item types. No longer must all the odds and ends in your library be collected into the "Other" catch all type.

## Location Types

I have added support for keeping track of *where* your library items are. From which shelf to who has borrowed the item. There are also upto ten user defined location types.

## Item Categories

I have added support for upto 256 Categories – you can now place each item in one of 256 Categories to make searching easier.

## New GUI

Version 2.0 brings a brand new GUI interface. The original set of 6 command-line and “glass tty” GUI programs have been replaced with a single, cross-platform, Tcl/Tk GUI interface application. All of the functions provided by the original six programs is incorporated into this one application.

# Preface

The *Home Librarian* system maintains a database of "cards", which describe books, magazines, video and audio tapes, CDs, record albums, laser disks, and other similar items. The data is indexed by a unique ID for each "card", a title index, an author index, and a subject index. Each card contains fields for a title, an author, a publisher, a year, a volume number, a description, and an item type.

The package consists of two parts: a basic class library, which can be used as the basis for any sort of user-written "front end" or utility program, with a loadable shared library meant to loaded into a stock Tcl/Tk 8.0 shell[1]. The second part is a set of Tcl/Tk script files that implement a GUI for accessing and maintaining card catalog database files.

I had two main goals in mind when I originally set out to write the *Home Librarian* package. First I wanted to learn `C++` by writing some "real" programs, not just the exercises in the book, and secondly, I wanted to write a package to help me deal with my growing library of science fiction books, video tapes, and audio cassettes, which I estimate at about 1,000 items. The first goal has pretty much been reached, in that I believe I have a good working feel for `C++`. My second goal is still in progress[2]. I also wanted to experiment with writing a shareware package, to see if it is a viable source of income and not too much hassle, that is, to see if the cost/benefit tradeoff is reasonable. Things have gone beyond these original goals, although I still have not put all of my library into a database yet (only up to the C's), I have extended the database and by popular demand I have created a MS-Windows version[3].

[1]The package also compiles as a loadable shared library that can be loaded into Tclsh 7.6 or Wish 4.2.

[2]As of this writing, I have not completed putting everything into the database.

[3]With the help of some folks in Kiev – Val Stone and ICN, Ltd.

If you have *any* comments about this package, please let me know. My electronic mail addresses are listed on the back side of the title page. My postal address is 51 Locke Hill Road, Wendell, MA 01379, USA. I would be very interested in any comments users of the *Home Librarian* package might have.

Robert Heller
Deepwoods Software
Wendell, MA, USA
April, 1998

# Acknowledgments

There are a number of people who have done one thing or another that have directly or indirectly helped make this program possible. These people include Linus Torvalds, for creating Linux, so I would have a proper operating system to get this system working under, John K. Osterhout for creating Tcl/Tk, so I would have a cross-platform and easy to use GUI system available, and Leslie Lamport for creating LaTeX so I would have a type setting system available to write the documentation with. I would also like to thank Bjarme Stroustrup for inventing C++ and Sharam Hekmatpour for the nice little book I used to learn C++ (*C++ A Guide for C Programmers*).

More directly, I would like to thank Bruce Wilson for the artwork used for the librarian and the card drawer animation, my brother-in-law Chris Schepp for the artwork I use for my logo (scanned off my lettethead), and Val Stone and the crew at ICN, Ltd. of Kiev in the Ukraine, for producing the MS-Windows port of the Home Librarian package.

And last, but not least I'd like to thank the Pepsicola Company for inventing Mountain Dew, which helped me work late into many nights getting this system working.

# Chapter 1

# Introduction

## 1.1 Basic Terminology

The *Home Librarian* package works over files that are "Card Catalog Data Bases" which contain a number of "cards".

### 1.1.1 What is a Card Catalog Data Base?

A Card Catalog Data Base file contains a collection of "cards" that describe items in your library, such as books, magazines, records, and tapes. These cards are indexed with four indices as shown in Table 1.1. Each index uses string of up to 35 characters as keys. The keys are always compared in a case-insensitive fashion. That is, `"The Door Into Summer"` is the same as `"THE DOOR INTO SUMMER"`. Internally, the keys are converted to all uppercase letters. The `id` index indexes to the card records directly. The other three indices index to lists of `ids`, and thus index to the cards *indirectly*, as shown in Figure 1.1.

| Name | Description |
|---|---|
| `id` | This index is by a unique identifier. Each card has its own unique identifier. |
| `title` | This index is the title cross-reference. |
| `author` | This index is the author cross-reference. |
| `subject` | This index is a subject index. |

Table 1.1: Indices in a "Card Catalog Data Base".

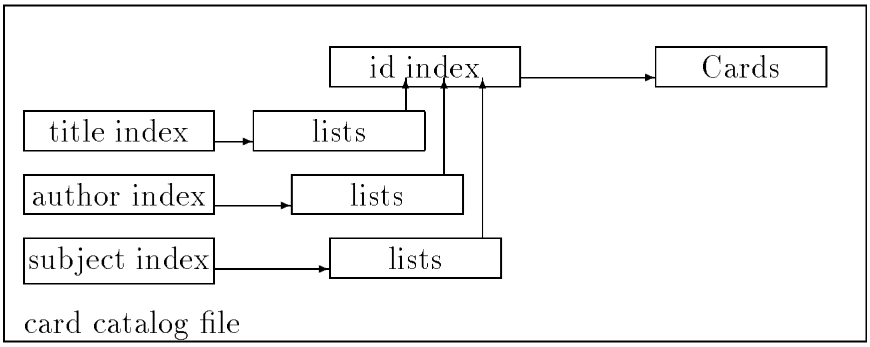


Figure 1.1: Structure of a "Card Catalog Data Base".

### 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 Dewey-decimal call number, or any other unique identification code. The `type` field may be one of the types listed in Table 1.3 or one of ten user-defined types. The `ltype` field may be one of the types listed in Table 1.4 or one of ten user-defined types. The `catcode` field is one of 256 category codes.

## 1.2 How This Manual is Organized

Chapter 2 describes the features of the main GUI window. Chapter 3 describes searching a database. Chapter 4 describes creating and updating a database. Chapter 5 describes printing operations. Chapter 6 describes importing and exporting a database file from or to plain ASCII text files[1]. Finally, Chapter 7 describes additional features and functions.

Appendix A provides the license and the copyright *Home Librarian* is distributed under.

[1] Database files are binary files with native data objects and cannot be transfered to disimular computer systems. ASCII text files are a platform independent format that can be easily exchanged across differing computer systems.

| Name | Description |
|---|---|
| id | A unique identifier. |
| type | The type of item the card describes. |
| ltype | The location 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. |
| locdetail | The location detail of item the card describes. |
| volume | For the volume number. |
| year | For the year published. |
| catcode | Category code of item the card describes. |

Table 1.2: Fields in a "card".

| Name | Description |
|---|---|
| Book | For books. |
| Magazine | For magazines and periodicals. |
| CD | For compact disks. |
| AudioCassette | For audio cassettes. |
| Album | For record albums. |
| LaserDisk | For laser (video) disks. |
| VHSVideo | For VHS format video tape recordings. |
| BetaVideo | For Beta format video tape recordings. |
| EightMM | For 8mm video tape recordings. |
| EightTrack | For 8-track audio recordings. |
| DAT | For 4mm digital audio tapes. |
| Other | For anything else. |

Table 1.3: Predefined types for a "card".

| Name | Description |
|---|---|
| `OnShelf` | Item is on a shelf. |
| `OnLoan` | Item is out on loan. |
| `OnOrder` | Item is on order. |
| `Destroyed` | The item has been destoryed. Record is for informational purposes. |
| `InStorage` | The item is in storage. |
| `Unknown` | The item's location is unknown or unspecified. |

Table 1.4: Predefined location types for a "card".

# Chapter 2

# Main GUI Window

The main window of the *Home Librarian* application, as shown in Figure 2.2, has a progress and status section at the top of the GUI[1], consisting of a "face" area and a progress and status area. The face area holds either a forward facing librarian image, when the GUI is idle and waiting for user input or a right profile when the application is working and not accepting input from the user. The status and progress either holds the message "How May I Help You?" or some sort of status message, usually detailing the number of matches found when searching, when the application is idle. When the application is working an animation of a card catalog drawer being opened and flipped through is displayed, as shown in Figure 2.1.

Below the progress and status section is the main interaction area. Initialially, this contains four big buttons giving access to the major functions of the application. At the bottom is a small space where the name of the currently open file is displayed, along with its open mode.

[1]Below the attached menu bar on UNIX and MS-Windows.

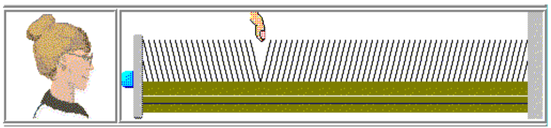

Figure 2.1: Face (state) with progress animation.

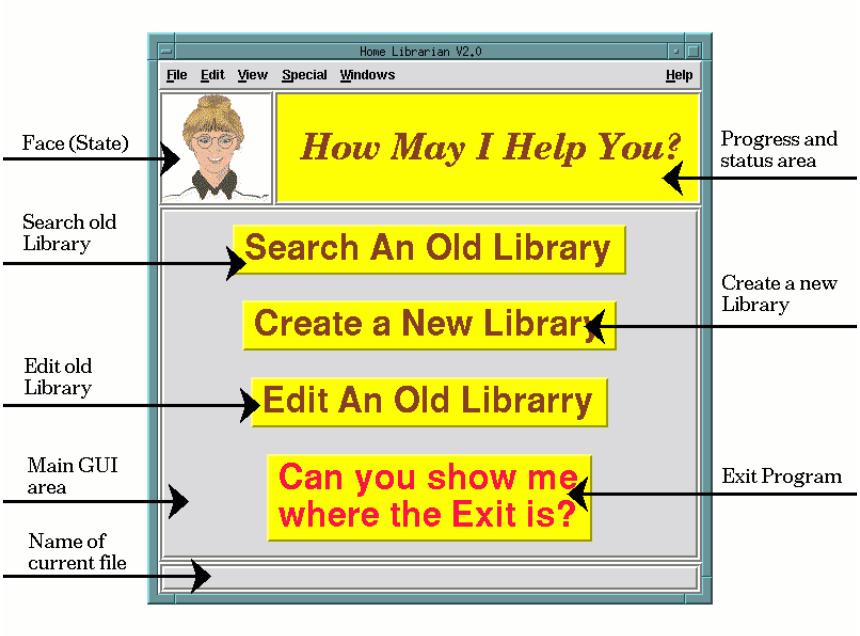


Figure 2.2: Initial (idle) view of the main GUI window.

## 2.1 Buttons on the initial (idle) Main GUI Window

There are two main functions supplied by the application: searching a card catalog database and editing a card catalog database. The top most button in the initial (and idle) state of the main GUI selects the searching function. The next two buttons select the editing function, either creating a new, empty database file or opening an existing database file. The bottom button exits the application. All of these operations are available from the `File` menu item as well, along with some secondary operations.

## 2.2 The Search function Main GUI Window

The Search function version of the Main GUI Window is shown in Figure 2.3. This version of the main window has a place for entering a search prefix string, a set of search index selection buttons, a search results list box, a set of search results disposition buttons, a notepad (text) area for making notes, and a

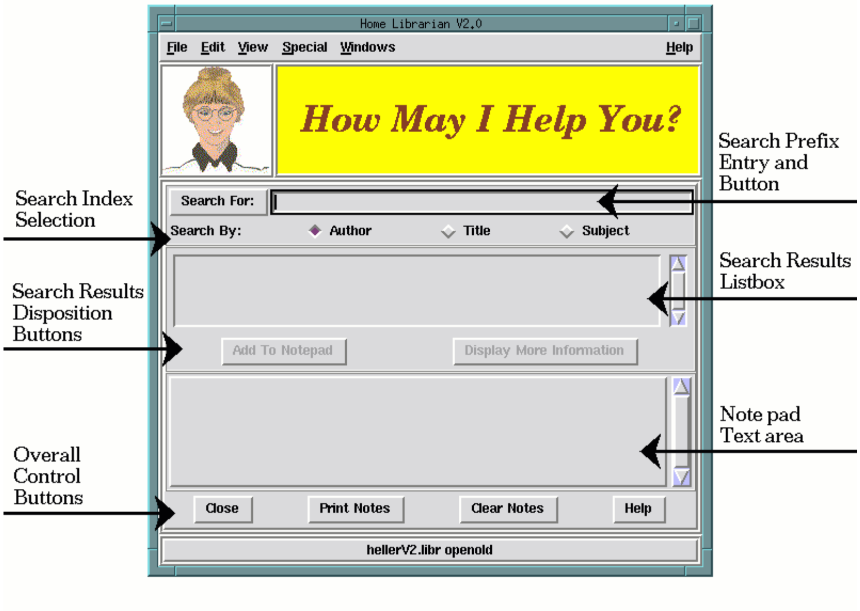


Figure 2.3: The Search function Main GUI Window

set of overall control buttons. Searching a database is described in detail in Chapter 3.

## 2.3 The Edit function Main GUI Window

The Edit function version of the Main GUI Window is shown in Figure 2.4. This version of the main window has a place for entering the id of a card to edit, searching for card(s) to edit, searching for card(s) to delete, and a set of overall control buttons. Editing a database is described in detail in Chapter 4. Related to editing is printing, which is a sub-function of editing and is described in Chapter 5. Also related to editing is importing and exporting ASCII text files, which is described in Chapter 6.

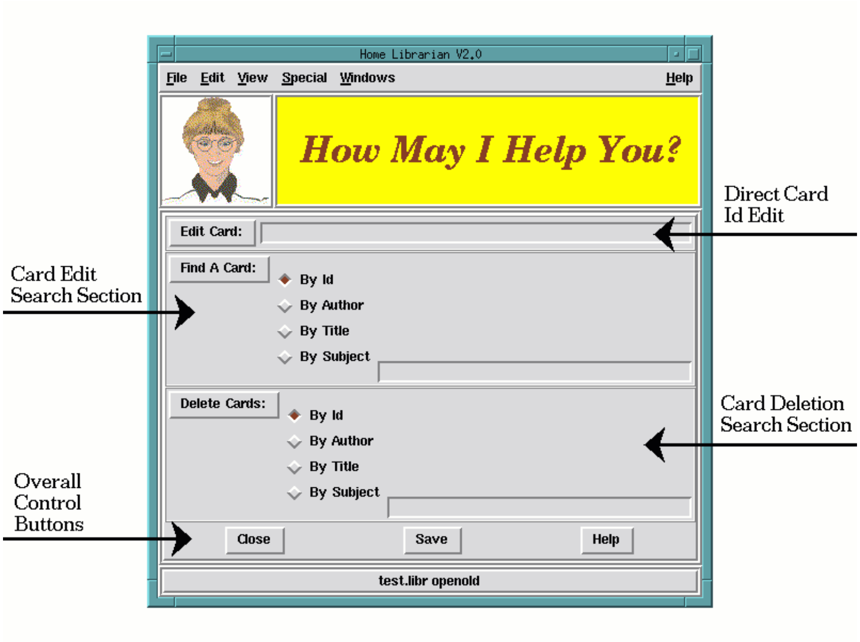


Figure 2.4: The Edit function Main GUI Window

# Chapter 3

# Searching a card catalog database

The search function opens a *Home Librarian* card catalog database file in read-only mode and provides an interface to search for items in the database.

## 3.1 Types of searches

There are three types of basic searches, by author, by title, or by subject. The search strings are considered a proper prefix to the keys in the database and are used to scan the selected index: author, title, or subject.

## 3.2 Search results and what to do with them

The results of the search are displayed in the search results listbox (see Figure 2.3). Depending on the number of matches, this will either be a list of possible exact keys or a listing of matching cards. Selecting one of these items allows one of two further actions, adding the item to the notepad (see Section 3.3) or getting more information on the displayed item, as shown in Figure 3.1.

### 3.2.1 The `Add To Notepad` button

Clicking the left button, labeled "Add To Notepad" adds one line of text to the notepad window. This provides a way to accumluate the results of several searches and later print the contents of the notepad window out or

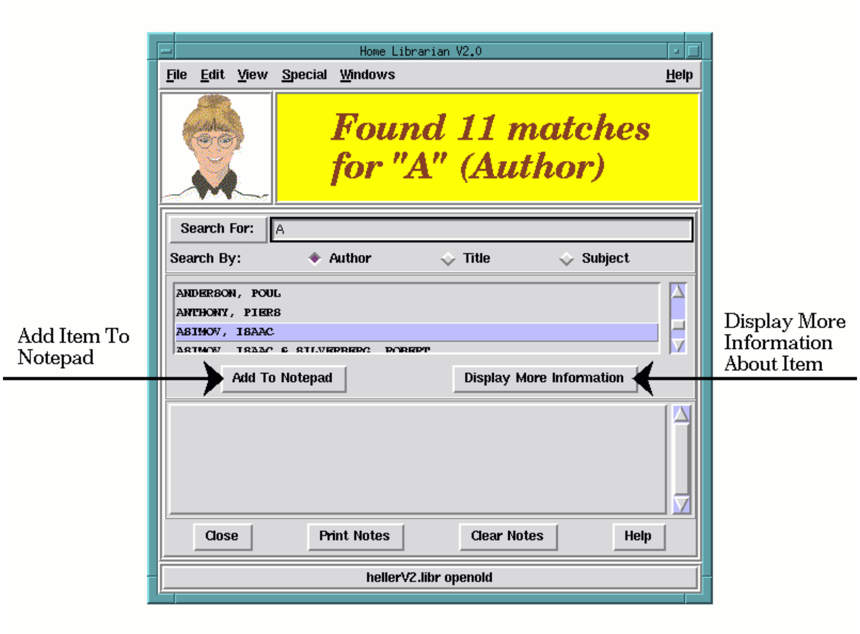


Figure 3.1: Selecting one of the items from the search results list

copy it to the clipboard to be pasted into a document or E-Mail message, etc.

### 3.2.2 The `Display More Information` button

Clicking the right button, labeled "Display More Information" expands on the information. If the item is a key, then the list of cards the key covers is displayed in a popup window, as shown in Figure 3.2. It the item refers to a single card, the contents of the card record is displayed in a card display popup window, as shown in Figure 3.3.

#### The card display popup window

A card record is displayed in a card display popup window, as shown in Figure 3.3. This popup window has two important buttons at the bottom: one labeled "Add To Notepad", which adds one line of text to the notepad window and one labeled "Print Card", which causes the card record to be printed out.

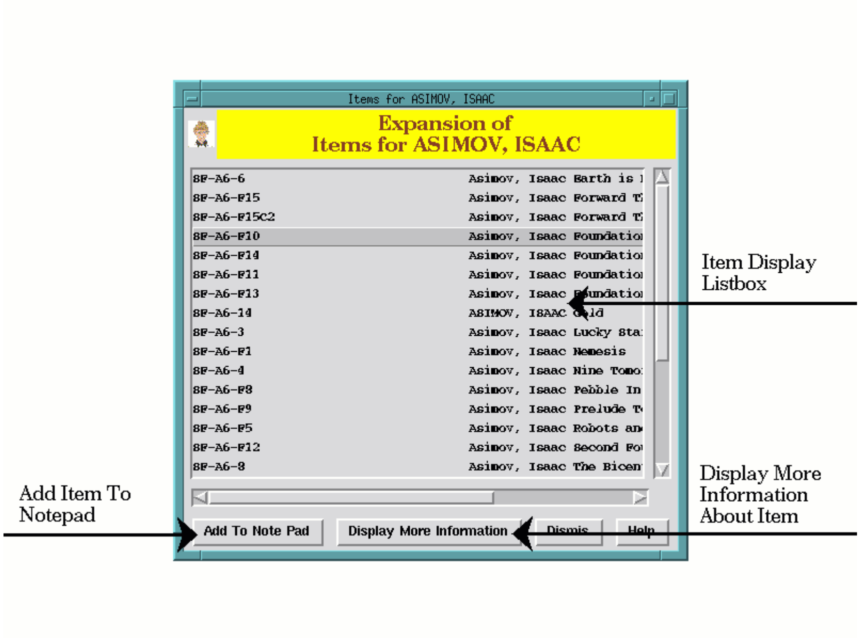


Figure 3.2: Item list popup window

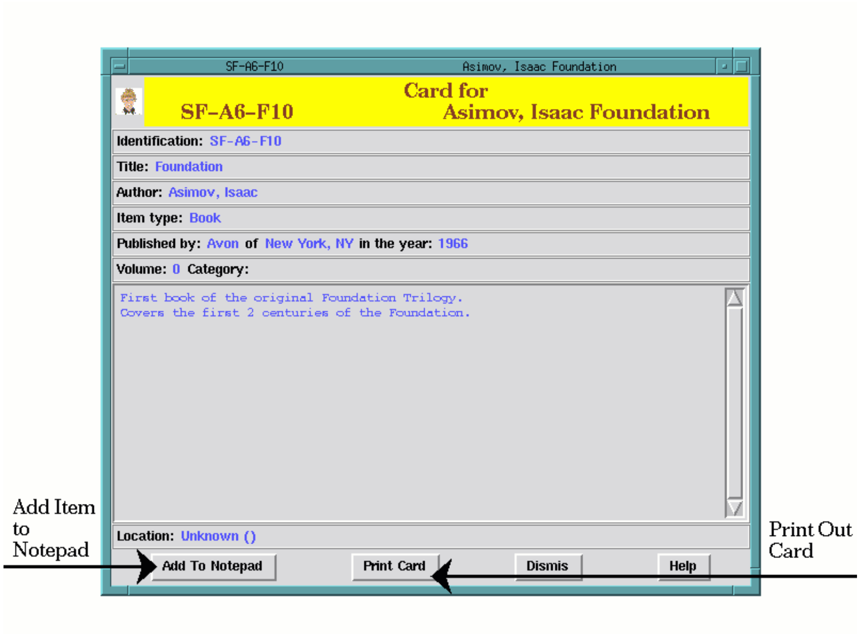


Figure 3.3: Card display popup window

## 3.3 The notepad

The notepad section of the search GUI is an editable text area, which can be used to accumulate notes on the items being searched. In addition to lines of text being added by the various “Add To Notepad” buttons, the user can type in notations or use the standard cut and paste operations to transfer text to and from the clipboard or other applications.

The contents of the notepad can also be sent to the printer with the button labeled “Print Notepad”.

# Chapter 4

# Creating and editing a card catalog database

The edit and create function opens a *Home Librarian* card catalog database in read-write mode (the create function creates a new and empty database file). The edit function also provides for printing (see Chapter 5) and exporting (see Chapter 6). The import function (see Chapter 6) on the `File` menu also enters the edit function GUI.

## 4.1 Creating or editing a card by its Id

If you know the (unique) identification of a card record you want to edit or are creating a new card record, you can enter the identification string in the direct card id edit entry (see Figure 2.4). This pops up a card record edit popup window[1]. A card edit popup window is displayed, as shown in Figure 4.1.

### 4.1.1 Layout of the edit card popup window

The edit card popup window contains a eight editing sections, from top to bottom:

1. Title section. This section contains an entry for the item's title and a button to edit the list of title keys for this item in the title index.

[1]If this is a new card, you will be asked to select a template card to use as the initial contents of the card record. See Chapter 7

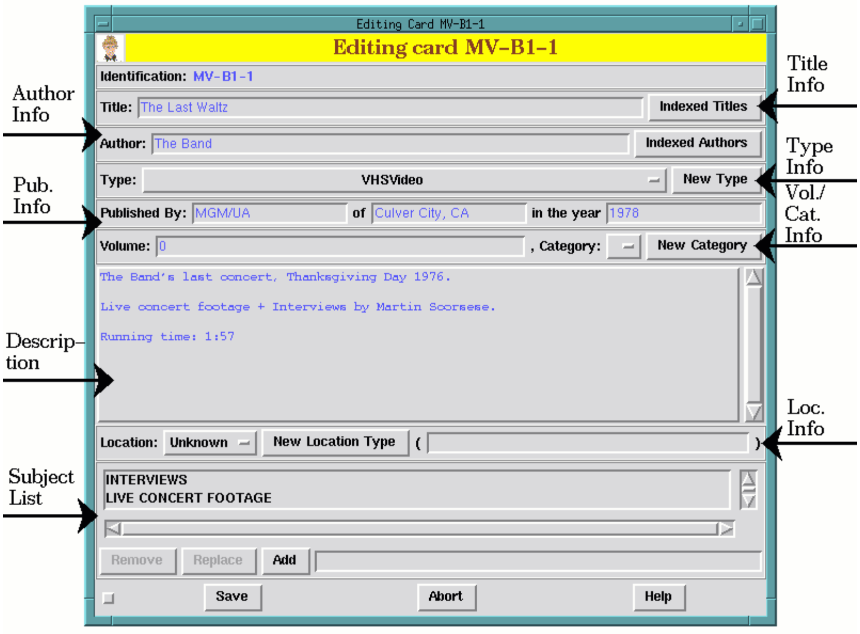


Figure 4.1: The Edit Card popup window

2. Author section. This section contains an entry for the item's author and a button to edit the list of author keys for this item in the author index.

3. Type section. This section contains an option menu for the item's type and a button to create a new item type.

4. Publisher section. This section contains entries for the publisher, the publiser's city, and the publish year.

5. Volume and Category section. This section has an entry for the volume number and an option menu for the item's category and a button to create a new category.

6. Description section. This is a text area for descriptive text about the item.

7. Location section. This section contains an option menu for the item's location type, a button to create a new location type, and an entry for location details.

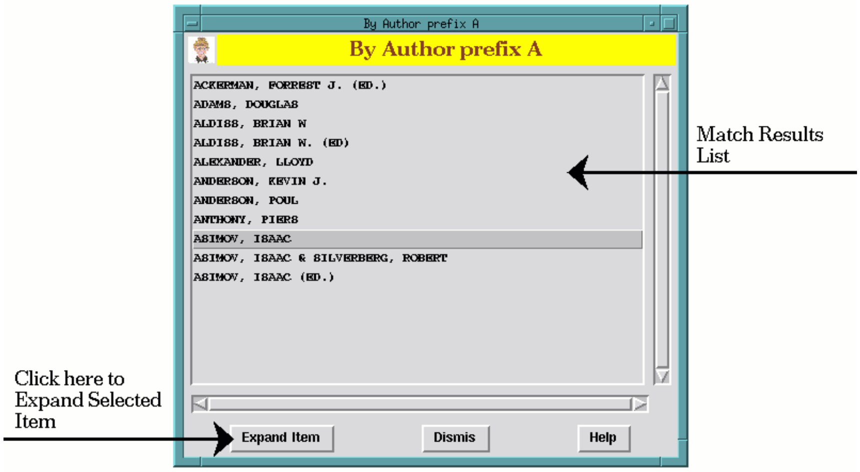


Figure 4.2: Author prefix edit search results

8. Subject list section. This section contains a list box containing the item's subject keys, along with buttons to remove, add, or replace a subject key and an entry to edit old (replace) or enter new (add) subject keys.

## 4.2 Finding one or more cards to edit by searching

The card edit search section of the edit function main GUI window (see Figure 2.4) provides for searching for one or more cards to edit. You can search by id, author, title, or subject prefix. Searching by author will produce a popup window much like Figure 4.2. The selecterd author key can be expanded to produce a popup window much like Figure 4.3. A listed card id can then be selected for editing.

The *Home Librarian* application will automatically expand the search results to an id list (like Figure 4.3) when the number of displayed item would be fewer than 5 in number.

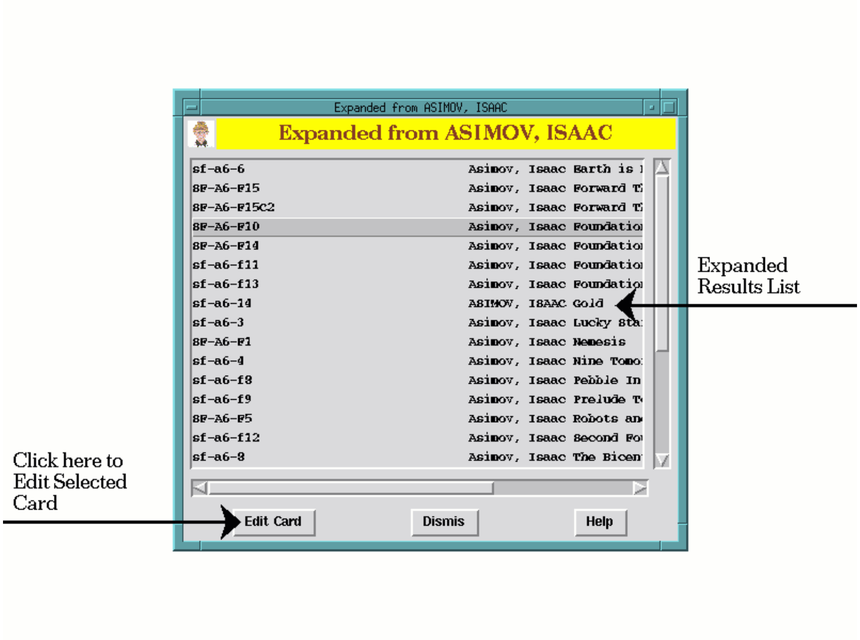


Figure 4.3: One author expanded

## 4.3 Finding one or more cards to delete by searching

The card delete search section of the edit function main GUI window (see Figure 2.4) provides for searching for one or more cards to delete. You can search by id, author, title, or subject prefix. This function works just like the card edit search section (see Section 4.2), except you are given the option "Delete Card" on the expansion to id popup window.

# Chapter 5

# Printing a card catalog database

A card catalog database can be printed out in two predefined and an indefinate number of user-defined formats. The database can be printed in whole or in part, based on a number of criteria. The database can be printed in id (a "shelf list"), author, title, or subject order.

Printing is selected while in edit mode from the `File` menu, as shown in Figure 5.1.

## 5.1 Print dialog

Selecting the `Print` item on the `File` menu pops up the print dialog, as shown in Figure 5.2. The print dialog has four main sections:

1. The Print Destination Selection section[1]. This section selects where to send the output, either to a PostScript file or a PostScript printer via the lpr command.

2. The Print Format Selection section. This section selects the format of the output. The choices are $3 \times 5$ cards, $5 \times 8$ cards, or another (custom) format (see Section 5.2).

3. The Print Order Selection section. This section selects ordering of the output. The choices are "Shelf List" – ordered by identifier, "by

[1]The MS-Windows and MacOS versions ommit this section – its functionallity is replaced by the native printer job and style dialog boxes.

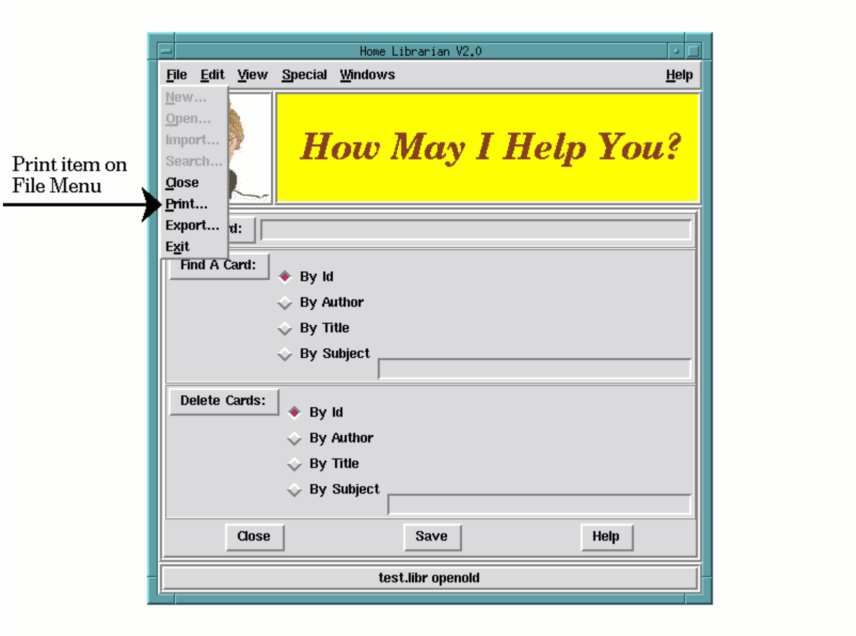


Figure 5.1: `Print` item on the `File` menu

Subject" – ordered by subject, "by Title" – ordered by title, or "by Author" – ordered by author.

4. Print Subset Selection section. This section selects a subset of items to print. This defines a collection of logical tests comparing various fields to a constant. Multiple tests all have to succeed before the item will be printed.

### 5.1.1 Print Destination Selection section

The print destination selection section is used to select where the printout will go: to a PostScript printer or to a PostScript file. This section is only displayed in the UNIX version – this functionallity is handled by the printer API job dialog boxes available under MS-Windows and MacOS.

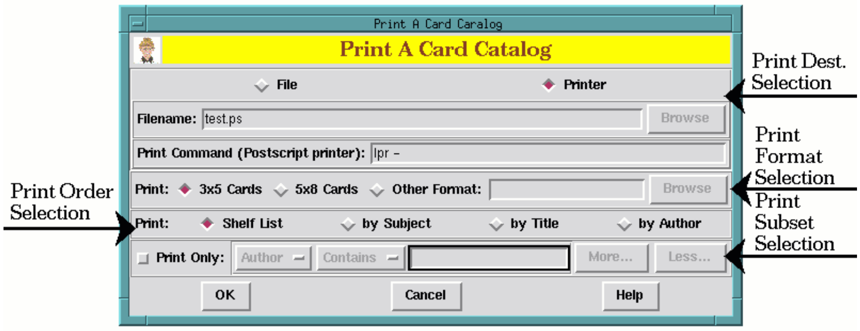


Figure 5.2: Print Dialog

### 5.1.2 Print Format Selection section

The print format selection section selects the format of the printout. The format can be either the of the two "built in" formats, for $3 \times 5$ cards or $5 \times 8$ cards, or else it can be some custom format, as specified by a file containing the Tcl code to implement the custom format. See Section 5.2 for information on creating a custom format file.

### 5.1.3 Print Order Selection section

The print order selection section selects the print ordering. There are four possible orderings, by identification (called a "Shelf List"), by Subject, by Title, or by Author.

### 5.1.4 Print Subset Selection section

The print subset selection section selects a subset of items to print. The subset is defined as a set of one or more comparison operations. Each operation compares one field of the item with some constant value. The fields that can be compared are `Author`, `Title`, `Type`, `Location Type`, `Location`, `Category`, `Publisher`, `Id`, `Volume`, `Year`, or `City`. The possible comparisons are `Contains`, `Is`, `Is Not`, `Is Less Than`, `Is Greater Than`, or `Matches (regular expression)`. The comparisons work as follows:

**Contains** The string specified in the comparison occurs somewhere in the field's value.

**Is** The string specified in the comparison is exactly the same as the field's value.

**Is Not** The string specified in the comparison is completely different from the field's value.

**Is Less Than** The string specified in the comparison is later in lexagraphical order then the field's value.

**Is Greater Than** The string specified in the comparison is earlier in lexagraphical order then the field's value.

**Matches (regular expression)** The string specified in the comparison is a regular expression and a regular expression match is applied using this expression to the field's value.

## 5.2 Custom Print Format files

Custom print format files are just Tcl scripts that define a set of procedures to handle the printing of card records. There are two ways these files are created, either with the Print Layout Generator, as described in Section 5.2.1, or by manually editing a text file, as described in Section 5.2.2.

### 5.2.1 Using the Print Layout Generator

The Print Layout Generator is invoked from the `Special` menu as shown in Figure 5.3. The Print Layout Generator dialog looks like Figure 5.4. There are two main sections to this dialog, the layout sheet area and the insertable items list. The layout sheet shows a sample of what the printout will be like. In additional to the usual scrollbars, it can also be dragged around with the middle mouse button[2].

[2]On UNIX. Under MS-Windows, use Ctrl+Left buton and Command ("cloverleaf") + button on the Mac.


Figure 5.3: `Edit or Create a Print Layout file` item on the `Special` menu

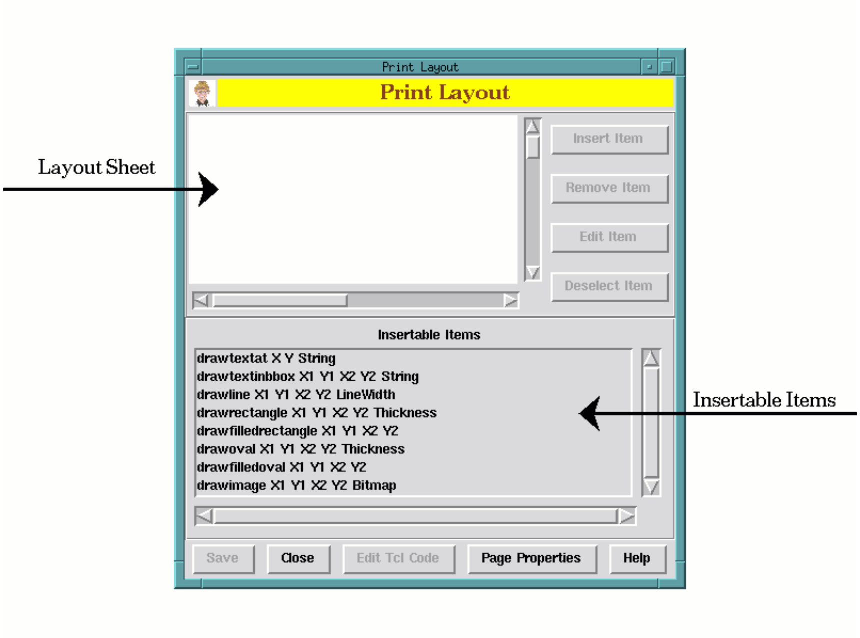


Figure 5.4: Print Layout Generator dialog

### Inserting, editing, and deleting items

To insert an item, select the item type from the item list and click insert. An item specific dialog box is popped up with the items properties. Each item type has a different set of properties.

**drawtextat X Y String** The dialog box for `drawtextat` is shown in Figure 5.5. The text field uses the percent character to introduce places to insert card specific information, as shown in Table 5.1.

**drawtextinbbox X1 Y1 X2 Y2 String** The dialog box for `drawtextinbbox` is shown in Figure 5.6. The text field is treated the same way as for `drawtextat X Y String` – see Table 5.1.

**drawline X1 Y1 X2 Y2 LineWidth** The dialog box for `drawline` is shown in Figure 5.7.

**drawrectangle X1 Y1 X2 Y2 Thickness** The dialog box for `drawrectangle` is shown in Figure 5.8.

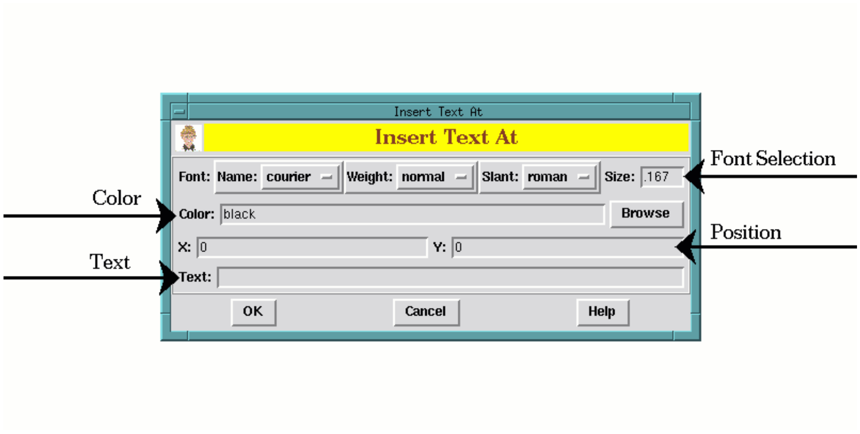


Figure 5.5: Insert Text At properties dialog

| Key | Description |
|---|---|
| % | Insert a % |
| heading | Insert the heading for this card |
| id | Insert the id key for this card |
| cardtype | Insert the item type for this card |
| author | Insert the author |
| title | Insert the title |
| publisher | Insert the publisher |
| city | Insert the publisher's city |
| description | Insert the item's description |
| vol | Insert the volume number |
| year | Insert the year published |
| locationtype | Insert the location type |
| locationdetail | Insert the location detail |
| category | Insert the category |
| subjects | Insert the subject list |
| authors | Insert the author list |
| titles | Insert the title list |

Table 5.1: Percent (%) substitutions in text fields

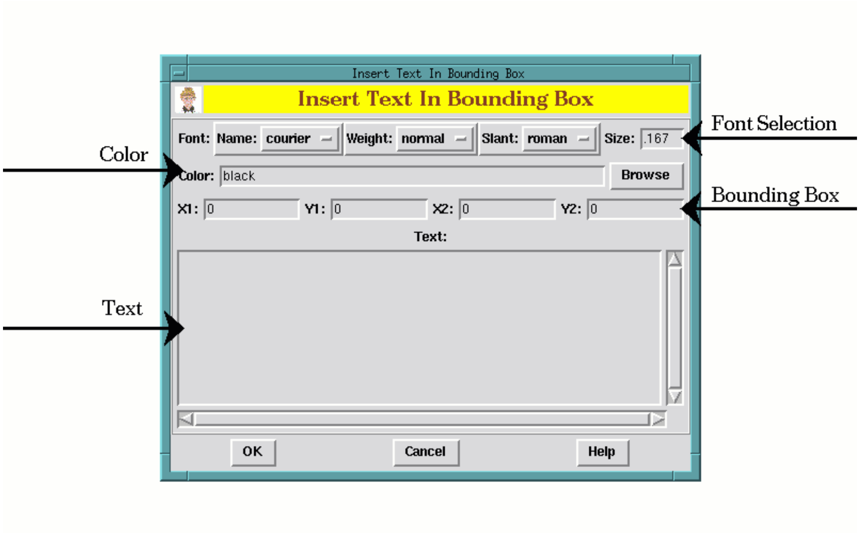


Figure 5.6: Insert Text In Bounding Box properties dialog

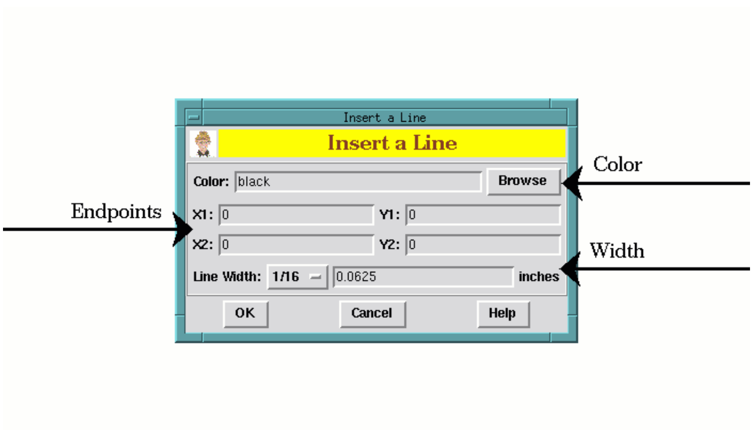


Figure 5.7: Insert Line properties dialog

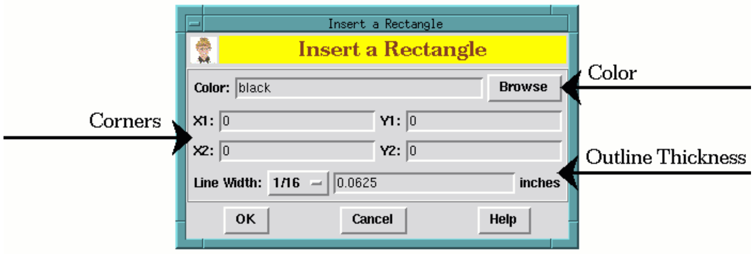


Figure 5.8: Insert Rectangle properties dialog

**drawfilledrectangle X1 Y1 X2 Y2** The dialog box for `drawfilledrectangle` is shown in Figure 5.9.

**drawoval X1 Y1 X2 Y2 Thickness** The dialog box for `drawoval` is shown in Figure 5.10.

**drawfilledoval X1 Y1 X2 Y2** The dialog box for `drawfilledoval` is shown in Figure 5.11.

**drawimage X1 Y1 X2 Y2 Bitmap** The dialog box for `drawimage` is shown in Figure 5.12.

#### Setting the page properties

Clicking the `Page Properties` button on the Print Layout Generator dialog (Figure 5.4), brings up the Page Properties dialog, as shown in Figure 5.13. This dialog box allows setting the size of each card image, the number of card images per row and column on each page and the row and column offsets of each row and column of card image per page.

### 5.2.2 Manualy creating your Custom Print Format files

If you are familure with the Tcl language, you might prefer to write your own Print Format files without the Print Layout Generator dialog. A Print Format file is just a Tcl script which defines these three procedures:

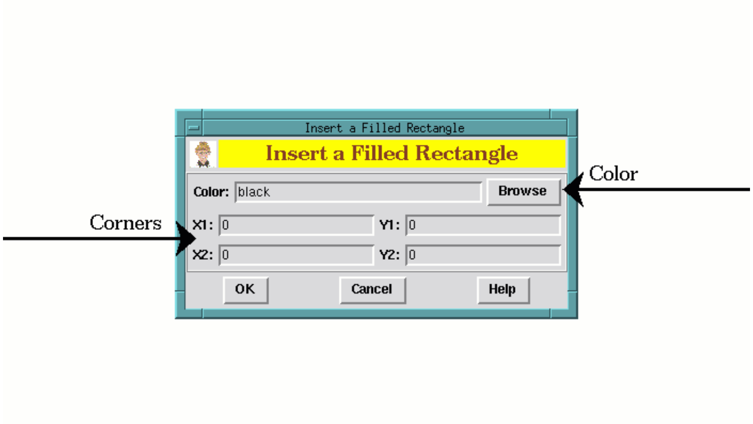


Figure 5.9: Insert Filled Rectangle properties dialog

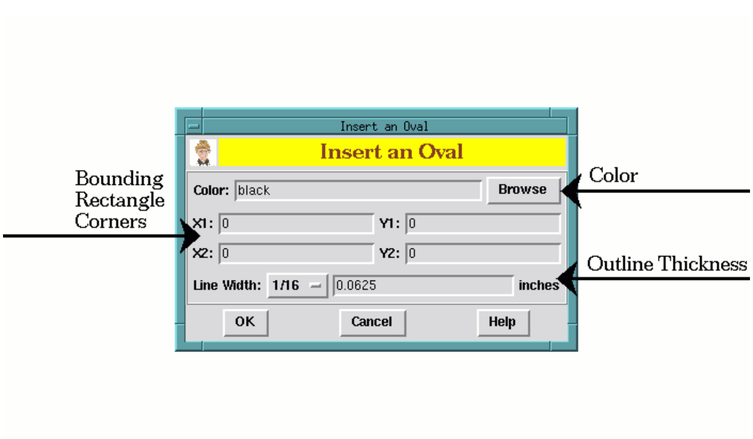


Figure 5.10: Insert Oval properties dialog

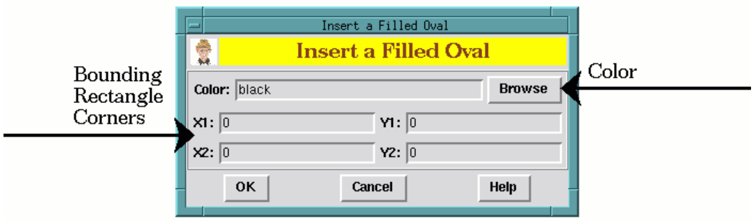


Figure 5.11: Insert Filled Oval properties dialog

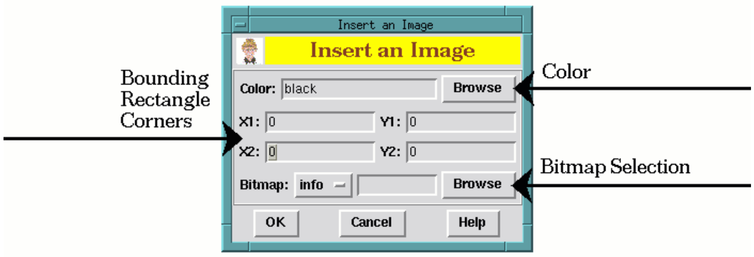


Figure 5.12: Insert Image properties dialog

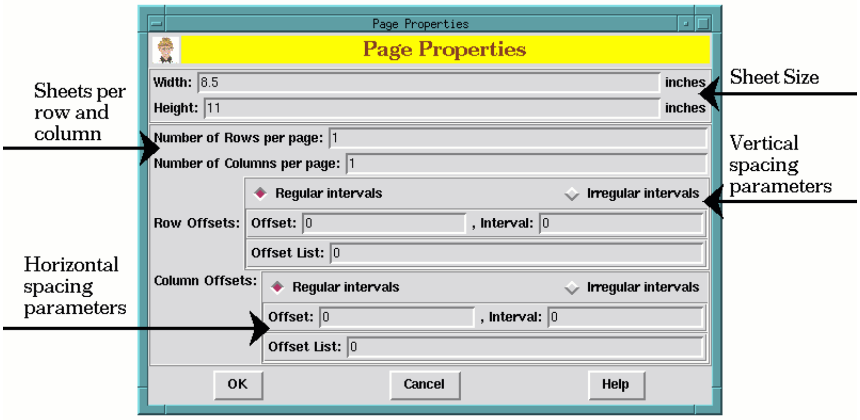


Figure 5.13: `Page Properties` dialog

**BeginCards** The procedure is run just after opening the document and before printing any card images. You can do any initialization here that needs the printer state. This procedure is optional. If not defined, a dummy no-op version is defined.

**EndCards** This procedure is run just after the last card has been processed and before closing the document. You can do any clean up here, such as flushing the last page. Like `BeginCards`, this procedure is optional. If not defined, a dummy no-op version is defined.

**DoCard** This procedure is run for each card image that will be printed. It takes three arguments:

**heading** This is the heading string to put on this card.

**id** This is the id key string for this card.

**record** This is the `CardRecord` handle for this card. This handle is a Tcl command that can be queried to get the various field values for the card's item. See the separate manual, *A Computerized*

*Library Card Catalog System Internals Manual*, for a complete details on `CardRecord` handles.

The script runs in a *safe* Tcl interpreter. Several useful function aliases are provided:

**printer** This interfaces to the printer. See the separate manual, *A Computerized Library Card Catalog System Printer Interface Documentation*, for complete details on this procedure.

**FetchSubjects** This procedure fetches the subject list. It takes one argument, the card's id key.

**FetchAuthors** This procedure fetches the author list. It takes one argument, the card's id key.

**FetchTitles** This procedure fetches the title list. It takes one argument, the card's id key.

**FetchTypeName** This procedure translates the item type code to an item type name. It takes one argument, the type code.

**FetchLTypeName** This procedure translates the location type code to a location type name. It takes one argument, the location type code.

**FetchCategory** This procedure translates a category number to a category name. It takes one argument, the category number.

In additional the standard output and standard error streams are duplicated from the main interpreter. It is best to have a good understanding of Tcl and to have the Printer Interface Documentation and the Internals Manual at hand.

# Chapter 6

# Importing and exporting a card catalog database

## 6.1 Importing from an ASCII text file

You can import a database in the form of an ASCII text file using the `Import` item under the `File` menu. The menu item pops up the `Import Card Catalog Library` dialog box as shown in Figure 6.1. There is a space for supplying the source (ASCII text) file, a place to supply the *Home Librarian* database file to create, and the format to read. There are three formats supported:

**V1 ASCII** This is the format generated by the V1 Libr2Ascii program.

**V2 ASCII** This is the V2 ASCII format generated by the version 2.0 *Home Librarian* Tcl/Tk GUI.

**Delimited** This is a set of 4 delimited files files, one for each index. The section below the import format describes the delimiter and quote and escape characters.

## 6.2 Exporting a database to an ASCII text file

You can export a currently open for editing database file to an ASCII text file by selecting the `Export` item under the `File` menu. The menu item pops

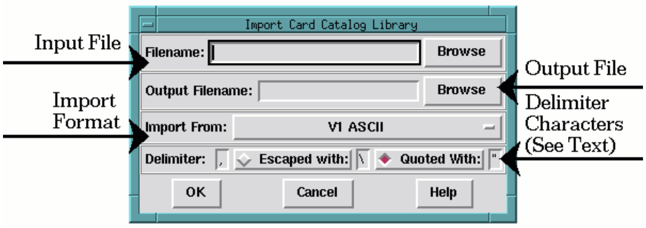


Figure 6.1: `Import Card Catalog Library` dialog

up the `Export Current Card Catalog` dialog box as shown in Figure 6.2. The is a space for supplying the destination (ASCII text) file and the format to use. There are four formats supported:

**V1 ASCII** This is the format readable by the V1 Ascii2Libr program.

**V2 ASCII** This is the V2 ASCII format generated by the version 2.0 *Home Librarian* Tcl/Tk GUI.

**Delimited** This is a set of 4 delimited files files, one for each index. The section below the export format describes the delimiter and quote and escape characters.

**Formatted** This is a simple human-readable text file. It is not meant to be machine readable, but can be used as a simple text format for handy reference.

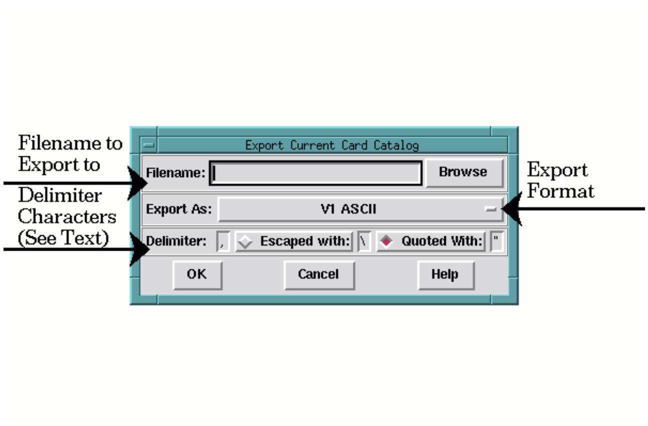


Figure 6.2: `Export Current Card Catalog` dialog

# Chapter 7

# Odds and ends: features and functions not covered elsewhere

## 7.1 Custom item types, location types, and categories

When you are editing a version 2 library, you can create custom item and location types (up to 10 each) and up to 256 item categories. You can create these custom types either by selecting the corresponding define item under the `Special` menu or by selecting the labeled button on the edit card dialog (see Figure 4.1).

### 7.1.1 Custom item types

Selecting `Define a new Card Type` item under the `Special` menu pops up the `New Card Type` dialog box as shown in Figure 7.1.

There are ten type codes, each of which can be associated with a single word or phrase in each card catalog database file.

### 7.1.2 Custom location types

Selecting `Define a new Location Type` item under the `Special` menu pops up the `New Location Type` dialog box as shown in Figure 7.2.

There are ten location type codes, each of which can be associated with a single word or phrase in each card catalog database file.


Figure 7.1: `New Card Type` dialog


Figure 7.2: `New Location Type` dialog

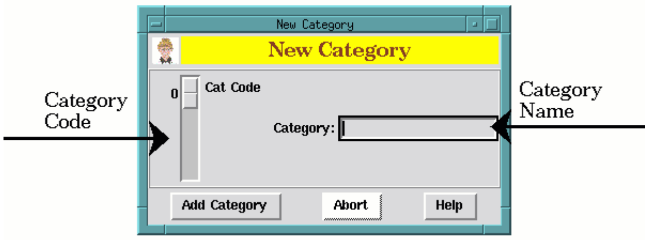


Figure 7.3: `New Category` dialog

### 7.1.3 Categories

Selecting `Define a new Category` item under the `Special` menu pops up the `New Category` dialog box as shown in Figure 7.3.

There are 256 category codes, each of which can be associated with a single word or phrase in each card catalog database file.

## 7.2 Setting and showing the registration information

When you get a registration serial number for your copy of the *Home Librarian* package you should set the registration. You can also check the registration at any time.

### 7.2.1 Setting the registration

Setting the registration is done by selecting the `Register Home Librarian` menu item under the `View` menu. This menu item pops up the `Registration` dialog box as shown in Figure 7.4. This dialog box has three free-form text fields. It is only necessary to fill in the *Name* and `Serial Number` fields.


Figure 7.4: `Registration` dialog

### 7.2.2 Showing the registration

You can view the current registration by selecting the `Show Registration` menu item under the `View` menu. This menu item pops up the `Show Registration` dialog box as shown in Figure 7.5.

## 7.3 Converting a Version 1 database file to a Version 2 database file

You can convert an old Version 1 database file[1] to a Version 2 database file by selecting the `Convert a V1 library to V2` menu item under the `Special` menu. You will be asked for source and destination files.

## 7.4 Template Cards

To aid in editing your database, you can create "template" cards. These are cards that you have set up with some pre-set fields. On the `Special` menu are a set of menu items relating to handling these "template" cards, either one at a time or in groups.

[1] You don't need to convert old database files to edit them with this version of the Home Librarian package, unless you wish to make use of the new edit fields and other features.


Figure 7.5: `Show Registration` dialog

**Create a template Card** Create a new template card. You will be asked for a name for this template. Then the Edit Card popup window (See Figure 4.1) will be poped up.

**Edit a template Card** Edit an existing template card. You will be asked for the name of the template card to edit. The Edit Card popup window will be poped up.

**Remove a template Card** Remove a named template card from memory. You cannot remove the builting blank template.

**Save Card template** Save a template card to a file. You will be asked for a file and then a textual representation of the card is appended to the named file.

**Load Card templates** You will be asked for a file containing one or more template cards. This file will be read in and the template cards defined in it added to the list of available template cards. If a card in the file has the same name as a previous loaded or created card it replaces the previous defination.

# Appendix A

# License and Warranty Information.

## A.1 Copyright.



## A.2 License.

### A.2.1 License to use.

The software package known as the *Home Librarian* package is licensed to anyone to use at no charge or fee, on any number of machines, by any number of users.

### A.2.2 License to copy.

The *Home Librarian* package may be freely copied, **SO LONG AS THE COMPLETE PACKAGE OF LIBRARIES, SCRIPTS, AND EXECUTABLES (AND/OR COMPLETE SOURCE CODE) AND THE DOCUMENTATION FILES ARE COPIED TOGETHER AS**

**A UNIT**. No charge or fee may be demanded for a copy of this package[1].

### A.2.3 No Support.

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

## A.3 No Warranty.

**BECAUSE THE *Home Librarian* PACKAGE IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY 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 PERMITTED 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 BY YOU OR THIRD PARTIES OR A FAILURE OF THE PACKAGE TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR**

[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.

**OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.**

# Index