Norman Lawrence

Digital Press

Compaq Visual Fortran

A Guide to Creating Windows Applications

OPERATING SYSTEMS

Compaq Visual Fortran

This Page Intentionally Left Blank

Compaq Visual Fortran

A Guide to Creating Windows® Applications

Norman Lawrence

Digital Press
An imprint of Butterworth-Heinemann
Boston • Oxford • Auckland • Johannesburg • Melbourne • New Delhi

Library of Congress Cataloging-in-Publication Data

.

.

British Library Cataloging-in-Publication Data

A catalogue record for this book is available from the British Library.

The publisher offers special discounts on bulk orders of this book.
For information, please contact:

Manager of Special Sales
Butterworth–Heinemann
225 Wildwood Avenue
Woburn, MA 01801-2041
Tel: 781-904-2500
Fax: 781-904-2620

For information on all Butterworth–Heinemann publications available, contact our World Wide Web home page at: http://www.bh.com.

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

To Albert, Lizzie, and Maria, and the memories of times filled with warm summer days and snow at Christmas.

This Page Intentionally Left Blank

Contents

This Page Intentionally Left Blank

Preface

Welcome to *Compaq Visual Fortran, A Guide to Creating Windows® Applications*. My computing experiences originated during the early 1970s when, as an undergraduate engineering student, I was introduced to FORTRAN IV complete with the joys of card readers, syntax errors, and an express four to six hour response time between job submission and output. Soon afterward, I was attracted by the immediacy of BASIC, and that language served my computing needs until the late 1980s, when I was involved in some research in which the client wanted software to be written in Fortran for use on the client's mainframe. I then saw Fortran in a new light; the power of portability meant that I could develop a program on a PC and send it to others to compile on their workstation or mainframe. The reverse was also true; I could use on my PC a vast range of software that had been developed for mainframes during a 30-year period.

Around 1994, I started to use Microsoft Visual BASIC 3.0 to provide a Windows graphical user interface (GUI) for my DOS-based Fortran programs. It was a revolutionary way of developing graphical user interfaces. However, my graphics requirements stretched Visual BASIC to the limit, and I often used the Windows Application Programming Interface to get some extra features. With the switch to a 32-bit operating system signaled by Windows 95 and Windows NT 3.5, Microsoft introduced the Fortran PowerStation Version 4.0 compiler, in which one could compile DOS Consol Window Projects, QuickWin Projects, and Windows Projects where the Win32 APIs were called directly. This program and its successors, Digital Visual Fortran 5.0 and 6.0 and now Compaq Visual Fortran 6.6, endowed programmers with the capability to create full-fledged Windows applications using only Fortran. The mixed language capabilities also meant that code segments written in Fortran and C could be compiled and linked into one single executable program.

Until now, one major drawback was the absence of a Fortran-specific book with information on how to develop graphical user interfaces in Fortran. Fortran programmers either had to read books written in C to learn about the Windows API or else buy proprietary software that provided interfaces to the API as a set of subroutines that could be called from Fortran. This book is aimed at Compaq Visual Fortran (CVF) users who want to develop applications with Windows-style graphical user interfaces by using the Win32 API interface definitions supplied with Visual Fortran. It illustrates, through numerous ready-to-run examples, how to develop QuickWin programs, explore the possibilities offered by the Win32 API, and create professional quality graphics using OpenGL. When creating the examples in this book, I worked with the belief that "if it can be done in C, it can also be done in Visual Fortran."

This is not a book on Fortran programming, so you will need to be comfortable with programming in FORTRAN 77 or, preferably, Fortran 90/95. You should also have some familiarity with the online Programmer's Guide that comes with Visual Fortran. To gain maximum advantage from using this book, you will need access to a standard version of CVF 6.6. Because most of the examples were originally created in CVF 6.1, users of previous versions of CVF should be able to use the techniques outlined. However, be aware that differences exist in the interface definitions and the symbolic constants between the previous editions of Visual Fortran and CVF 6.6. For example, the symbolic constant NULL_POINT is used in some applications, but users of CVF 6.1 will have to provide their own equivalent to NULL_POINT. In some places where NULL is used, CVF 6.1 users may need to substitute %Val(0).

Writing this book required me to make judgments on both the content covered and the amount of detail to be included. I have heard from readers of other computing books that they do not like to have details of the Win32 API functions reproduced in the text of a book because they already have access to that information online in Visual Studio. Therefore, I have tried to reach a balance in the material provided in this book by providing step-by-step information at the introductory level to topics and, as the topics progress, to include only the more salient points. I am interested in getting feedback through the publishers as to what readers would like to see included in any subsequent edition.

The source code for all the programs in this book can be downloaded from the companion Web site—the URL, http://www.bh.com/companions/1555582494/, takes you directly to the downloadable material. You

can also access it by going to www.digitalpressbooks.com (or www.bh.com/digitalpress). Find your way to Compaq Visual Fortran (by using the Search feature, or going to the page on Software Development by using the navigation buttons in the lower right of the screen). Once you reach the full page devoted to this book, click on the hot link "Provides downloadable supplementary materials from author" in the box at the right edge of the screen to go to the material.

This Page Intentionally Left Blank

Acknowledgments

On the shoulders of giants

Programming graphical user interfaces in any language can be very challenging, and my own understanding has been greatly enhanced over a number of years by various books, user groups, and individuals. In particular, I would like to acknowledge the following sources of information used during the creation of this book.

Appleman, D., 1996, *Visual BASIC, Programmer's Guide to the Win32 API*, Ziff-Davis Press.

Cluts, N., 1998, *Programming the Windows 95 User Interface,* Microsoft Press.

Etzel, M., and Dickinson, K., 1999, *Digital Visual Fortran Programmer's Guide*, Digital Press.

Jerke, N., and Brierly, E., 1996, *Visual BASIC 4 API How-To, the Definitive Guide to Using the Win32 API with Visual BASIC 4*, Waite Group Press.

Neider, J., Davis, T., and Woo, M., 1993, *OpenGl Programming Guide, The Official Guide to Learning OpenGL, Release 1*, Addison-Wesley.

Petzold, C., 1998, *Programming Windows* (5th ed.), Microsoft Press.

Simon, R., 1997, *Windows NT, Win32 API Superbible*, Waite Group Press.

Wright, R., and Sweet, M., 1996, *OpenGL Superbible, The Complete Guide to OpenGL Programming for Windows NT and Windows 95*, Waite Group Press.

I want to acknowledge Leo Treggiari, who showed me how to do C style pointers in Fortran; Steve Lionel, who convinced Digital Press that there was a need for this book and showed me how to use the OpenGL auxiliary library with Fortran Windows Application Projects; John Ready, who introduced me to OpenGL Utility Toolkit (GLUT) and showed me how to tessellate polygons; Bill Conrad, who gave me advice on QuickWin Projects; Guus Nijhuis, who showed me how to use text and mouse selection in OpenGL; Jakub Zlamal, who showed me how to work with toolbars; Adam Kris, who showed me how to put bitmaps and icons in buttons; and Bill Buchholz, who helped me figure out how to save and print OpenGL screens.

Information in this book relating to Win32 API and OpenGL functions is based on the information contained in the Platform SDK online documentation as provided by Microsoft with Visual Studio 6.0.

Information relating to the creation of Help files is based on the online Help documentation for the Win32 Help Compiler Workshop as provided by Microsoft with Visual Studio 6.0.

My thanks

I thank Digital Press and Compaq Computer Corporation for their many contributions to the creation of this book. In particular, I thank Compaq for supplying various beta versions, and Christian Staudinger for providing a release version of Visual Fortran 6.6.

Reviewing draft manuscripts of a book is a big task, and I would like to thank the CVF team for the thoroughness of their comments on all the draft chapters and especially Bill Conrad, Leo Treggiari, Lorri Menard, John Ready, and Steve Lionel. Thanks to Bill Buchholz, John Termine, Jean Vezina, Matt Allen, Ozgur Deli, and Tim Hatamian for their various comments on the draft manuscript.

My thanks to Pam Chester for providing me with the opportunity to write this book and for all her positive and encouraging comments during the entire period of writing it, even when progress was way behind my overly optimistic schedule.

My deepest thanks to Mike Etzel, a very special person, for his incredible help, wisdom, and support during the entire period of writing this book.

I would like to thank my wife Cathy for her unfailing support during both the writing of this book and our 26 years of married life together. A

very special thanks to my children, Colin, Chris, and Maree, who patiently watched Dad's show and tell sessions for every program and then would ask simple questions like, "What happens if?" Their lateral thinking and bug-discovering skills are incredible. A big thanks to my brother Al and sister Susan for always being around whenever I needed them.

This Page Intentionally Left Blank

1

Getting Started

1.1 A look at Developer Studio

Compaq Visual Fortran (Visual Fortran) uses the same development environment as Microsoft Visual C++. This development environment is shown in Figure 1.1, and it is known as the Microsoft Developer Studio. In Figure 1.1, the left pane has tabs to select a FileView, a ResourceView, and a ClassView. The FileView shows each project and the project files associated with the Workspace. The ResourceView appears if the Workspace uses Resources such as dialogs and icons. The ClassView appears only if Visual C++ is installed. ClassView is not used by Visual Fortran.

The FileView pane shown in Figure 1.1 contains the files associated with the Workspace for the project Plots. The Source Files folder indicates that there are six files in this project workspace: Callbacks.f90, Globals.f90, Main.f90, Menu.f90, Plot.f90, and Resource.rc. The Resource Files folder contains one bitmap and three icons, while the External Dependencies folder lists the files global.mod and resource.fd. Double-click on any of the file icons in the FileView pane to open that file with the Developer Studio text editor in the right-hand pane.

In the right-hand pane of Figure 1.1, the file Main.f90 has been opened in the default text editor. This editor uses green to identify comments in the code, blue for Fortran standard code, and black for other text. There are actually 12 different "colorable items" (including comments and Fortran keywords), and the default color of each item can be changed in Developer Studio by using the Tools menu, selecting the Options menu item, and then choosing the tab labeled Format (Tools -> Options -> Format).

The bottom of Figure 1.1 shows the output pane, which shows text displayed from building the project. Currently it is displaying information about compiler and link errors. The output pane has multiple tabs. The

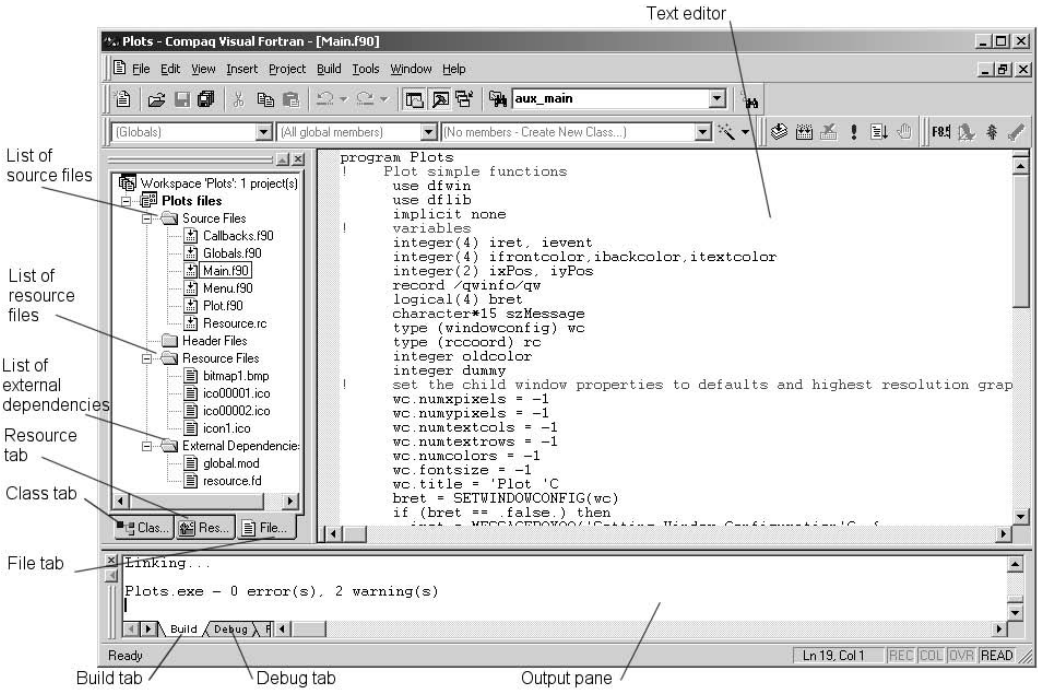

Figure 1.1 *Developer Studio window.*

output pane Build tab is selected after you open a Workspace and compile one or more project files or build the project.

Windows in the Visual Development Studio can be set to have either docking or floating properties. Docking windows do not have a title bar, and they are docked (attached) along an edge of the visual development environment window. Floating windows have a title bar and can be moved by dragging the title bar. By holding down the Ctrl key and pressing the Tab key, you can cycle through each floating window in turn. The docking or floating properties of a window are set by selecting Options from the Tools menu. Click the Workspace tab in the docking view list and click the check box for each window to be displayed with a docking view (see Figure 1.2). Unchecked boxes indicate floating windows.

1.2 Visual Fortran project types

Projects are contained in a workspace and consist of the source files required for an application, along with the specifications for building the project. A project type must be chosen every time you create a new project. The

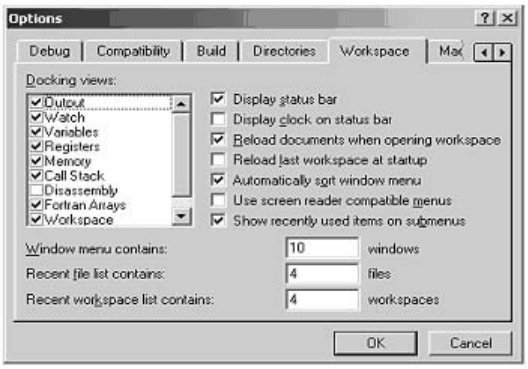

Figure 1.2 *Setting the docking properties for windows in the Developer Studio.*

project type determines some of the options that Developer Studio will set by default for the project and what needs to be generated. It determines, for instance, the options that the compiler uses to compile the source files, the static libraries that the linker uses to build the project, the default locations for output files, and defined constants.

The following seven project types are available with Visual Fortran:

- *Fortran Console Applications*—These are single-window applications without graphics (character-cell applications), used for a traditional Fortran program. Standard Fortran I/O statements are used to write to the window.
- *Fortran Standard Graphics Applications*—Standard Graphics Applications have a single maximized window covering the entire screen. Standard Graphic Applications do not have programmable menus, but they can use QuickWin graphic routines to draw to the window. Standard Fortran I/O statements are used to write to the window.
- *Fortran QuickWin Applications*—QuickWin Applications can have a range of graphics capabilities and receive user input through menu selections, dialog boxes, and mouse button actions. QuickWin Applications can have multiple windows. Standard Fortran I/O statements are used to write to the window.
- *Fortran Windows Applications*—These are single- or multiple-window projects with a full graphical interface, which can include menus, dialog boxes, and graphical routines and can use any of the available Win32 API routines. In Fortran Windows Applications, the programmer directly calls any Win32 routines that are required. Win32 routines are used to write to the window.

- *Fortran COM Server*—Uses the component object model (COM) to implement a COM server. COM supports client/server interaction between a user of an object, the client, and the server. Clients may be written in Visual Fortran using the Fortran Module Wizard or in other languages, such as Visual C++ and Visual BASIC.
- *Fortran Static Library*—These are library routines that encapsulate some specific functionality. They are linked into .exe files.
- *Fortran Dynamic-Link Library*—These are library routines that encapsulate some specific functionality. They are linked into .DLL files and are used by an .exe file during execution.

1.3 Developing graphical user interfaces

The focus of this book is on using Visual Fortran Version 6.6 to create Fortran applications that have a Windows-based graphical user interface (GUI). The traditional character-cell console applications, including those written in Fortran, are procedural in nature. That is, execution begins with the first line of executable code and then sequentially follows a defined pathway through the application, calling functions and subroutines as required until it reaches the end of the program. Windows-based GUI programs are event-driven in nature. The order in which the code executes depends on which events have been chosen. An event occurs every time the user makes a selection from a range of choices. It could be to open or close a file, to enter data, or to copy the contents of the screen to the clipboard.

GUI applications should always present the user with a valid set of options, respond according to the selections made by the user, and ensure that the user has responded correctly to any input information that the program requested. This may mean that some options are disabled, or it may entail checking that a text box has valid input before proceeding with the next operation.

1.3.1 Graphical user interface etiquette

It is reasonable to assume that most, if not all, of the people who use your applications will have previously used Windows-based applications such as word processors and spreadsheets. This being the case, your application users will have certain expectations of your GUI because applications in the Windows environment share a common "feel" in their user interfaces. Referring to Figure 1.3, users will expect File and Help to be the first and