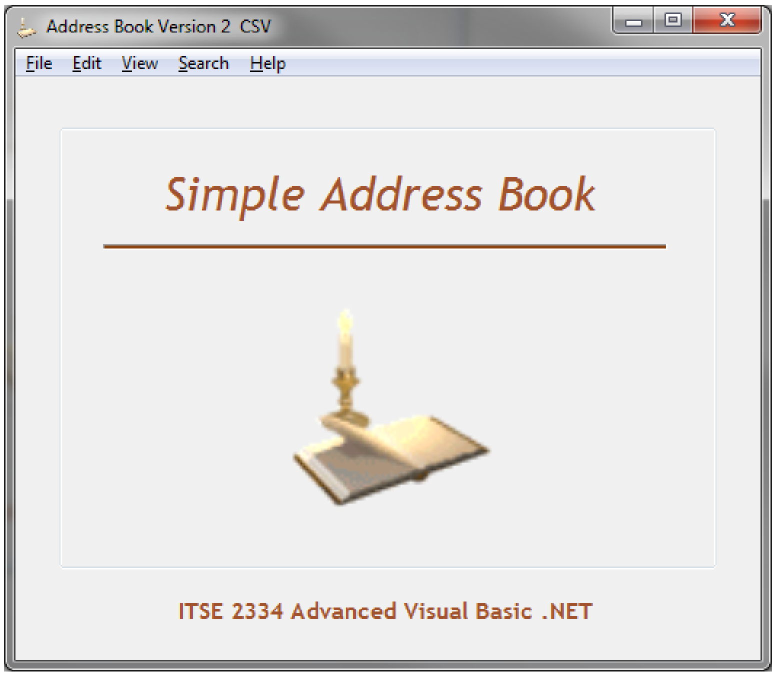
Chapter 9: Address Book Revisited

For this assignment, we will create a simple address book that stores specific data for each address entry. All address book data is read from an external CSV file and stored in an array of structures when the program begins. When the program is running, address entry data must be read from and written to the array of structures in memory. When the program closes, all data stored in the array of structures is written to and archived in the external CSV file. Each entry in the address book will consist of the following data fields:

- First Name
- Last Name
- Street Address
- City
- State
- Zip Code
- Phone Number
- Email Address.

The project will require several forms demonstrated below. Note: the forms shown are just an example. You are welcome to design your own forms but, make sure you include all the program features I am showing in your version.

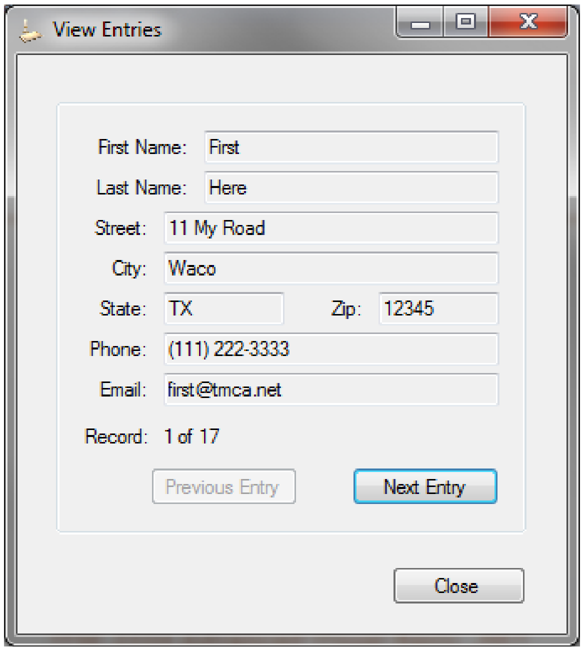
Main Form


Address Book Version 2 CSV
File Edit View Search Help
Simple Address Book
ITSE 2334 Advanced Visual Basic .NET

The Main form contains a menu that includes the following items:

- File: Print, Exit
- Edit: Update Entry, Insert Entry
- View: View Entries
- Search: By Name
- Help: About

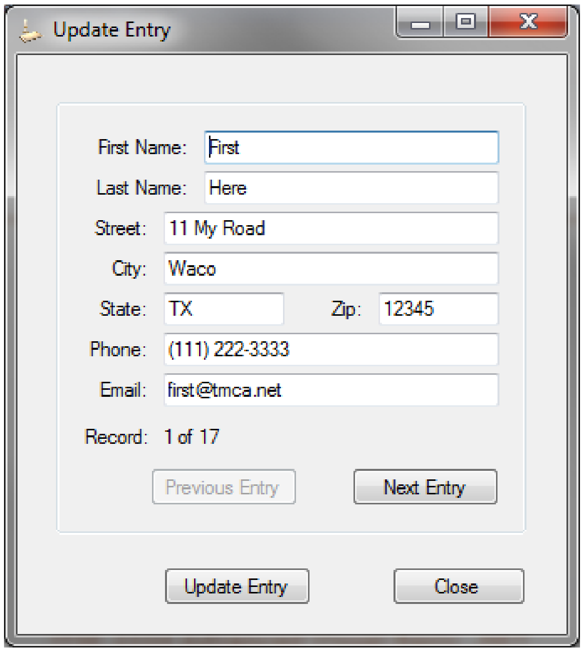
View Entry Form



First Name:	First
Last Name:	Here
Street:	11 My Road
City:	Waco
State:	TX
Zip:	12345
Phone:	(111) 222-3333
Email:	first@tmca.net
Record:	1 of 17

The View Entries form allows the user to view all entries in the address book. When the View Entries form is opened, it will always read from the first entry in the array of structures and display the first address entry. Remember, this is a sequential read operation based on the array indices. Since we are at the first record, the Previous Entry button should be disabled. This allows the user only the ability to look at the next record. If the user is on record two or greater, allow access to the Previous Entry button and the Next Entry button which lets the user either view the next record or return to the previous record. If the user is viewing the last record, the Next Entry button should be disabled and only the Previous Entry button should be enabled since no more records exist. Please note the View Entries form does not allow the user to modify any record data. Also, provide a way for the user to return to the main form. In this case, the Close button closes the View Entry form and returns to the Main form.

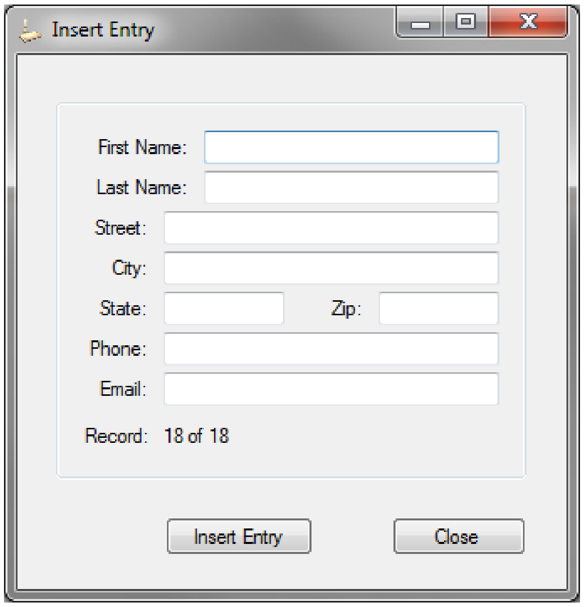
Update Entry Form



First Name:	First
Last Name:	Here
Street:	11 My Road
City:	Waco
State:	TX
Zip:	12345
Phone:	(111) 222-3333
Email:	first@tmca.net
Record:	1 of 17

The Update Entries form is much like the View Entries form except it does allow the user to modify all entry data and contains an Update Entry button. When the Update Entries form is opened, it will always read from the first entry in the array of structures and display the first address entry. Again, this is a sequential read operation based on the array indices. When the user displays the entry they wish to update (using the Previous / Next buttons), the user can update any data in the record and then click the Update button to save the data back to the current record of the array of structures. The Previous / Next buttons should use the same enable / disable logic described on the View Entries form.

Insert Entry Form

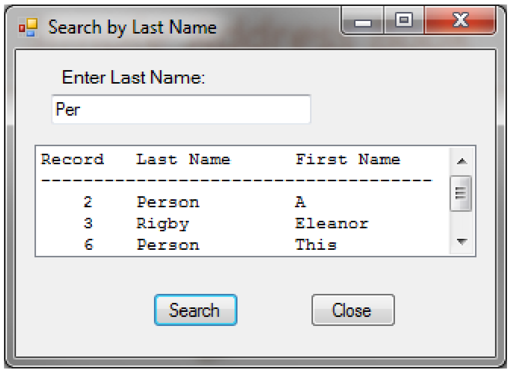


Record: 18 of 18

The Insert Entry form allows the user to enter data fields that make up a new record entry. The Insert Entry button allows the user to append or add this entry directly to the array of structures that holds the current address book data in memory. You should require that a few items are not blank before allowing the user to enter a new record. For instance, my program requires First Name, Last Name, and Phone or Email entries before allowing the record to be written. If the user fails to include the required fields, I alert the user to this error and allow the user to continue or cancel the insert operation. Please note in the View Entries and Update Entries forms above there are a total of 17 records in the address book. When the Insert Entry form opens the user is provided with a blank form that indicates the new record is the 18th record in the address book. If the user fills in the required data fields and then clicks the Insert Entry button, the array of structures must be expanded to hold the new record. At no time while the program is running should the array of structures contain more record elements than is necessary to hold the existing data. The Cancel button allows the user to ignore the insert record operation and return to the main form without making any changes to the array of structures in memory. The Close button closes the Insert Entry form and returns to the Main form.

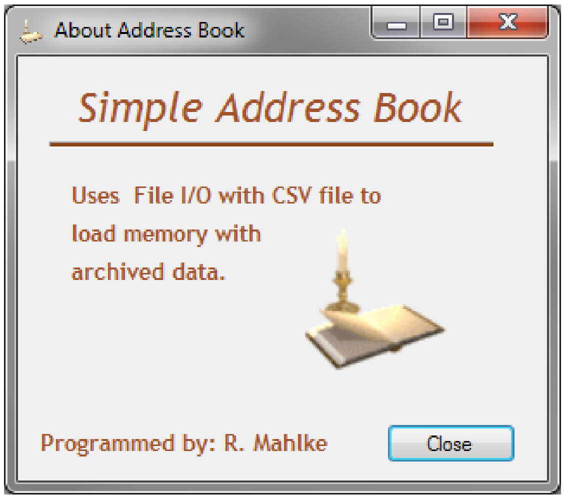
Search by Last Name Form

The Search by Last Name form allows the user to enter some text in the Last Name textbox and the form will display a result that includes all entries that are greater than or equal to the text the user enters. The Close button closes the Search by Last Name form and returns to the Main form.



```
Record   Last Name      First Name
-------------------------------------
   2     Person         A
   3     Rigby          Eleanor
   6     Person         This
```

About Form

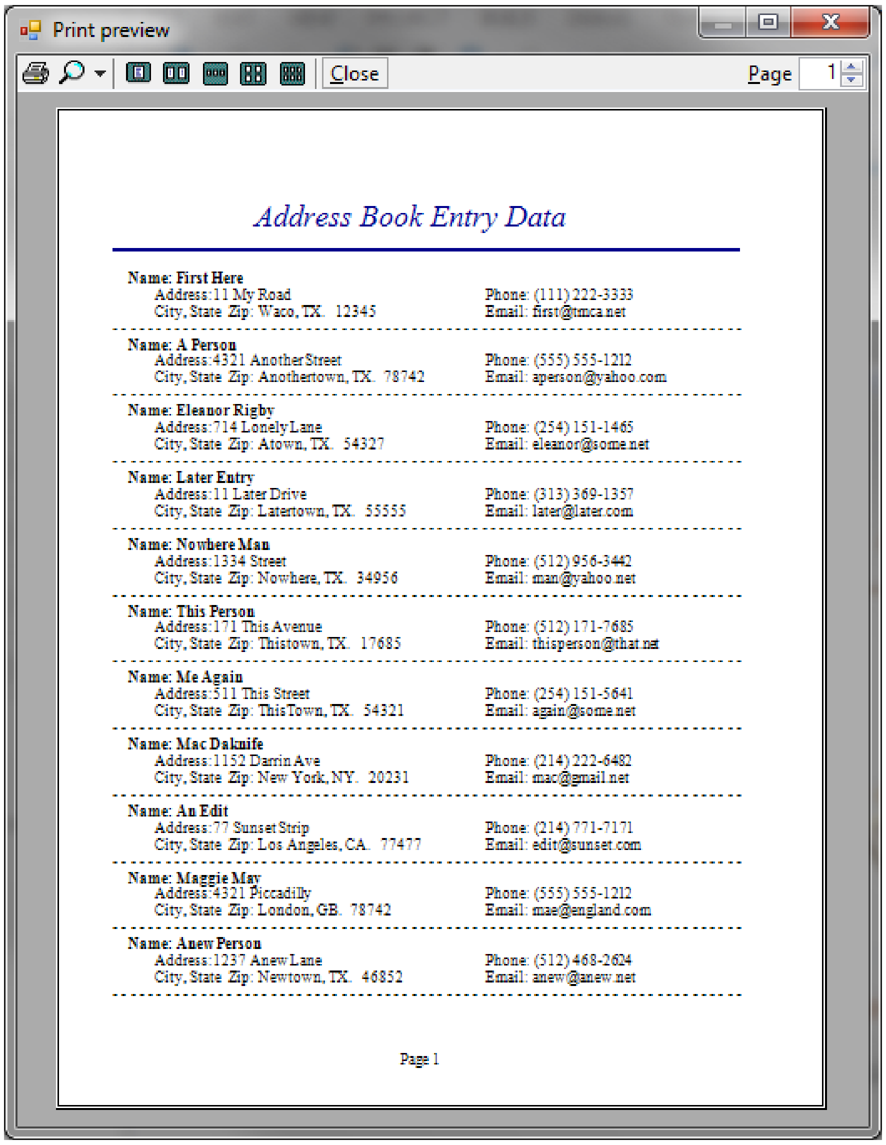


Uses File I/O with CSV file to load memory with archived data.

Programmed by: R. Mahlke

This form should identify your project (please provide your name) and provide a method for the user to return to the main form. Other than that, I will leave any other features up to you.

Print Preview and Print Functions



Address Book Entry Data

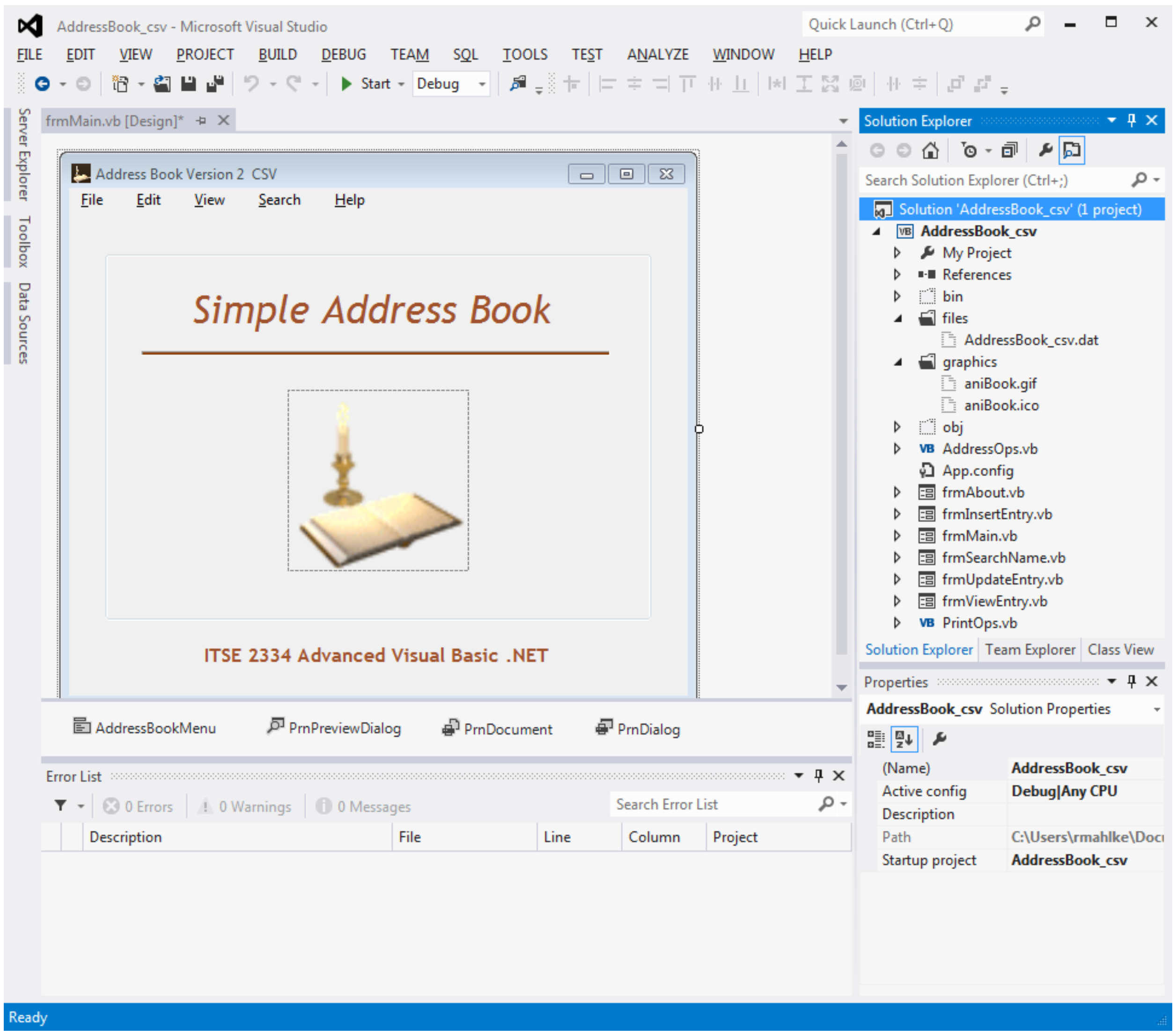
```
Name: First Here
     Address:11 My Road                        Phone: (111) 222-3333
     City, State  Zip: Waco, TX.  12345        Email: first@tmca.net
--------------------------------------------------------------------------
Name: A Person
     Address:4321 AnotherStreet                Phone: (555) 555-1212
     City, State  Zip: Anothertown, TX.  78742 Email: aperson@yahoo.com
--------------------------------------------------------------------------
Name: Eleanor Rigby
     Address:714 Lonely Lane                   Phone: (254) 151-1465
     City, State  Zip: Atown, TX.  54327       Email: eleanor@some.net
--------------------------------------------------------------------------
Name: Later Entry
     Address:11 Later Drive                    Phone: (313) 369-1357
     City, State  Zip: Latertown, TX.  55555   Email: later@later.com
--------------------------------------------------------------------------
Name: Nowhere Man
     Address:1334 Street                       Phone: (512) 956-3442
     City, State  Zip: Nowhere, TX.  34956     Email: man@yahoo.net
--------------------------------------------------------------------------
Name: This Person
     Address:171 This Avenue                   Phone: (512) 171-7685
     City, State  Zip: Thistown, TX.  17685    Email: thisperson@that.net
--------------------------------------------------------------------------
Name: Me Again
     Address:511 This Street                   Phone: (254) 151-5641
     City, State  Zip: ThisTown, TX.  54321    Email: again@some.net
--------------------------------------------------------------------------
Name: Mac Daknife
     Address:1152 Darrin Ave                   Phone: (214) 222-6482
     City, State  Zip: New York, NY.  20231    Email: mac@gmail.net
--------------------------------------------------------------------------
Name: An Edit
     Address:77 Sunset Strip                   Phone: (214) 771-7171
     City, State  Zip: Los Angeles, CA.  77477 Email: edit@sunset.com
--------------------------------------------------------------------------
Name: Maggie May
     Address:4321 Piccadilly                   Phone: (555) 555-1212
     City, State  Zip: London, GB.  78742      Email: mae@england.com
--------------------------------------------------------------------------
Name: Anew Person
     Address:1237 Anew Lane                    Phone: (512) 468-2624
     City, State  Zip: Newtown, TX.  46852     Email: anew@anew.net
--------------------------------------------------------------------------

                                 Page 1
```

You are required to provide a print preview and a print function that can access the computer's default printer. Above is page 1 of a Print Preview showing all address book entries in my file. Remember, when the program is operating the file is not accessed for data. Instead, you must access the array of structures holding the data in memory. This requirement includes all print functions in the program.

Additional Notes:

- You might have a look at Chapter 9 for information on file I/O operations, structures, and print operations you will need for this project. Below I show a design time view of my Main form which should give you some insight into items needed in this assignment.


AddressBook_csv - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL TOOLS TEST ANALYZE WINDOW HELP
Start Debug
Server Explorer Toolbox Data Sources
frmMain.vb [Design]*
Address Book Version 2 CSV
File Edit View Search Help
Simple Address Book
ITSE 2334 Advanced Visual Basic .NET
AddressBookMenu PrnPreviewDialog PrnDocument PrnDialog
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'AddressBook_csv' (1 project)
AddressBook_csv
My Project
References
bin
files
AddressBook_csv.dat
graphics
aniBook.gif
aniBook.ico
obj
AddressOps.vb
App.config
frmAbout.vb
frmInsertEntry.vb
frmMain.vb
frmSearchName.vb
frmUpdateEntry.vb
frmViewEntry.vb
PrintOps.vb
Solution Explorer Team Explorer Class View
Properties
AddressBook_csv Solution Properties
Error List
0 Errors 0 Warnings 0 Messages
Search Error List
Description File Line Column Project
Ready

(Name)	AddressBook_csv
Active config	Debug|Any CPU
Description	
Path	C:\Users\rmahlke\Doc
Startup project	AddressBook_csv

- All objects should be named with "meaningful" names and you should use a consistent naming scheme throughout the program. As an example, I typically use camel casing for variable names and Hungarian or Pascal casing for object names. It's most important that you do not use default names for variables and objects. Default names simply do not provide the "self-documenting" features that meaningful names provide.

- If you have questions, bring them to class and we can discuss the issue and find a solution to the problem.