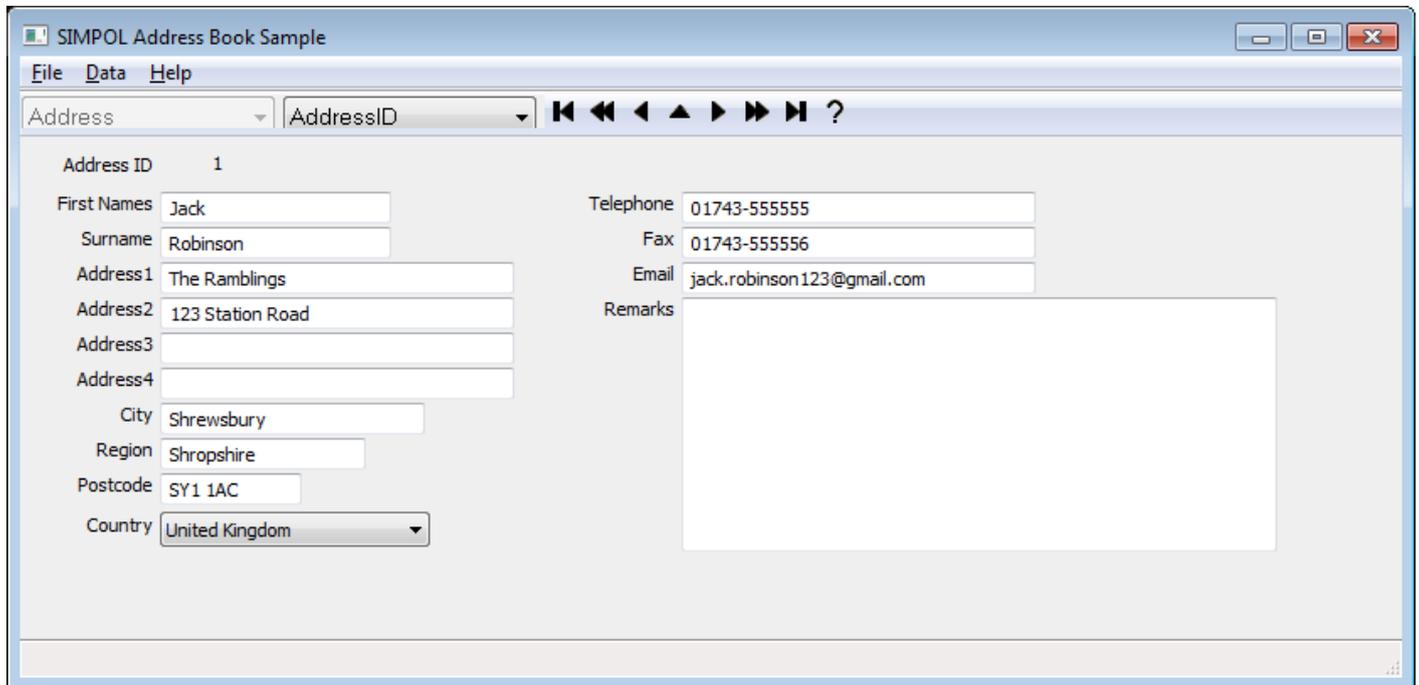


The Finished Product

After all of that work, when we finally run it it shows up looking like the image below:



The screenshot shows a window titled "SIMPOL Address Book Sample" with a menu bar (File, Data, Help) and a toolbar with navigation icons. The main area contains a form for address details. The form fields are as follows:

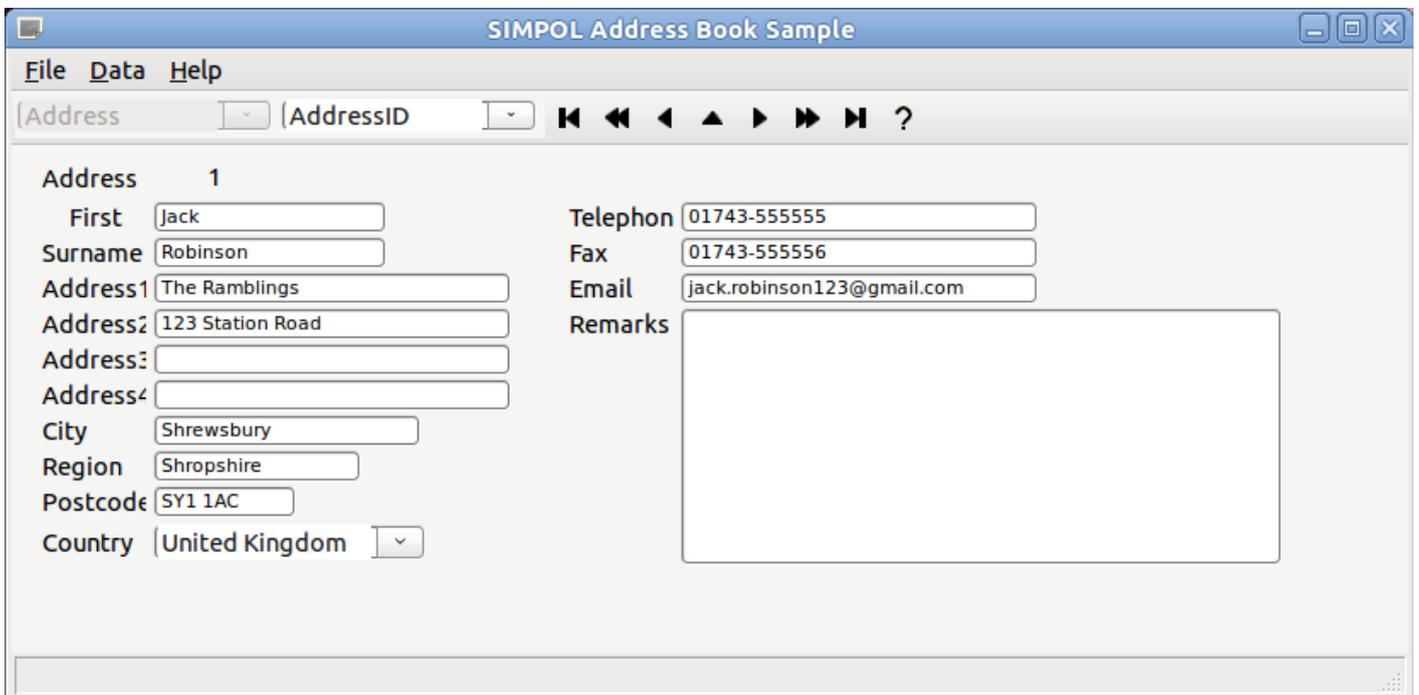
Field	Value
Address ID	1
First Names	Jack
Surname	Robinson
Address1	The Ramblings
Address2	123 Station Road
Address3	
Address4	
City	Shrewsbury
Region	Shropshire
Postcode	SY1 1AC
Country	United Kingdom
Telephone	01743-555555
Fax	01743-555556
Email	jack.robinson123@gmail.com
Remarks	

The finished Address Book application

As we can see, everything is now in place. Making changes to the application would be as simple as modifying the form or adding more forms and loading them based on button selections or menu or tool bar items.

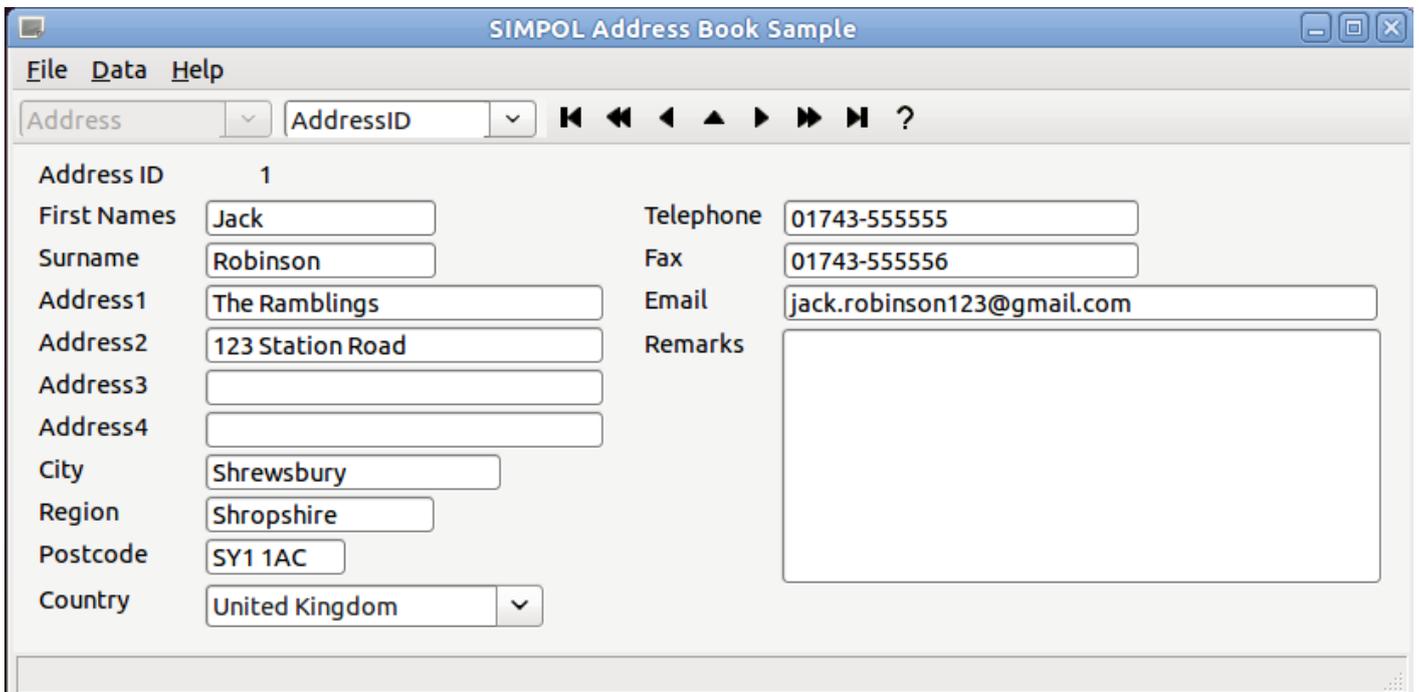
A Word About Linux

When moving an application to Linux, it is important to recognize that different fonts will be available. Even more important though, is that some Linux window managers will change the fonts to the current theme setting completely replacing the fonts that are used in the form. Also, they may remove or not support attributes like right alignment. As a result, if you intend to produce a version that runs on both platforms, you should either plan your fonts and design accordingly, or use two different forms, one for each platform and design each using the fonts and font sizes for that platform. For example, on Windows the form was designed with 8 point fonts. The font used was actually a placeholder name: "MS Shell Dlg 2", which will become different fonts on different version of Windows. On Ubuntu Linux, the default font was Ubuntu and the size was 11 points. The size, more than the font choice will impact the design, as seen below:



The finished Address Book application on Ubuntu Linux

The Ubuntu fonts can be installed on Windows as well, which makes the task of designing the form using the Form Designer much easier. After some basic adjustments, including modifying the code that creates the combo boxes in the tool bar to increase the font size if the OS is not Windows, the new result on Ubuntu Linux can be seen below:



The updated Address Book application on Ubuntu Linux

Summary

In this chapter we have built a database container, and a table to hold our data. We then built a form to allow us to perform easy data-entry. Finally, we wrapped the whole thing in some program

code that creates the window, the menu and tool bars, and which can run as a standalone program and act as the basis for a much larger and more complex system. Although we may choose to do some things slightly differently in a larger or in a networked program, we now have a sound foundation on which to build.

More importantly though, by using the supplied sample application together with the `appframework.sml` library, it is possible to quickly build a working prototype application with data-aware forms. The only areas that must be modified are:

- The name of the form being opened in function `main()`
- The name of the data source and table being opened in method `addressbookapplication.new()`
- The content of the function `ab_onnewrecord()` (if required – otherwise remove the call to the `preaddressbookform()` from function `main()`)

The following items should be changed as well for a long-term project:

- The name of the `addressbookapplication` type should correctly reflect the application being built.
- The `preaddressbookform()` and `ab_onnewrecord()` functions should be replaced with appropriately named functions.
- The `helpabout()` function should show the correct information, and could be replaced with a modal dialog.

Hopefully this chapter and the chapters up until now will have provided you with the tools that you need to make a fast start in the world of SIMPOL programming. Like in a good book, there is something to appreciate right at the start, but the more you investigate, the more there is to discover, if you wish to go there. Post your investigations and questions in the online forum and as a community we can go there together.

Advanced Topics

Now that you have gotten a basic single form and database table package running in single user mode, it is worth thinking about where to go from here. There are a couple of steps that come next:

- Loading other forms
- Changing to a multi-user system

Loading other forms turns out to be fairly easy. The basic call is `appw.openformdirect("myform.sxf")`. The framework takes care of the rest. It is a good idea to open all the required database tables during initialization of the application and ensure they are part of the `appwindow` object.

Changing to a multi-user system is a bit more complicated. First the database tables need to be opened and shared using a PPCS server. A sample server is included with Superbase NG. It is located in the `SIMPOL\Utilities\simplserver` directory. Also in that directory is a file called

`readme.txt`]. That file discusses everything necessary to share the database files via PPCS. Once the files are shared, the remaining change that is required is to open them using a PPCS data source.

```
if bUSEPPCS
  me.ppcs =@ ppcstype1.new(udpport=.nul, error=e)
  if me.ppcs =@= .nul
    wxmessagedialog(appw.w, "Error starting PPCS", sAPPMSGTITLE, \
      "ok", "error")
  else
    src =@ me.opendatasource("ppcstype1", "127.0.0.1:4000", appw, \
      error=e)
    if src =@= .nul
      wxmessagedialog(appw.w, "Error opening the PPCS server", \
        sAPPMSGTITLE, "ok", "error")
    end if
  end if
else
  src =@ me.opendatasource("sbme1", "address.sbm", appw, error=e)
  if src =@= .nul
    wxmessagedialog(appw.w, "Error opening the address.sbm file", \
      sAPPMSGTITLE, "ok", "error")
  end if
end if

if src !=@= .nul
  t =@ appw.opendatatable(src, "Address", error=e)
  if t =@= .nul
    wxmessagedialog(appw.w, "Error opening the 'Address' table", \
      sAPPMSGTITLE, "ok", "error")
  else
    me.address =@ t
    ok = .true
  end if
end if
```

The above program listing assumes that a boolean constant called `bUSEPPCS` has been defined earlier in the program. That is all that is required to switch the program to run as a multiple user system, other than a fully licensed database engine, though the 3-user license that is provided with Superbase NG for testing should be sufficient while doing development.

