

Microsoft® Quick Pascal

Up and Running

Microsoft®

pcjs.org

Brief Guide to Keystrokes

ALT	Activates the menu bar. Press the highlighted characters in the menu titles to open menus and to choose commands from a menu. Or use the LEFT and RIGHT direction keys to highlight a menu title, and then press UP and DOWN to choose a command. Pressing ALT a second time closes all open menus and deactivates the menu bar.
ESC	Closes menus and dialog boxes (canceling any entries). ESC also stops the compiler from continuing.
F1	Invokes the on-line help system. To get help, position the cursor on the topic about which you want more information: a keyword, a menu, a command, a dialog box, or an error message. Then press F1. To close the Help window, press CTRL+F4.
ALT+ F1	Displays the previous Help window. QuickPascal remembers the last 20 help topics.
SHIFT+ F1	Explains the on-line help system in more detail.
F4	Toggles the visible screen between the QuickPascal environment and the output screen.
F5	Compiles and runs the currently active main program.
F6	Moves the cursor between visible windows.

Microsoft® QuickPascal

UP AND RUNNING

MICROSOFT CORPORATION

pcjs.org

Information in this document is subject to change without notice and does not represent a commitment on the part of Microsoft Corporation. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or nondisclosure agreement. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of Microsoft.

©Copyright Microsoft Corporation, 1989. All rights reserved.

Printed and bound in the United States of America.

Microsoft, MS, MS-DOS, and QuickC are registered trademarks of Microsoft Corporation.

Borland and Turbo Pascal are registered trademarks of Borland International, Inc.

BRIEF is a registered trademark of UnderWare, Inc.

Epsilon is a trademark of Lugaru Software, Inc.

Hercules is a registered trademark of Hercules Computer Technology.

IBM is a registered trademark of International Business Machines Corporation.

WordStar is a registered trademark of MicroPro International Corporation.

Document No. LN0121-500-R00-0489

Part No. 06395

10 9 8 7 6 5 4 3 2 1

Table of Contents

iii

Introduction

vii

Performance and Compatibility	vii
Read This Manual First	viii
Key to Document Conventions	ix

Chapter 1 Unpacking QuickPascal

1

System Requirements	1
The QuickPascal Package	2

Chapter 2 Installing QuickPascal

3

A Quick Overview	3
Before You Begin	4
Running SETUP	5
Installing on a Floppy-Disk System	14

Chapter 3 Using QuickPascal

19

The QuickPascal Environment	19
Using Windows and Menus	20
Using the Editor	30
Compiling Your Program	33
The Debugger	43

iv Up and Running

Chapter 4 Getting Help	45
The QP Advisor	45
Keyword Help	46
Topic-Based Help	51
Environment Help	56
Error Help	57
Help On Help	59
Chapter 5 Where to Go from Here	61
Read README.DOC	61
For All Programmers	61
For New Programmers	62
For Pascal Programmers New to QuickPascal	62

Figures and Tables

v

Figures

Figure 2.1	SETUP's Introductory Screen	5
Figure 2.2	SETUP's Main Menu	6
Figure 2.3	The Easy Setup Menu	7
Figure 2.4	Easy Setup Installation Screen	8
Figure 2.5	Insufficient Space Warning	9
Figure 2.6	The Full Setup Menu	10
Figure 2.7	The First Set Paths/Directories Screen	11
Figure 2.8	Command Files Supplied with Quick Pascal	12
Figure 2.9	Full Setup Installation Screen	13
Figure 2.10	Insufficient Space Warning	13
Figure 2.11	The .EXE File Path Ready for Floppy Installation	15
Figure 2.12	Show Options with Floppy Paths Installed	15
Figure 2.13	Installing QuickPascal on Floppies	16
Figure 3.1	Parts of a Window	22
Figure 3.2	Display Dialog Box	29
Figure 3.3	Save As... Dialog Box	36
Figure 3.4	Setting the Main File	40
Figure 4.1	Help on Writeln	47
Figure 4.2	Writeln Details Screen	49
Figure 4.3	QP Advisor Contents Screen	52
Figure 4.4	Crt Unit Topics	53

vi Up and Running

Figure 4.5	GotoXY Summary Screen	54
Figure 4.6	Details of GotoXY	55
Figure 4.7	Help on the Merge Command	56
Figure 4.8	Error Dialog Box	57
Figure 4.9	Help on an Error Message	58

Tables

Table 3.1	Parts of a Window	23
-----------	-----------------------------	----

Welcome to Microsoft® QuickPascal, a powerful and sophisticated yet easy-to-use integrated environment for writing programs in the Pascal language.

Many people—beginners and experienced programmers alike—find Pascal an ideal programming language because it combines the best features of many languages. Like BASIC, Pascal uses many intuitive keywords. The **Read** statement, for example, reads information into a program. Like C, Pascal ensures adherence to structured programming techniques. And Pascal executable files are portable between most personal computers. For these and other reasons, many programmers prefer Pascal to any other language.

Microsoft QuickPascal combines the power of Pascal with an environment that makes Pascal easy to learn and to use. You can write code, compile it, run the program, and debug it—all in the QuickPascal environment.

Performance and Compatibility

QuickPascal offers the best features of other Pascal compilers, plus several unique advantages. In particular, QuickPascal provides

- An integrated, multiwindow interface. The QuickPascal environment supports up to nine source windows open simultaneously, each of which can be independently sized and positioned.
- A powerful on-line help system, the QuickPascal Advisor. The QuickPascal Advisor is a source for language reference, a tutor on the programming environment, and a guide to error messages. It even includes sample programs that you can copy and paste directly into your own programs.
- Graphics. The graphics units provided with QuickPascal are compatible with Microsoft QuickC® 2.0 and Borland® Graphics Interface (BGI).
- Object-oriented programming extensions. QuickPascal supports extensions used in object-oriented programming.

- Excellent debugging capabilities. QuickPascal shows where an error occurred and provides several methods for finding errors in a program's logic.
- Syntax compatibility with Turbo Pascal ® Versions 4.0 and 5.0. QuickPascal can compile most Turbo Pascal programs without modification.
- A self-paced introduction to the QuickPascal programming environment. The QP Express computer-based training program introduces QuickPascal's programming and debugging environment.
- An introduction to the Microsoft family of Quick language products. Since all Microsoft Quick language products use a similar environment, you'll be able to pick up another Quick language with ease.

NOTE Microsoft documentation uses the term "DOS" to refer to both the MS-DOS® and IBM® Personal Computer DOS operating systems. The name of a specific operating system is used when it is necessary to note features that are unique to that system.

Read This Manual First

This manual contains all the information you need to install and begin using QuickPascal on your computer. There are five chapters:

Unpacking QuickPascal Chapter 1 lists the system requirements (hardware that you provide) and the contents of this package (software and documentation that Microsoft provides). Read this chapter before you proceed.

Installing QuickPascal Before you can start using QuickPascal, you must install it by running SETUP. Chapter 2 guides you through installation and provides answers to commonly asked questions.

Using QuickPascal Chapter 3 explains how to run QuickPascal and introduces the window and menu environment. Next you are stepped through the writing, compiling, and running of two short sample programs.

Getting Help The Microsoft QuickPascal Advisor (on-line help) provides important reference information at the click of a button or press of a key. Chapter 4 illustrates the many facets of this powerful help system.

Where to Go from Here Once you unpack QuickPascal, install it on your system, and compile a sample program, you will probably want to investigate QuickPascal further. Chapter 5 provides some suggestions on what to do next.

Key to Document Conventions

This book uses the following document conventions:

<u>Example</u>	<u>Description</u>
SETUP.EXE, COPY	Uppercase type represents DOS commands and file names.
BEGIN, Writeln	Boldface type (whether in all uppercase or in both upper- and lowercase letters) indicates standard features of the QuickPascal language: keywords, operators, and standard procedures and functions.
Topic: <code>Writeln</code>	This typeface is used for example programs, program fragments, and the names of user-defined functions and variables. It also indicates user input and screen output.
F1, ESC, CTRL+ENTER	Small capital letters denote names of keys on the keyboard. A plus (+) indicates a combination of keys. For example, SHIFT+F5 tells you to hold down the SHIFT key while pressing the F5 key.
ENTER	The carriage-return key, sometimes appearing as a bent arrow on the keytop is called ENTER.
PGUP	The cursor-movement keys (sometimes called “arrow” keys) are called the DIRECTION keys. Individual keys are referred to by their direction (LEFT, UP) or by the name on the key (PGUP).
“hyperlinks”	The first time a new term is defined, it is enclosed in quotation marks. Since some knowledge of programming is assumed, common terms such as memory or branch are not defined.
kilobytes (K)	An acronym is spelled out the first time it appears.

Unpacking QuickPascal

You're probably eager to install QuickPascal immediately. But first take a few minutes to make sure your system meets some minimum requirements and to determine that your package is complete.

System Requirements

QuickPascal requires the following minimum configuration:

- An IBM Personal Computer or compatible running DOS Version 2.1 or later.
- One hard-disk drive and one floppy-disk drive or two floppy-disk drives. A hard-disk drive is strongly recommended.
- 448 kilobytes (K) of available memory (512K is recommended for medium to large projects).

The QuickPascal Package

Check your QuickPascal package to see if everything is there. If any pieces are missing, contact the retailer from whom you bought QuickPascal. In the package, you should find the following items:



Registration card: there are many advantages to being a registered owner of QuickPascal, including notification of future software releases and easy access to customer assistance. Please take the time to fill out and mail the registration card now.



Disks: five 5 1/4-inch floppy disks or three 3 1/2-inch disks. The distribution disk labeled Setup/Utilities (or Setup/Program/Utilities) contains a file named PACKING.LST that lists the location and description of all disk files in the Microsoft QuickPascal package.



Up and Running: the book you're reading now. It explains how to install and use QuickPascal.



Pascal by Example: this book comes with the retail version of QuickPascal and is written for beginning programmers and programmers new to Pascal. It covers all the topics necessary to become proficient with Pascal, and contains a Quick Reference Guide in the back.

Installing QuickPascal

This chapter tells you how to install QuickPascal on your system. The program SETUP.EXE on the Setup/Utilities disk does this for you. It creates new directories and copies several programs—the compiler, the help system, and others—from the distribution disks to your hard disk (or floppy disks).

If you follow the instructions, when you finish this chapter you'll have a working version of QuickPascal on your system. You will be ready to start programming.

A Quick Overview

There are three steps to installing QuickPascal:

1. Make backup copies of all distribution disks.
2. Read the first section of the README.DOC file for information on installing QuickPascal. If there are any corrections made to this book, they're listed in README.DOC.
3. Run SETUP.EXE, which is an interactive program. If you prefer, you can install QuickPascal quickly by accepting the default settings used by the Easy Setup option. These default settings work for nearly all systems. The Full Setup option allows you to alter the default file locations; it is recommended only if you are experienced with DOS. Remember that you can always change the file locations later with DOS commands.

Each process is straightforward and the screens provide helpful guidance. If you have enough DOS programming experience to complete these three steps, feel free to do so after you read the next section. (If you want further assistance, return to this chapter.) You may then skip ahead to Chapter 3, “Using QuickPascal.”

NOTE *If you make a mistake during the setup process, just run the program again. SETUP.EXE never erases files from the distribution disks.*

Before You Begin

Before you run SETUP, back up the QuickPascal disks using the COPY command from DOS or the DISKCOPY program. Insert the purple-labeled Setup/Utilities distribution disk and change to that drive. You can read the first part of the file README.DOC by loading it into a word processor or by using the TYPE command at the DOS prompt:

```
TYPE README.DOC | MORE
```

The first page of README.DOC appears on your screen. Check for any changes to the SETUP instructions. To see the next page of text, press the SPACEBAR. When you finish, press CTRL+C to return to the DOS command line.

NOTE *The following instructions assume that you are installing QuickPascal on a system that has at least one floppy-disk drive and one hard-disk drive. If you don't use a hard drive, you must have at least two floppy drives. Please read the section "Installing on a Floppy-Disk System" at the end of this chapter if you plan to install QuickPascal on floppy disks.*

Correcting Mistakes

If you run the Easy Setup option you can review the default settings, but you will not be asked to enter any specific information. Short of unplugging the computer, you cannot make any mistakes.

Under Full Setup each screen lets you exit the program or return to a menu from which you can exit the program. If you exit without explicitly telling SETUP to install QuickPascal, no changes occur. Your disks are not modified in any way and QuickPascal remains on the distribution disks.

If you make an error in Full Setup prior to installing QuickPascal, simply re-choose the option where you made the error and correct the entry. If you catch an error after installing QuickPascal, just run SETUP again.

*R*unning **SETUP**

When you're ready to install QuickPascal, find the purple-labeled Setup/Utilities distribution disk and insert it in drive A (if you haven't already done so). Change to the disk in drive A (type `A:`). At the DOS command line, type

`SETUP`

The program starts and the introductory screen tells you what SETUP does and how much space QuickPascal requires on your hard disk (see Figure 2.1).

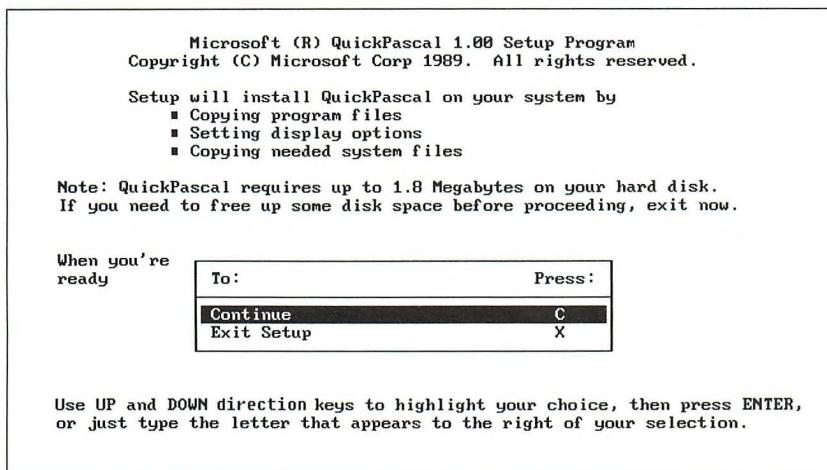


Figure 2.1 SETUP's Introductory Screen

Press C to continue. SETUP's main menu now appears, as shown in Figure 2.2.

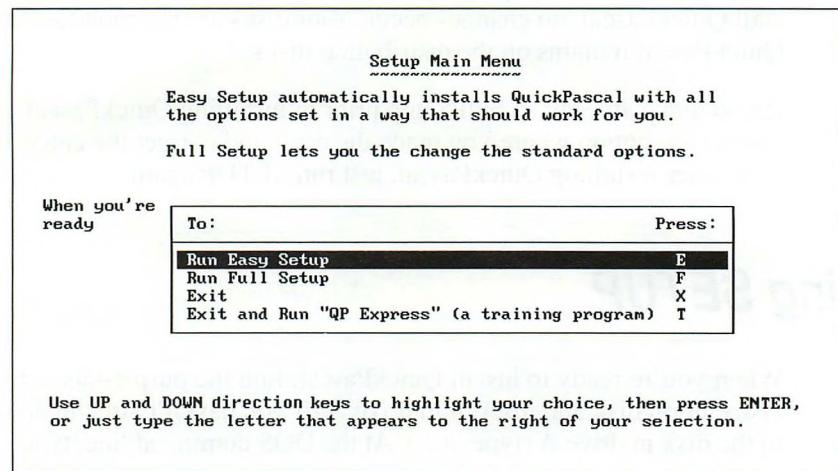


Figure 2.2 SETUP's Main Menu

From here you have several choices: Easy Setup, Full Setup, Exit, or Exit and Run QP Express. Easy Setup creates a new directory (called QP) on your hard disk and installs QuickPascal's subdirectory and files within that main directory. Full Setup also installs all of QuickPascal's files, but allows you to change the name and location of the directories (the "paths" to the files). Unless you have a particular reason for altering the default paths, choose Easy Setup by pressing E. You can also exit the program to the DOS command line or exit and run QuickPascal's self-paced tutorial, QP Express.

The following sections describe Easy Setup and Full Setup in detail; read the one that applies to the option you choose.

Easy Setup

Once you choose Easy Setup from the main menu, the screen changes to the Easy Setup menu shown in Figure 2.3.

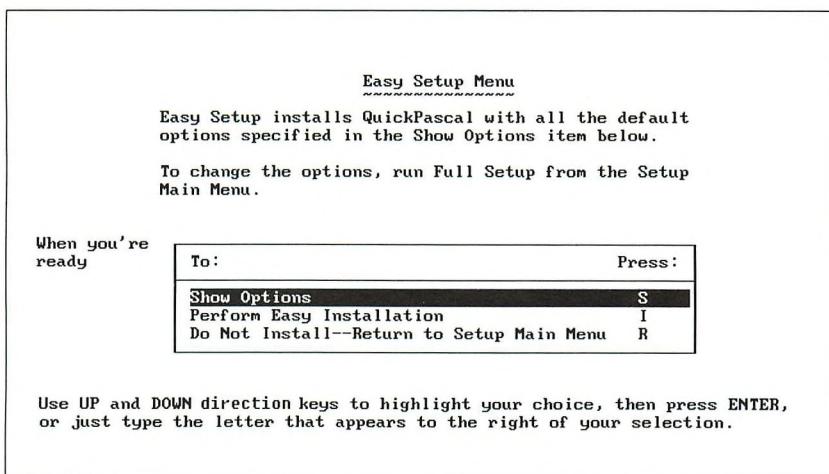


Figure 2.3 The Easy Setup Menu

The Show Options command lists the directories where SETUP will place the various QuickPascal files. The default paths for the file types are:

<u>Path</u>	<u>File Type</u>
C:\QP	Executable (.EXE)
C:\QP\DOC	Document (.DOC)
C:\QP	Unit (.QPU)
C:\QP	Help (.HLP)

Press the PGDN key to display the second page. It shows two command files supplied with QuickPascal, MOUSE.COM and MSHERC.COM. SETUP installs

1. MOUSE.COM, ensuring that QuickPascal can use a Microsoft mouse (or compatible mouse) if one is available.
2. MSHERC.COM, if it detects a Hercules® graphics adapter. SETUP enables Hercules and compatible graphics cards for use by the graphics routines. If it does not detect a Hercules card, it does not install MSHERC.COM.

If you decide to change any of the default file locations, you must return to the main menu and run Full Setup. To continue with the installation, press I for Perform Easy Installation. You'll see the installation screen in Figure 2.4. (Note that if you use 3 1/2-inch disks, the screen reads "Page 1 of 3," and asks for the Setup/Program/Utilities disk.)

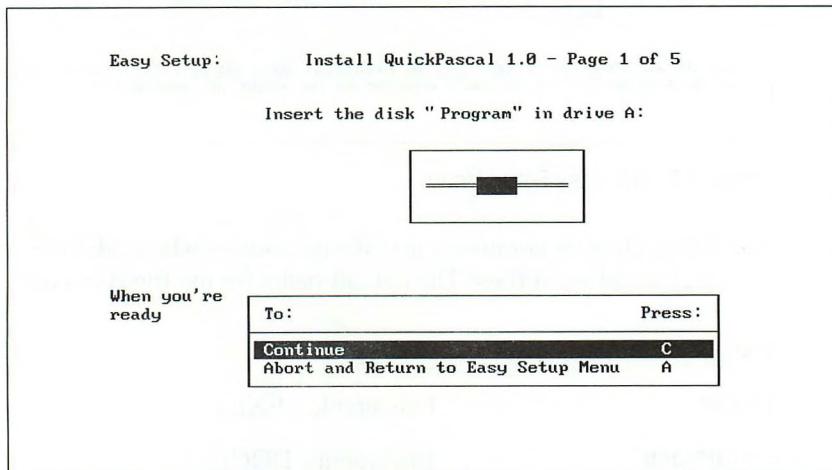


Figure 2.4 Easy Setup Installation Screen

SETUP now asks you to insert the first of the distribution disks. Insert the Program disk in drive A and press C to continue. As SETUP finishes working with each disk, it asks you for the next one. Remove the current disk from drive A, insert the requested disk, and press C to continue.

If your hard disk is nearly full, it's possible that QuickPascal cannot be installed. If SETUP finds insufficient disk space on drive C, it displays the warning shown in Figure 2.5. If this occurs, remove some files from your hard disk and rerun SETUP.

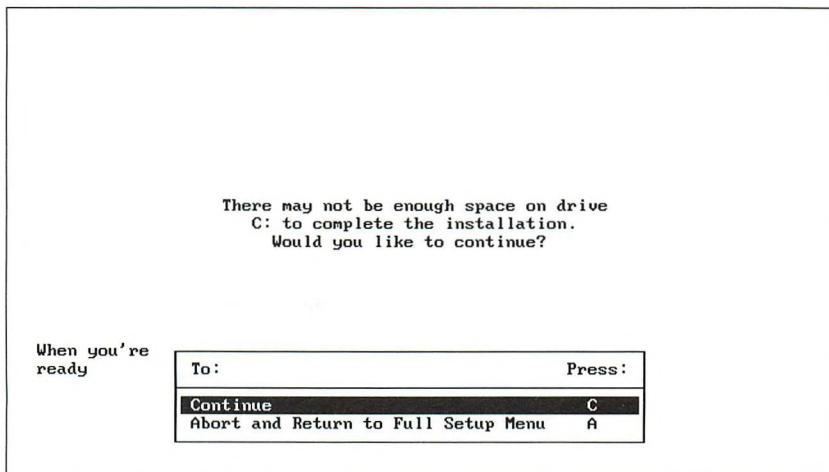


Figure 2.5 Insufficient Space Warning

Once SETUP completes the installation, it displays the message

QuickPascal Successfully Installed

Microsoft QuickPascal is now installed on drive C.

Next a menu appears showing two options, Exit and Run QP Express (T) and Return to Setup Main Menu (R). If you press T, SETUP runs LEARN (the QP Express). If you press R, SETUP returns to the main menu.

Now you are ready to take a break, run the QP Express tutorial, or begin your first QuickPascal programming session. To work with QuickPascal directly, go on to Chapter 3, “Using QuickPascal.” It introduces the QuickPascal environment and works through several example programs.

Full Setup

When you choose Full Setup from the main menu, the screen changes to the one shown in Figure 2.6.

Choose any of the following options from the Full Setup menu:

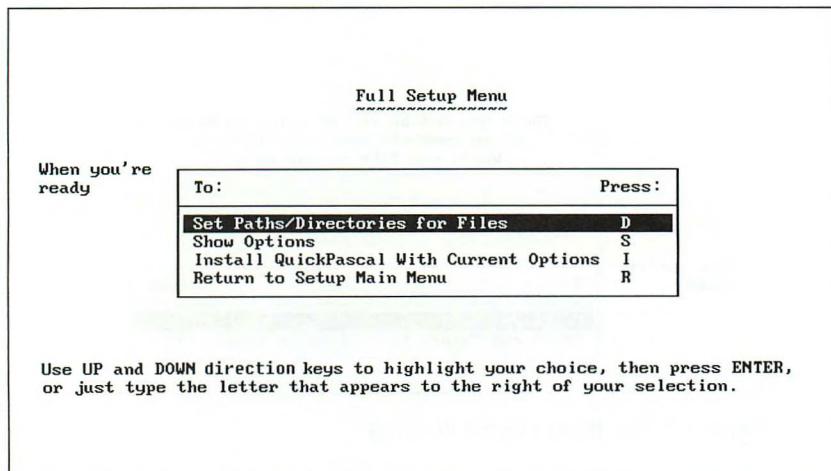


Figure 2.6 The Full Setup Menu

<u>Option</u>	<u>Description</u>
Set Paths/Directories for Files	Sets the paths for the different QuickPascal file types
Show Options	Reviews the current path settings
Install QuickPascal With Current Options	Installs QuickPascal using the paths shown under Show Options
Return to Setup Main Menu	Returns to the main setup menu

QuickPascal's default paths for the different file types are:

<u>Path</u>	<u>File Type</u>
C:\QP	Executable (.EXE)
C:\QPDOC	Document (.DOC)
C:\QP	Unit (.QPU)
C:\QP	Help (.HLP)

To alter QuickPascal's default paths, choose Set Paths/Directories for Files by pressing D. The next screen has a text box with the default path for executable files (see Figure 2.7). Use the TAB key to move the cursor into the text box. Delete the old path and type the new one. Press the TAB key again to shift back to the dialog box.

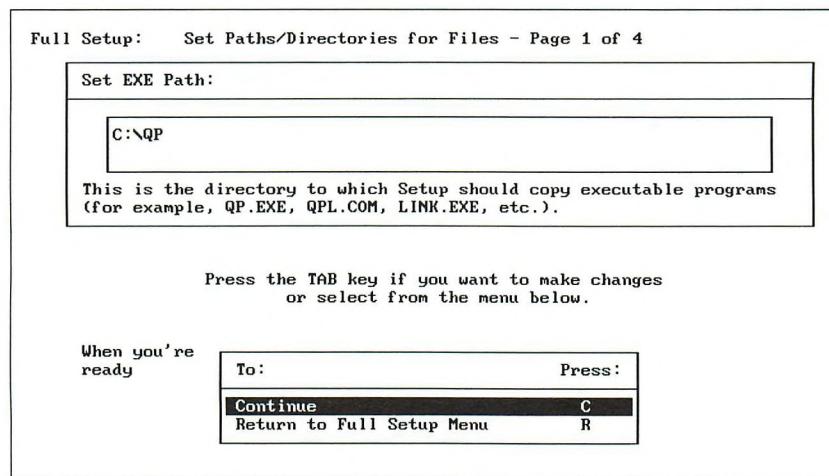


Figure 2.7 The First Set Paths/Directories Screen

Accept the path entered in the text box by continuing to the next screen (press C to continue). Repeat this process for each of the next three screens; they set the paths

for document (.DOC) files, unit (.QPU) files, and help (.HLP) files. At any point you can return to the previous screen or to the Full Setup menu.

After you finish with Set Paths/Directories and return to the main Full Setup menu, verify the new paths with the Show Options menu. It displays two pages, the first reviewing the paths you have just set. Press C to continue to the second page. It shows two command files supplied with QuickPascal, MOUSE.COM and MSHERC.COM (see Figure 2.8).

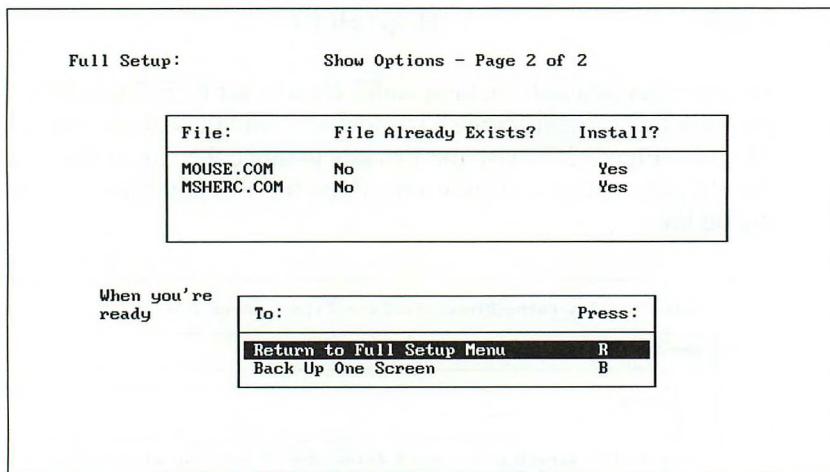


Figure 2.8 Command Files Supplied with QuickPascal

SETUP installs

1. MOUSE.COM, ensuring that QuickPascal can use a Microsoft mouse (or compatible mouse) if one is available.
2. MSHERC.COM, if it detects a Hercules graphics adapter. Running SETUP lets the graphics procedures use Hercules and compatible graphics cards. If it does not detect a Hercules card, it does not install MSHERC.COM.

Press R to return to the main Full Setup menu. If you want to modify any of the paths further, repeat the Set Paths/Directories procedure. When the new paths are correct, go ahead and install QuickPascal. Choose Install QuickPascal With Current Options by pressing I. The installation screen appears and asks you to begin

inserting the distribution disks as shown in Figure 2.9. (Note that if you use 3 1/2-inch disks, the screen reads “Page 1 of 3,” and asks for the Setup/Program/Utilities Disk.) Insert the disks as requested, pressing C to continue after each disk exchange. You can abort the installation at any time, if necessary.

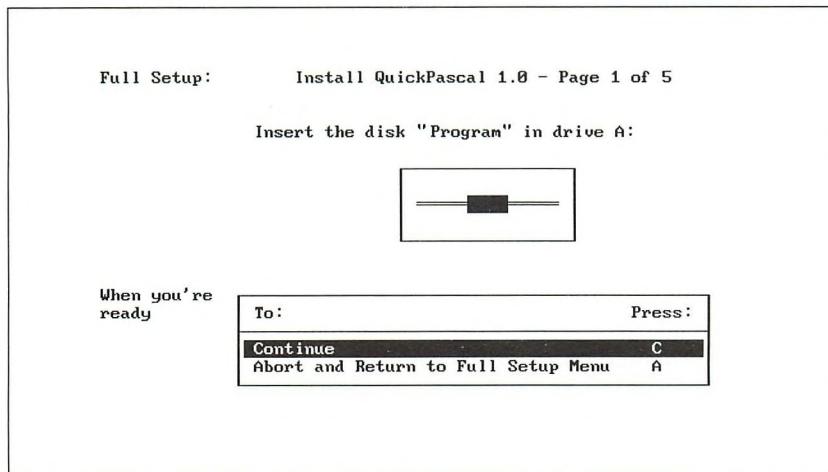


Figure 2.9 Full Setup Installation Screen

If SETUP detects insufficient disk space to install QuickPascal on drive C, it displays the screen shown in Figure 2.10.

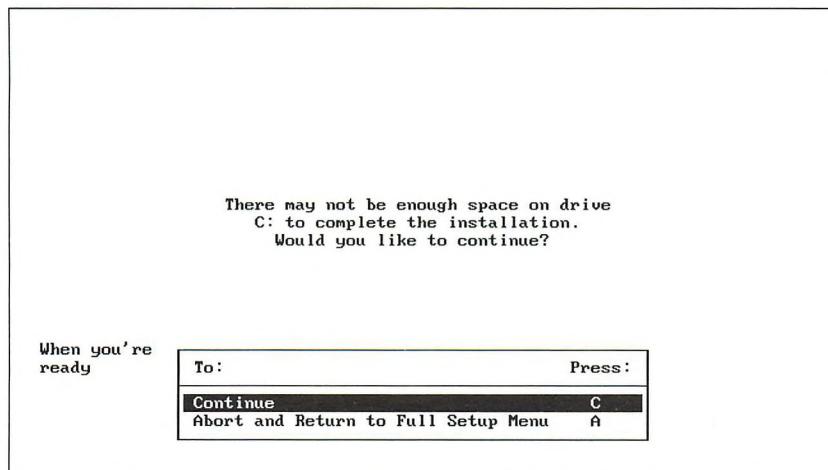


Figure 2.10 Insufficient Space Warning

If this occurs, free some space on the hard disk by removing files and rerun SETUP.

When SETUP completes the installation, a menu appears asking you to either Exit and Run QP Express or Return to Setup Main Menu. Either you can run the tutorial program QP Express, or exit the main menu and continue with Chapter 3, "Using QuickPascal." Chapter 3 discusses the QuickPascal environment and steps you through a couple of sample programming sessions.

Installing on a Floppy-Disk System

The procedure for installing QuickPascal on a system with two floppy-disk drives is similar to the hard-drive installation procedure.

Before you begin the SETUP program for a floppy-disk system, you'll need to have ready five blank 5 1/4-inch formatted disks (360K each) or three blank 3 1/2-inch formatted disks (720K each). Number the disks sequentially, starting at 1.

Review the procedure for installing QuickPascal on a hard disk with the Full Setup option described in the "Running Setup" and "Full Setup" sections. You follow a similar process, except that you install on drive B instead of drive C.

Begin the installation process by inserting the purple-labeled Setup/Utilities distribution disk in drive A and changing to that drive (type A:). At the DOS command line, type

```
SETUP
```

The program starts and the introductory screen tells you what SETUP does and how much space it will require (see Figure 2.1). Press C to continue.

The main setup menu appears (see Figure 2.2). The Easy Setup option assumes you use a hard disk; Full Setup also assumes you use a hard disk but allows you to change the paths of the various types of files. You can also exit SETUP, or exit and run the QP Express tutorial program.

Press F to choose Full Setup. The screen changes to the Full Setup menu (see Figure 2.6). To set the new paths, press D for Set Paths/Directories for Files.

The first of four path-setting screens appears (see Figure 2.7). This one is for the path to executable (.EXE) files. Press the TAB key once to move the cursor to the text box. Erase the old path by pressing the BACKSPACE key as many times as necessary. Next, type

```
B:\
```

The screen now looks like Figure 2.11.

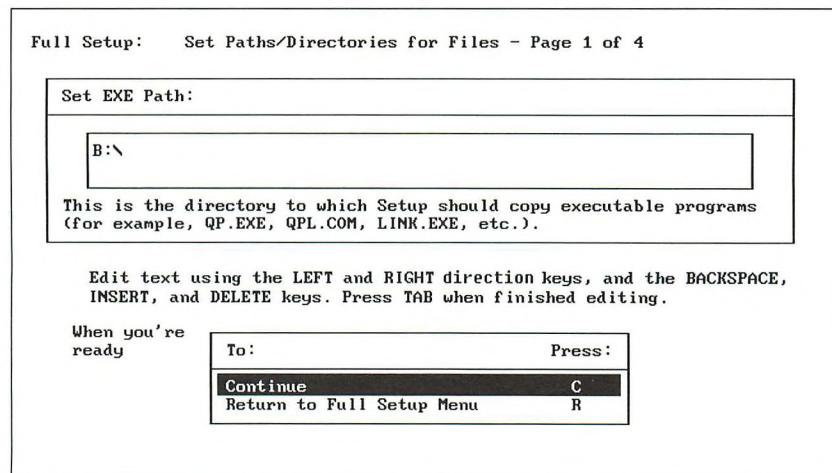


Figure 2.11 The .EXE File Path Ready for Floppy Installation

Press the TAB key again to highlight the Continue option. Press C to continue to the next screen, the paths for document (.DOC) files. Repeat the same process as before, changing the path to B :\ . Then do the same for the paths to the unit (.QPU) files and help (.HLP) files. When you choose Continue on the last screen, SETUP returns to the Full Setup menu.

Review the new settings by pressing S for Show Options (Figure 2.12):

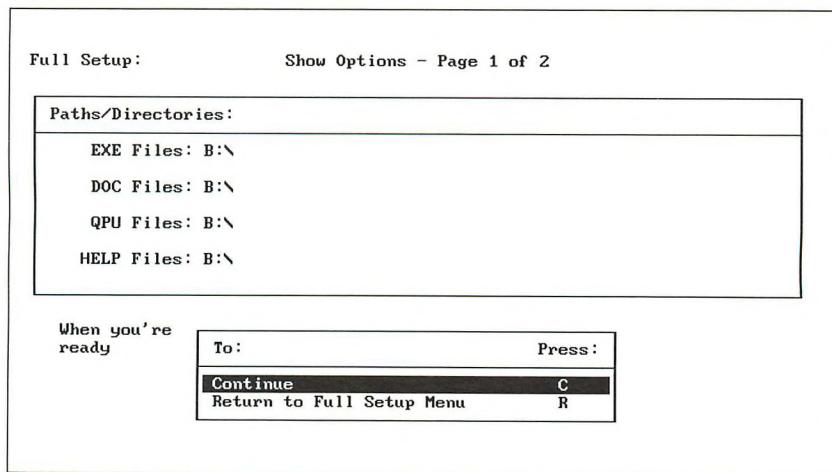


Figure 2.12 Show Options with Floppy Paths Installed

Press C to continue to the second page. It shows two command files supplied with QuickPascal, MOUSE.COM and MSHERC.COM. SETUP installs

1. MOUSE.COM, ensuring that QuickPascal can use a Microsoft mouse (or compatible mouse) if one is available.
2. MSHERC.COM, if it detects a Hercules adapter. SETUP enables Hercules and compatible graphics cards for use by the graphics routines. If it does not detect a Hercules card, it does not install MSHERC.COM.

Now you're ready to perform the actual installation. Choose Install QuickPascal With Current Options by pressing I. SETUP displays the screen in Figure 2.13.

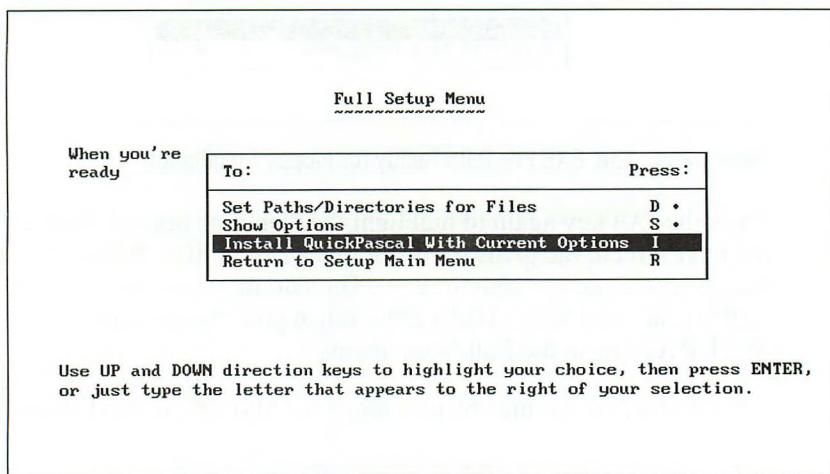


Figure 2.13 Installing QuickPascal on Floppies

Insert the requested disk in drive A and insert the first blank disk in drive B. Press C to continue. SETUP copies the contents of the distribution disk onto your floppy. Repeat this process for each of the distribution disks, placing the requested disk in drive A and the next blank disk in drive B.

WARNING You must replace each filled floppy with the next blank one; if you do not, QuickPascal will not function properly.

Once you complete the installation, keep the distribution disks in a safe place. Save them as backup copies.

Run QuickPascal by putting disk 1 in drive A and a data disk in drive B. The data disk holds the programs you write, along with any units those programs may need (other than SYSTEM.QPU, which is on disk 1).

Note that you must explicitly copy the units your program needs to your data disk. QuickPascal will not do it for you. For more information on units, read the section “Writing a Program with Multiple Source Files” in Chapter 3.

When you ask for help, a dialog box warns you that the help files could not be found, and gives you the option of canceling or retrying. Exchange the data disk in drive B with the help disk (disk 3 for 5 1/4-inch or disk 2 for 3 1/2-inch) and select “Retry.” When you finish with help, replace the data disk in drive B.

Continue now with Chapter 3, “Using QuickPascal,” which discusses the QuickPascal environment and steps through a couple of sample programming sessions.

Using QuickPascal

If you followed the instructions in Chapter 2, “Installing QuickPascal,” you now have a working version of QuickPascal and are ready to write your first program. In this chapter you’ll work through a sample compiling and linking session. When you finish you will have written, saved, built, and tested a working program.

If you prefer to experiment on your own, skip this chapter (or skim through it). If you do nothing else, review the next chapter, “Getting Help.” In addition, it is recommended that all QuickPascal users run the QP Express program, a tutorial on how to use the QuickPascal environment.

The QuickPascal Environment

QuickPascal is a window-based programming environment that integrates a text editor, a compiler, a debugger, and an on-line help database. This chapter introduces and describes the following aspects of QuickPascal:

Windows and Menus The menu system allows you to find the command or action you need quickly. The first part of this chapter explains how to open and close windows and how to navigate through the menus. It also defines terms used throughout the chapter.

Programming Environment When you’re writing a Pascal program (or “source code”), you’ll spend a lot of time using the QuickPascal programming environment. If you’re already familiar with Microsoft Quick languages such as QuickC or QuickBASIC, you’ll know how to use the QuickPascal environment.

If you prefer different editing commands, you can use the QPMKKEY utility to change the meaning of command keys. QPMKKEY is explained in the “Customizing the Editor” section of this chapter.

Compiler You can compile and test a program without ever leaving the programming environment. QuickPascal’s integrated environment saves you hours of development time.

Debugger The debugger allows you to set places or conditions where you want the program to stop (“breakpoints”), to monitor important variables, and to trace program execution line by line. This section of the chapter provides a brief overview of the debugger. The QP Express program includes a lesson that provides more details about these debugging features.

Using Windows and Menus

Even if you have never used windows and menus before, you will find that the QuickPascal programming environment is easy to learn.

This section introduces the QuickPascal environment. You’ll learn how to control windows and how to choose commands from the menus.

Getting Started

To run QuickPascal, move to the directory containing the file QP.EXE (if you’re not already there) and type

QP

at the DOS prompt. You’ll immediately enter the QuickPascal programming environment.

If you enter a file name after the QP command, as in

QP MYFILE

QuickPascal automatically adds the extension .PAS that marks a Pascal source file. Typing the line above causes QuickPascal to load MYFILE.PAS.

If QuickPascal can't find the file you specified in the current directory, it asks if you want to create a new file.

If you do not enter a file name after the `QP` command, QuickPascal opens an empty file named UNTITLED.PAS, which you can later rename or save with another name.

Command-Line Options

Depending on your particular hardware configuration, you may want to include one of the following options on the command line after the `QP` command (for example, `QP /b MYPROG`):

<u>Option</u>	<u>Hardware</u>
/1	For EGA and VGA (16-color) displays
/2	For monochrome monitors that support character attributes (e.g., high intensity)
/3	For CGA (8-color) displays
/b	For LCD screens and black-and-white monitors that do not support character attributes (e.g., high intensity)
/25, /43, /50	For systems equipped with MCGA-, EGA-, or VGA-compatible graphics cards capable of displaying 25, 43, and 50 lines of text, respectively

Using the Mouse and Keyboard

You can enter all commands from the keyboard. If you own a Microsoft Mouse (or fully compatible mouse), you can choose to use either the keyboard or the mouse to enter commands. When this book explains a command, the two options are marked with icons of either a key or a mouse as follows:



Keyboard Example: press the ALT key, then F, S.



Mouse Example: click the File menu, then click Save.

NOTE Unless the Right button is specifically mentioned, "clicking" means that you click the Left mouse button once.

Windows

QuickPascal's windows and menus are intuitive and simple to use. If you're confused about windows or menus, call up a Help window that explains how a menu works (see Chapter 4, "Getting Help"). However, if you want to read further in this chapter, you need to understand the terms used.

Figure 3.1 shows a typical QuickPascal screen, with labels that describe its parts. Some of the parts provide information only. For example, if the CAPS LOCK or NUM LOCK key is on, the letters C or N appear in the bottom right corner. The letters C or N are informational. Some parts of a window perform actions triggered by a specific key or mouse action. For example, if you click the dot between two vertical lines in the upper left corner of a window (the Close box), the window closes. The box is active rather than informational.

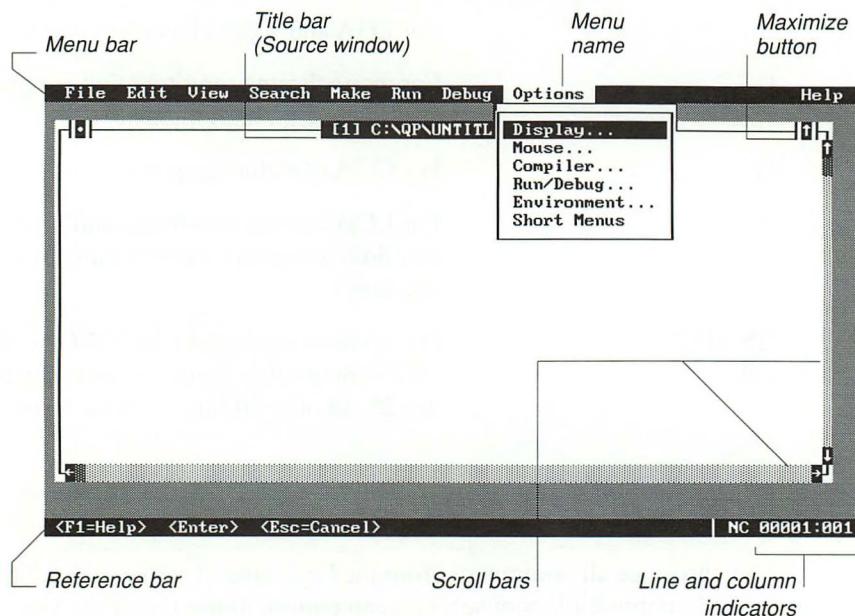


Figure 3.1 Parts of a Window

The parts of a window and their use are listed in Table 3.1 below.

Table 3.1 Parts of a Window

Name	Use
Close box	Closes the current window. Appears in upper left corner.
Menu bar	Lists names of the available menus.
Title bar	Shows name of the window (a source-window title bar lists the file currently being edited).
Source windows	Contain the Pascal code for the program you're writing. Multiple source windows can be open at once. Three other types of windows are supported: Debug, Help, and Output.
Maximize button	Shrinks or enlarges the current window.
Reference bar	Lists shortcut keystrokes (keyboard users) and direct commands to QuickPascal (mouse users).
Scroll bars	Indicate your position in the current file. If you click in the gray area on either side of the elevator, you move in that direction by one page. If you click on the arrows, you move one line (or one character) in that direction. If you click and drag the elevator, you can move anywhere within the file.
Line/Column indicators	Show the current line and column of the text cursor.
Keyboard status indicators	C means CAPS LOCK is on. N means NUM LOCK is on. R means the file is set to Read Only status.

Using the Menu Bar

To choose a command from a menu, “pull down” or “open” the menu and choose the command you want:



1. Press the ALT key to activate the menu bar.
2. Press the highlighted character in the menu name (for example, F for File).
3. Press the highlighted character in the item name (for example, in the File menu, S for Save).

Or follow these steps:



1. Press the ALT key.
2. Use the right and left DIRECTION keys to move to the menu you want.
3. Use the up and down DIRECTION keys to highlight the command.
4. Press ENTER to choose the command.

Or:



1. Click the menu name to open the menu.
2. Click the command.

Cancelling an Open Menu

The menu closes immediately after you choose a command. However, if you decide not to choose a menu command, you can close the menu by



Pressing the ESC key.



Clicking the screen outside the menu.

Shortcut Keys

In the menus below, you'll notice that certain menu items are followed by names of keys. These are the "shortcut keys" for frequently used commands. For example, the Run menu's Restart command is followed by SHIFT+F5, which means that instead of opening the Run menu and choosing the Restart command, you can hold down the SHIFT key and press F5.

NOTE *The reference bar displays commonly used shortcut keys. In addition, the inside cover of this book lists the important shortcut keys for easy reference.*

Menus

The menu bar contains nine menus that you can pull down at any time. If you don't know what a menu does, call up the QP Advisor by highlighting the menu title (or highlighting a command within a menu) and pressing F1.

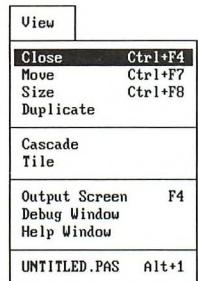
QuickPascal offers both a Short and a Full set of menus. Short menus include all of the basic commands needed to program in Pascal; Full menus include all of the Short menu commands, plus a few that increase your control over QuickPascal. The Full menus are described below:



The File menu controls files, allowing you to clear the Source window (New), load an existing file (Open), insert a file to the source code in memory (Merge), save the current file (Save), save the current file under a new name (Save As), save all the files currently open (Save All), print a file or selection (Print), temporarily invoke DOS (DOS Shell), or exit QuickPascal (Exit).



From the Edit menu you manipulate text by deleting (Clear), cutting (Cut), copying (Copy), and pasting (Paste). You can set Read Only status to protect source files, and you can tell QuickPascal to color-code the various parts of Pascal syntax in different colors (Pascal Syntax).

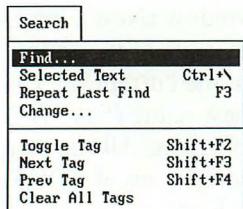


The View menu controls the windows that appear on the screen. You can close the active window (Close), shift the position of a window (Move), resize a window (Size), and open another window into the current file (Duplicate).

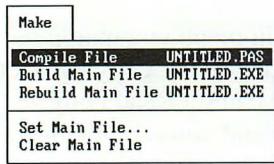
Two commands rearrange the windows on the screen. You can organize multiple windows into a stack with the top of each window showing (Cascade), or make each smaller so that they all fit on the screen without overlapping (Tile). If you run a program within the QuickPascal programming environment, you can review your program's screen output (Output Screen).

During typical program development, you use a window to view the values of your program's variables (Debug Window) and to get assistance on Pascal keywords (Help Window). The bottom part of the View menu contains a list of the currently open windows.

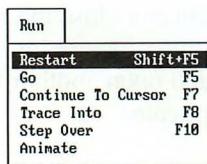
Choosing a window's name makes it the active file (UNTITLED.PAS). A bullet appears next to the name of a window that has been changed. (Note that you can make each window active by pressing ALT + the window's number.)



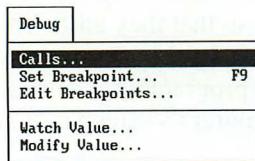
The Search menu invokes the commands that find or replace text in source files. It also moves to lines that you mark ("tag") as text you want to be able to find easily.



The Make menu allows you to compile the current file (Compile), or compile a program's main file (Build), or compile all the parts of a multiple-module program (Rebuild). From this menu, you also define the main component of a multiple-module program.



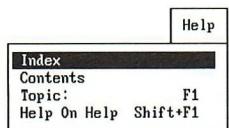
Once a program is compiled, use the Run menu to run it. You can run the program from beginning to end (Go), run the program until it encounters a breakpoint (Step Over), or trace through the program line by line (Trace).



If your program contains logic errors, the Debug menu allows you to set breakpoints (Set and Edit). In addition, you can keep track of variables and their changing values (Watch Value). (This feature means there's no need to place **Writeln** statements throughout your program just to watch variables as they change values.)



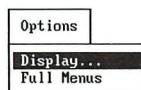
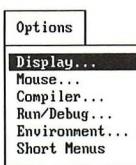
The items on the Options menu control the programming environment. The Display command lets you change the colors used on the screen, and the Mouse command controls the function of the Right mouse button. The Compiler command controls directives that affect the actions of the compiler. Run/Debug sets the screen attributes used during debugging, and Environment sets QuickPascal's default search paths for the different file types. The Full Menus/Short Menus command lets you switch to the menu system you prefer.



When you want additional information about any aspect of QuickPascal, use the Help menu. It leads directly to the index and table of contents for the QP Advisor, as well as help on the word containing the cursor (Topic) and a guide to the help system itself (Help On Help). For more about this topic, see Chapter 4, "Getting Help."

Short Menus and Full Menus

If you open the Options menu, you'll see one of the two menus below:



When six commands are listed under Options and Short Menus appears as the last command, it means the Full Menus command is turned on. When the Options menu holds only two commands and Full Menus is the last command, Short Menus is in effect.

NOTE When you set preferences from the Options menu—Display options, Compiler options, and so on—you can save your choices in a file called QP.INI. If you set an option and save it, it stays set from one QuickPascal programming session to the next or until you change it. If you set an option and do not save it, it will only remain active during the current programming session.

To change from short menus to full menus, choose the Full Menus command. To change from full menus to short menus, choose the Short Menus command.

The short menus contain the basic commands you need to write, compile, and run a Pascal program. The short menus may be easier to use, especially for beginners. If you prefer to see every possible option, enable Full Menus. The choice is yours.

Shaded Commands

When a command within a menu is shaded, it is unavailable. For example, when you first run QuickPascal and haven't yet highlighted any text, you can open the Edit menu and see that both Cut and Copy are shaded. You haven't selected anything, so there is no text to cut or copy.

Ellipses

When a command is followed by three periods (an ellipsis), it means QuickPascal requires you to fill out a dialog box before it executes the command. If a command is not followed by an ellipsis, the command executes immediately when you choose it.

For example, the File menu contains both the Save and Save As... commands. Choosing Save causes QuickPascal to save the current file with the current name (the file name appearing at the top of the Source window). Choosing Save As... causes a dialog box to appear. Within that dialog box, you type the new name for your file.

Dialog Boxes

Often, invoking a menu command causes a dialog box to appear. For example, Figure 3.2 shows the dialog box for the Display command.

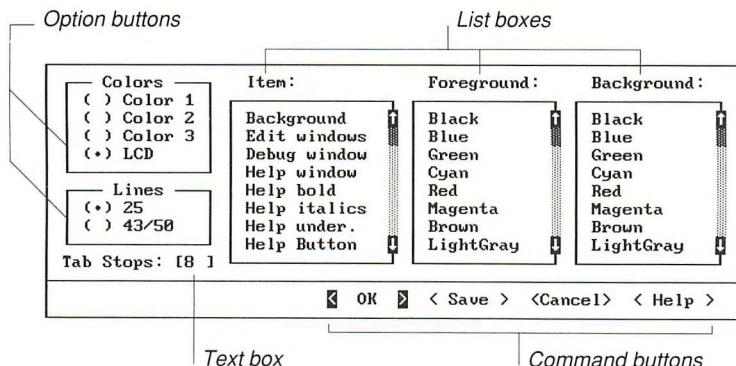


Figure 3.2 Display Dialog Box

NOTE Dialog boxes usually offer a set of shortcut keys. Press ALT to see which keys are active within a dialog box.

Dialog boxes can contain one or more of the items on the following list. Press TAB to move between the various items in a dialog box (or SHIFT+TAB to move backward between items). Press ESC to cancel a dialog box.

<u>Item</u>	<u>Description</u>
Option buttons (•)	Offer a list of choices, of which you can choose only one. Use the DIRECTION keys to move between the choices. In Figure 3.2, four option buttons allow you to pick the color set you prefer.
Check box [X]	An on/off switch (not shown). If the box is empty, the feature is turned off. If it contains the letter X, the feature is turned on. Press the SPACEBAR to turn a check box on or off. Use the TAB key to move between check boxes.

Text box []	Contains text that you enter. In Figure 3.2, the setting for Tab Stops allows you to enter the number of spaces to be inserted when you press the TAB key.
Command buttons <OK>	Pass commands to the dialog box (enclosed in angle brackets). The <OK> button means that you're satisfied with the choices you've made. The <Save> button saves the current options for future programming sessions. The <Cancel> button allows you to exit the dialog box with no changes. When one of the command buttons is highlighted, press ENTER to invoke that command.
List boxes	List items. If the number of items is too large for the list box, you can use the DIRECTION keys and PGUP/PGDN (keyboard) or click the scroll bar to move around the list (mouse).

Using the Editor

QuickPascal's program editor is an important part of the QuickPascal environment. This part of the chapter provides a brief overview of its functions.

Moving Around in a Source File

Many of the keys within the editor act as you would expect. The PGDN and PGUP keys advance you forward or backward within the file. The HOME key moves the cursor to the beginning of the current line. The END key moves the cursor to the end of the current line. The DIRECTION keys move the cursor one character at a time.

NOTE For a complete list of editor commands, use the QP Advisor. Open the Help menu, choose the Contents command, then ask for help about the Keyboard under the Using QuickPascal heading.

You can invoke many of the editor commands in two different ways. For example, to move one word to the right, you can press either CTRL + RIGHT or CTRL + F. The second choice is part of the WordStar ®-compatible command set. If you're familiar with WordStar commands, you already know how to use the QuickPascal editor.

Selecting Text

While you're working on a Pascal program, you may want to delete a large block of text or copy it to another place in the program. To do this, you must select the block:

-  Move the cursor to the beginning of the block. Hold down the SHIFT key and move to the end of the block. Use the other editing keys (HOME, CTRL+RIGHT, END, and so on) while you're holding down SHIFT to extend the block.
-  Move the mouse cursor to the beginning of the block. Hold down the Left button and move to the end of the block.

Once you select a block you can do several things: use the DEL key to erase it or use SHIFT+DEL to erase (Cut) and copy it into the Clipboard, a temporary storage location for the most recently cut or copied text. Press SHIFT+INS to paste the contents of the Clipboard into the source file at the current cursor location.

While a block is selected, anything you type replaces the selected text.

Customizing the Editor

If you prefer to use another set of editing commands, QuickPascal comes with four "key" files and a utility for making your own key file. The files provided let QuickPascal emulate the Microsoft, BRIEF®, and Epsilon™ text editors. The five key files are QP.KEY, ME.KEY, BRIEF.KEY, EMACS.KEY, and EPSILON.KEY.

To change to a new key file, use the /k: option when you run QuickPascal. For example, to load the BRIEF.KEY file, enter this line:

```
QP /k:BRIEF
```

Your preference is automatically saved in the QP.INI file. In future editing sessions, you won't need to specify the key file.

Creating Your Own Key File

The QPMKKEY program allows you to make your own key file. You must use three options: **-c**, **-i**, and **-o**. The first (**-c**) specifies the type of conversion: ASCII to binary (**ab**) or binary to ASCII (**ba**). The other two specify the input file (**-i**) and the output file (**-o**).

To modify the default QP.KEY file, you first convert it to an editable ASCII file:

```
QPMKKEY -c ba -i QP.KEY -o MYEDITOR.TXT
```

You now have a file to print out and use as a quick reference to the editing keys and associated functions.

Use any text editor (including QuickPascal) to edit the file MYEDITOR.TXT, which lists the keystrokes that perform certain actions. For example, the file assigns CTRL+G to delete a character. The line in MYEDITOR.TXT looks like this:

```
Del : CTRL+G
```

You can change that command to any other keystroke (CTRL+D, for example), as long as the key isn't already assigned to another function. Elsewhere in the file, CTRL+D is assigned to CharRight, so you'll have to delete or change that line if you want to use CTRL+D for the Del function. It's a good idea to review the file to see key assignments you've forgotten or aren't aware of.

When you're satisfied with the new functions, you must convert the ASCII file to a binary file, so it can be loaded into the QuickPascal editor:

```
QPMKKEY -c ab -i MYEDITOR.TXT -o MYEDITOR.KEY
```

Finally, use the **/k:** option described in the previous section to load the new key file. For a quick reference, you can print out your MYEDITOR.TXT file.

Using Another Editor

If you prefer to use your favorite word processor or text editor for writing programs, you can do so. However, you must save the program text as an ASCII-only (text-only) file without any formatting information. QuickPascal can work with any program saved as an unformatted ASCII file.

Compiling Your Program

Your ultimate goal in writing Pascal code is to create an executable program. To convert a Pascal source file to a program that runs, you must compile it. This section introduces the commands that create executable programs, a process called “building.”

Building within the QuickPascal Environment

In this section, we’ll demonstrate how to build a Pascal program. First, start QuickPascal. If you use a color monitor, note that as you enter the program, the keywords, comments, and variables all appear in different colors.

```
PROGRAM HI;      { Prints 'hi' and a name }

VAR
    Name : string[25];

BEGIN
    Write( 'Please enter your name. ' );
    Readln( Name );
    Writeln( 'Hi there, ', name,'!' );
END.
```

This program asks you to type your name. You can enter any characters and press ENTER, and it says “hi.”

All Pascal programs use the same basic structures. In the example above, the keyword **PROGRAM** identifies this file as a program. **VAR** identifies the variables—values that can change when the program runs—used by the program. In this case, the only variable is **Name**, a “string” (a series of letters or numbers) up to 25 characters long.

Pascal programs use two keywords, **BEGIN** and **END**, to delineate the program blocks. Blocks are groups of statements that share a common purpose. Every Pascal program has at least one **BEGIN** and one **END** statement.

The lines that begin with **Writeln** and **Readln** send text to the screen and accept characters from the keyboard. They are examples of standard Pascal procedures.

You can find complete information on any of the Pascal keywords, standard procedures, and standard functions in the QP Advisor by positioning the cursor on the keyword and pressing F1. When you finish looking at the information, close the window by pressing CTRL+F4. For more information on the the QP Advisor, read Chapter 4, “Getting Help.”

Building the Program

To build the program, open the Make menu and choose Build Main File.

A dialog box appears on the screen to show you the compiler’s progress. It may halt if anything goes wrong. When the source file contains errors, a dialog box appears and the cursor moves to where QuickPascal detected the error. If the building operation is successful, QuickPascal produces a new executable (.EXE) file in the current directory.

Running the Program

You can test the program immediately after building it without leaving the QuickPascal environment. (Once you build a program you can also run it at the DOS level without starting QuickPascal.) To run the program, open the Run menu and choose Go, or use the following shortcuts:



Press F5.



Click <F5=Run> on the reference bar.

Please enter your name.

appears. If you type Genghis Can Can and press ENTER, the program responds with

Hi there, Genghis Can Can!

Elapsed time = 00:00:04.45. Program returned (0). Press any key

Once you press a key, the screen returns to the QuickPascal environment.

Running the Program after Making Changes

During the programming process, you can expect to make changes to your source code even after you successfully compile and run it for the first time.

For example, if you change the line in the sample program

```
Write( 'Please enter your name. ' );
```

to

```
Write( 'Please type your name. ' );
```

the program must be recompiled before you can run it with the change. However, after you build the program once, it's not necessary to choose Build Main Program again before you press F5 to run it. If you just press F5, QuickPascal knows if the source code changed. If it has changed, QuickPascal asks you if you want to rebuild the program and then runs the program for you. To rebuild the program without running it:



Press SHIFT+F5.



Click Restart on the Run Menu.

A dialog box indicates the compiler's progress.

Viewing the Output Screen

When you choose Go from the Run menu (or press F5), the output goes to the output screen.

After you run a program, you can still switch between the programming environment and the program's output. To see the output:



Press F4 to switch between the two screens.



Open the View menu and choose Output Screen. Click once to return to the Source window.

Saving the Program

To save the source file, go to the File menu and choose either Save or Save As.... If you choose Save, QuickPascal saves the file with the name listed on the top line of the Source window (unless the file is still listed as UNTITLED). If you have not yet named the program or if you prefer to use another name, choose Save As.... The following dialog box appears:

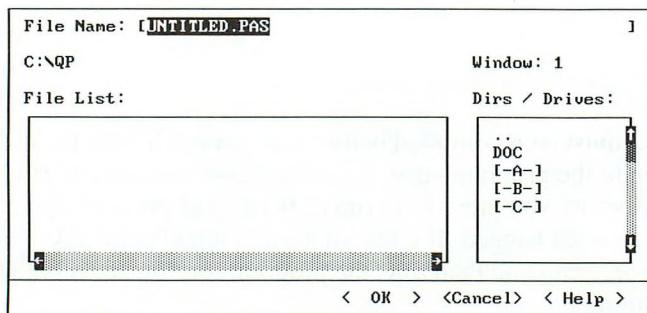


Figure 3.3 Save As... Dialog Box

For the example program, type in HI as the name of the source file and press ENTER. QuickPascal automatically adds the .PAS extension unless you specify a different one (HI becomes HI.PAS, for example). QuickPascal saves the program file on the disk and the new file name appears in the title bar of the Source window.

Writing a Program with Multiple Source Files

Now we'll make things a little more complicated by writing a program with part of its source code in a different file (a "multiple-module" program).

First, modify the HI.PAS program from the previous section, adding a few lines:

```

PROGRAM hia;      { Illustrates internal and external procedures }

USES
    hia_unit;    { Locates the external procedure }

VAR
    Name : STRING[25];

PROCEDURE welcome;    { A separate procedure }

BEGIN
    Writeln( 'Welcome to the program.' );
END;

BEGIN      { Begin main program }

    welcome;    { Call the local procedure }
    outsider;   { Call the external procedure }
    Write( 'Please enter your name.' );
    Readln( Name );
    Writeln( 'Hi there, ', name,'!' );

END.

```

Look at the new program carefully. It contains several new lines that illustrate important concepts in Pascal. Under the program name (HIA) the keyword **USES** indicates that this program uses the named units, which are similar to libraries in other languages. In this case, the unit HIA_UNIT refers to a file that contains a procedure used by HIA.PAS. All programs that use units declare them in a similar manner.

Procedures are an integral part of the Pascal language. Briefly, a procedure is a series of statements the main program refers to with a single name; they perform a task the program repeatedly uses. Procedures appear before the main body of the program. HIA.PAS uses a simple procedure named `Welcome`, but all procedures follow a similar structure.

The main program calls two procedures, `welcome` and `outsider`. Unlike `welcome`, HIA.PAS does not define `outsider`. When the time comes to compile the program, QuickPascal will search the referenced unit (HIA_UNIT) to find the definition of `outsider`.

If you have not already done so, save the new program as HIA.PAS. Don't compile it yet. Now choose New from the File menu and type in the following:

```
UNIT hia_unit; { Contains the external procedure for HIA.PAS }

INTERFACE      { Items available to any program using this unit }

PROCEDURE outsider;

IMPLEMENTATION { Items local to this unit and the actual
                 procedure/function definitions
                 }

PROCEDURE outsider; { Definition of Outsider }
BEGIN
  Writeln( 'This line is from the HIA_UNIT unit.' );
END;

END. { End of unit }
```

Save the new program as HIA_UNIT.PAS. There are now two new source files on your disk, HIA.PAS and HIA_UNIT.PAS.

A unit is any mixture of declarations and definitions of variables, types, constants, procedures, and functions. A program that includes a unit in its **USES** list can access any of the declarations and definitions in that unit's **INTERFACE** section.

In HIA_UNIT.PAS, the keyword **UNIT** identifies the file as a unit rather than a program. The curly braces (`{` and `}`) delineate comments in the source code. The **INTERFACE** section lists those items available to the program calling the unit: variables, constants, types, and the calling formats for each procedure and function. The compiler uses this section as a reference guide to these items. If you use them in a way that conflicts with these declarations, QuickPascal warns you when the time comes to compile your program.

The section under the **IMPLEMENTATION** keyword identifies variables used within this unit but not needed by the calling program, any other units required by this unit, and the actual definitions of the procedures and functions. Because HIA_UNIT.PAS is so simple, it does not need any additional variables or units. The rest of the unit contains the actual definitions of the procedures and functions in the unit.

NOTE *Although one unit can use other units, the reference cannot be circular (unit A can use unit B as long as unit B does not use unit A).*

HIA_UNIT.PAS is very simple, but you can use it as a model for building your own library of functions and procedures.

There are several reasons to split a file into multiple modules. If you write a function or procedure that's used in several programs, you can put it in a unit. If you write long programs, splitting up the file makes editing easier and lets you keep close track of what information is important globally and what is required only locally. If your program uses many variables, you can create a unit that just defines the types for your variables.

In addition, the Build Program command compiles a program faster if you make changes to one file, but not others. The Build Program command doesn't spend time recompiling a unit that hasn't changed.

Compiling a Multiple-Module Program

Since the example program now uses two source files, QuickPascal needs to know which one to use as the main file.

Open HIA.PAS if it is not already open. If more than one file is open, press F6 (keyboard) until HIA.PAS becomes active or click the HIA.PAS window (mouse). Choose the Set Main File command from the Make menu (you must have Full Menus enabled).

The following dialog box appears (Figure 3.4):

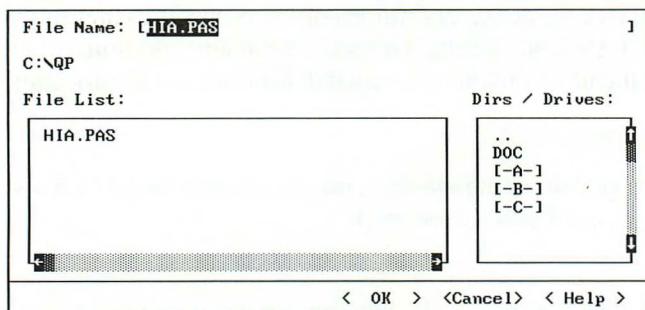


Figure 3.4 Setting the Main File

QuickPascal assumes you want the active file to become the main file, but you can pick any file you wish. For now, just press ENTER. However, if you want to change the file, type *.PAS and press ENTER to see all of the files with the .PAS extension in the current directory and then:



Press the TAB key once to move the cursor to the list box containing the source files (the other list box shows the directory names). Use the DIRECTION keys to highlight the file you want as the new main file, and press ENTER.

Or:



Type the name of the new file and press ENTER.



Click the file you want as the new main file and click the OK button (or just double-click the new main file).

Once you set the main file, choose Build Main File from the Make menu to build the application. QuickPascal automatically compiles both files into a single executable file called HIA.EXE. Because HIA_UNIT.PAS is a unit and not an independent program, QuickPascal produces a .QPU file instead of a .EXE file.

Notice that the base name of the executable file (HIA.EXE) matches the base name of the main file (HIA.PAS). When you work with HIA.PAS and press SHIFT+F5 (Restart) or choose the Build Main File command, the compiler checks the time and date “stamps” on all the .PAS and .QPU files. QuickPascal only re-compiles those source files changed since they were last compiled. If nothing has changed, the old files are still up to date; there’s no need to compile them again. The Rebuild Main File command, on the other hand, always recompiles all the files of a multiple-module program. For this reason, Build Main File is often faster than Rebuild Main File for multiple-module programs.

To run the new program, press F5 (keyboard) or click Go from the Run menu (mouse). The screen now displays:

```
Welcome to the program.  
This line is from the HIA_UNIT.PAS unit.  
Please enter your name.
```

The procedure `welcome` in HIA.PAS generated the first line, and the second line came from the separate HIA_UNIT.PAS unit. Then the program control was transferred back to HIA.PAS to print the third line. The rest of the program is the same as the original example program, HI.PAS.

While HIA.PAS is an uncomplicated program, it does illustrate many of the components of sophisticated Pascal applications. It can serve as a model for any program that uses units, procedures, strings, and screen input and output. And you've seen how to create a file, a unit, and build and run an executable file for a multiple-module program.

Compiling from the Command Line

If you're new to the Pascal language, you can skip this section; you perform the same functions from within the QuickPascal environment.

However, you can exit the QuickPascal environment to build programs from the DOS prompt. If you're building a series of related executable programs, you can put the compile commands in a batch file; this method is faster than building programs individually. Also, if you use an external editor, such as the Microsoft Editor or BRIEF, you don't need to enter the QuickPascal environment in order to compile your programs.

Use the file QPL.COM to build executable files from the command line. To build the example HI.EXE program, type

```
QPL HI.PAS
```

You can include a variety of command-line options between QPL and the file name. For example,

```
QPL /$I-HI.PAS
```

instructs the compiler not to perform any input and output checking. You can get a complete list of compiler options by typing `QPL /help`.

QPL also builds multiple-module programs. Since HIA.PAS is part of a multiple-module program, the compiler needs to be told to create the HIA_UNIT.QPU unit before it builds an executable file. By itself, the line

```
QPL HIA.PAS
```

produces the error message

```
CANNOT OPEN FILE
```

Instead, first create the .QPU unit by typing

```
QPL HIA_UNIT.PAS
```

Now compile HIA.PAS as before:

```
QPL HIA.PAS
```

Because it can locate the referenced HIA_UNIT.QPU unit, the compiler builds the HIA.EXE program.

Another solution is to explicitly tell the compiler to build any units necessary to compile the multiple-module program. You do this in a single step with the /B (Build) option:

```
QPL /B HIA.PAS
```

This command produces the same HIA.EXE file as before, but reduces the number of keystrokes.

These multiple-module cases assume that the source files are in the same directory. If you keep the files in different directories, either place them all in the same directory during compilation or use QPL.COM's /P option to specify the other directories.

The Debugger

When you make mistakes like misspelling a function name or forgetting to end a line with a semicolon, your code causes a compile-time error. The compiler refuses to continue until you correct the mistake.

Other mistakes cause run-time errors. Attempting to divide by zero is an example.

Still other mistakes are called logic errors. When a program includes a logic error, it may run, but it eventually acts unpredictably or yields incorrect results.

QuickPascal's built-in debugger helps you track down and correct run-time and logic errors. In the Run menu, you'll find Trace Into and Step Over, which execute the program in memory line by line. Trace Into follows procedures when they're called; Step Over lets you execute a procedure without showing its inner workings.

Use the Debug menu to set breakpoints and run a program up to the breakpoint.

The Watch Value command from the Debug menu is also useful. You can enter one or more variable names and then monitor their values as you step through the program. There's no need to place **Writeln** statements at various points in a program just to monitor the value of a variable.

The on-line tutorial includes a lesson explaining how to use the debugger. If you're interested in exploring this topic, run the QP Express tutorial.

The on-line help system keeps information on important Pascal topics at your fingertips. You need not thumb through a large reference manual to find a procedure's parameters or a function's return value. Just press a key and the information you need instantly appears on the screen.

The QP Advisor

This chapter explains the many ways to use the QP Advisor:

Keyword Help On-line help recognizes Pascal keywords, operators, standard procedures and functions, and symbolic constants as topics for which it can provide definitions, syntax, and examples.

Topic-Based Help If you don't know the name of a procedure or keyword, you can browse through the index of topics or look in the table of contents.

Environment Help If you're not sure what a menu does, you can ask for help about the menu in general or one of the menu commands in particular. In addition, whenever a dialog box appears and you want more information on the choices, press F1. A help screen appears and guides you through the available options.

Error Help When the compiler returns an error message, QuickPascal highlights the error. If you're not sure what's wrong, on-line help will provide more information about the error message.

Not only are the help screens a great learning tool for programmers new to the Pascal language, they're also a fast and useful reference for experienced Pascal programmers. Remember that you can copy and paste any text or sample code directly from the QP Advisor to your program.

Keyword Help

You may ask for help about any keyword, operator, symbolic constant, standard procedure, or standard function. This section explains how to open a help window, using the standard procedure **Writeln** as an example.

First, run QuickPascal and type **Writeln**.

Here's how to get help about the **Writeln** procedure:

Position the cursor anywhere on **Writeln** (or on the space just after the word) and open the Help menu. Choose the **Topic: Writeln** command. Note that the word under the cursor always appears after **Topic** in the Help menu.

You may prefer to use the following shortcuts:



Position the cursor somewhere on **Writeln** and press F1.



Move the mouse cursor to **Writeln** and click the Right mouse button.

NOTE *If nothing happens when you press the Right mouse button, open the Options menu, choose Mouse, and click Context Sensitive Help as the Right Mouse Button setting. Choose OK and click the Right mouse button on **Writeln** again.*

Whether you use the menu, the F1 key, or the Right mouse button, the help screen that describes **Writeln** appears, as shown in Figure 4.1:

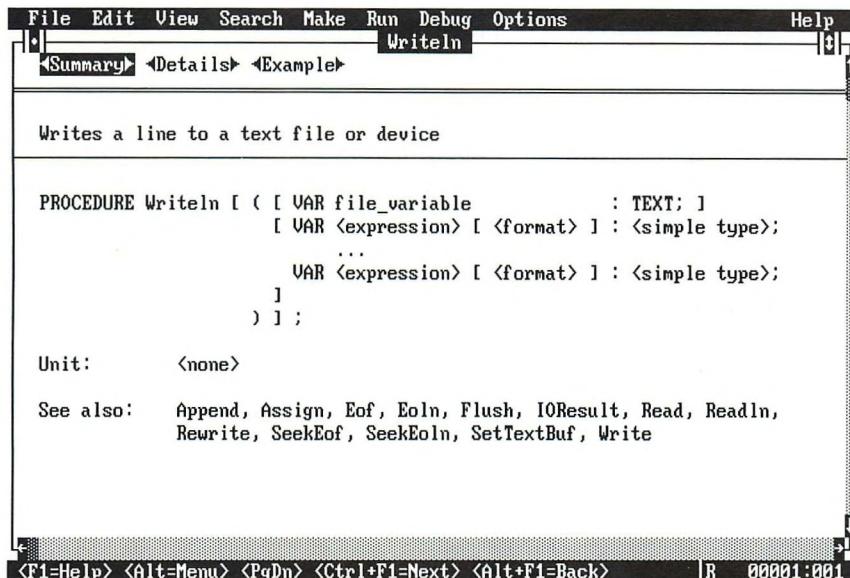


Figure 4.1 Help on **Writeln**

Topics and Hyperlinks

All keywords, operators, and library functions are called “topics” within the QP Advisor. The example above activated the help screen for the **Writeln** topic.

In the Help window describing **Writeln**, you’ll see certain highlighted words and phrases at the top of the screen. These are called “hyperlinks.” They link the current help screen to additional related topics.

The difference between a topic and a hyperlink is simple. You may ask for help on any topic at any time (by pressing F1, clicking the Right mouse button, or using the Help menu). You may even ask for help from within the Help window.

However, if you select a topic that doesn't exist—"elephant," for example—the QP Advisor checks the list of available topics to see if it can offer help. If it can't, it beeps and no help appears.

Hyperlinks, on the other hand, always lead to a related help screen. Activate hyperlinks the same way you ask for help about a topic: press F1 or click the Right mouse button.

For example, here's how you activate the hyperlink labeled **Details** from the **Writeln** Help topic:

-  With the **Writeln** screen still visible, use the TAB key to position the cursor on the **Details** hyperlink. Press F1. (If the source window is active, first press F6 to move to the Help window.)
-  Move the mouse cursor to the **Details** hyperlink and click the Right button.

The help screen with specific details on the use of **Writeln** appears, as shown in Figure 4.2:

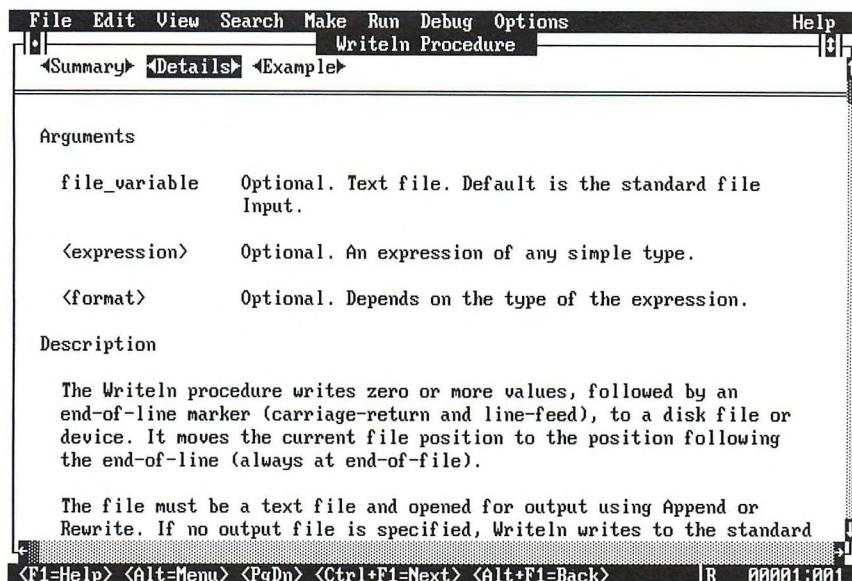


Figure 4.2 Writeln Details Screen

To close the Help window, make it active and press CTRL+F4 (keyboard) or click the Close box (mouse).

Copy and Paste from Help

Any text that appears in the Help window can be copied to the Source window.

This feature of the on-line help system is very useful. If you'd like to test an example program from the Help window, copy the program to the Source window and compile it.

If you need to copy only a few lines from the example, you can do it quickly. Just follow the steps below.



1. Move the cursor to the beginning or the end of the text you want to select. Hold down the SHIFT key and move the cursor to the other end of the text. The text is now highlighted.
2. Copy the text. Press ALT to activate the menus, E for Edit, C for Copy. Note that the menu lists the shortcut command CTRL+INS, which may be substituted for ALT, E, C.
3. Press F6 to switch to the Source window. Position the cursor where you want to insert the text and choose the Paste command: ALT, E, P. Or use the shortcut SHIFT+INS. The text you copied from the Help window is inserted at the current cursor position in the program.

Or:



1. Click and drag (hold down the Left button and move the mouse) to select the text you want to copy.
2. Choose Copy from the Edit menu (or press CTRL+INS).
3. Click once in the Source window to activate it, move the mouse cursor to the location where you want to insert the text, and click once. Then choose Paste from the Edit menu. The text you copied from the Help window is inserted at the current cursor position in the program.

Viewing the Previous Help Screen

The QP Advisor remembers the last 20 topics you accessed. Returning to one of the previous topics is easy. Hold down the ALT key and press F1 as many times as necessary to return to the screen you want to see. For example, if you want to see the third screen back, hold down ALT and press F1 three times. The help screen you see is active; you can ask for help on any of its hyperlinks or topics.

Topic-Based Help

The on-line help system also includes a table of contents for help topics, which comes in handy when you have only a general idea of what you need. To browse through the help system, go to the Help menu and choose Contents.



Press ALT, H, C.



Click once on the Help menu. Choose Contents.

The Contents screen appears, as show in Figure 4.3:

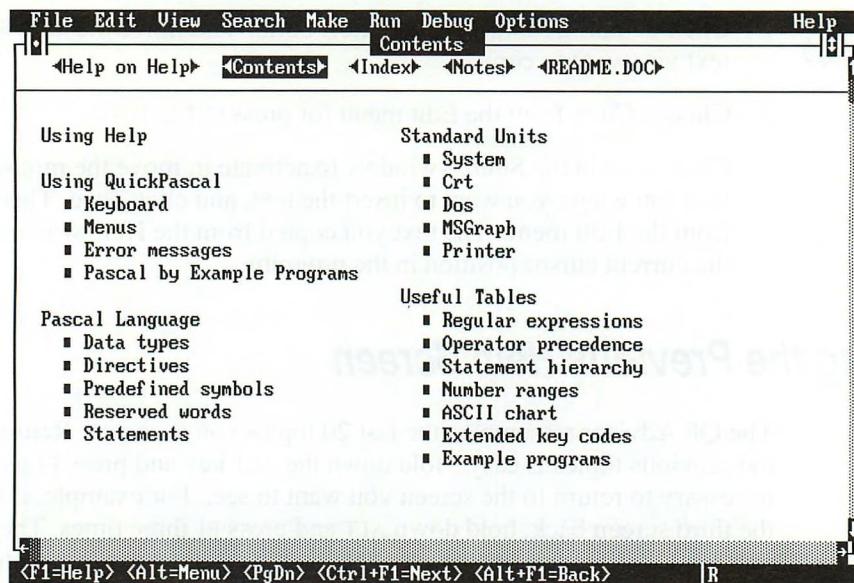


Figure 4.3 QP Advisor Contents Screen

Suppose you want more information about the **Crt** unit.

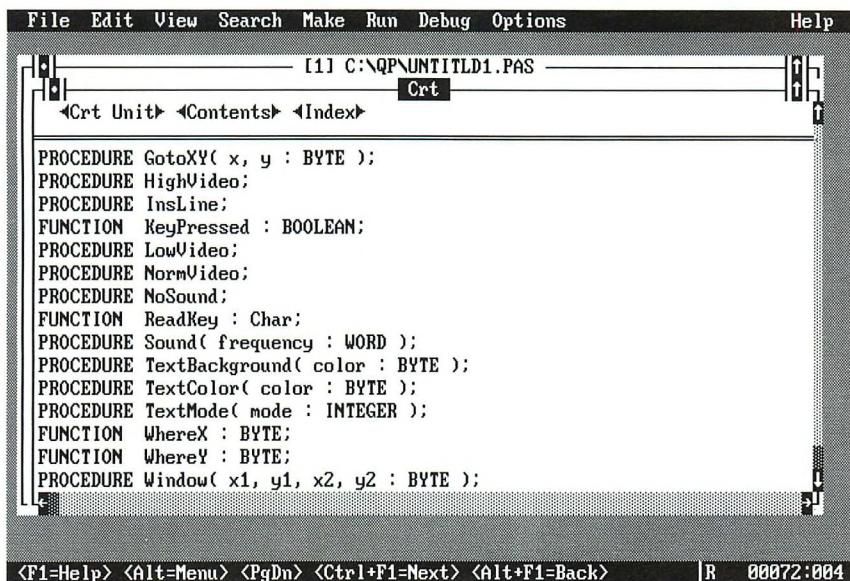


Move the cursor to the **Crt** unit topic and press F1.



Click the Right button on the **Crt** unit topic.

A screen appears with a list of the **Crt** unit's procedures and functions. Suppose that you're looking for a procedure that positions the text cursor. Scroll down the list until you find the **GotoXY** procedure, as shown in Figure 4.4:



The screenshot shows a Delphi IDE window. The menu bar includes File, Edit, View, Search, Make, Run, Debug, Options, and Help. The title bar shows the file path [1] C:\QP\UNTI1D1.PAS and the unit name Crt. The main window displays the contents of the Crt unit, listing various procedures and functions. The cursor is positioned over the word 'GotoXY'. At the bottom of the window, there is a status bar with keyboard shortcuts like <F1=Help> and <Alt=Menu>, and a timestamp R 00072:004.

```
PROCEDURE GotoXY( x, y : BYTE );
PROCEDURE HighVideo;
PROCEDURE InsLine;
FUNCTION KeyPressed : BOOLEAN;
PROCEDURE LowVideo;
PROCEDURE NormVideo;
PROCEDURE NoSound;
FUNCTION ReadKey : Char;
PROCEDURE Sound( frequency : WORD );
PROCEDURE TextBackground( color : BYTE );
PROCEDURE TextColor( color : BYTE );
PROCEDURE TextMode( mode : INTEGER );
FUNCTION WhereX : BYTE;
FUNCTION WhereY : BYTE;
PROCEDURE Window( x1, y1, x2, y2 : BYTE );
```

Figure 4.4 Crt Unit Topics

Ask for help about the procedure by positioning the cursor on **GotoXY** and pressing F1 (keyboard), or by clicking the Right mouse button on **GotoXY** (mouse).

The Summary screen appears, as shown in Figure 4.5:

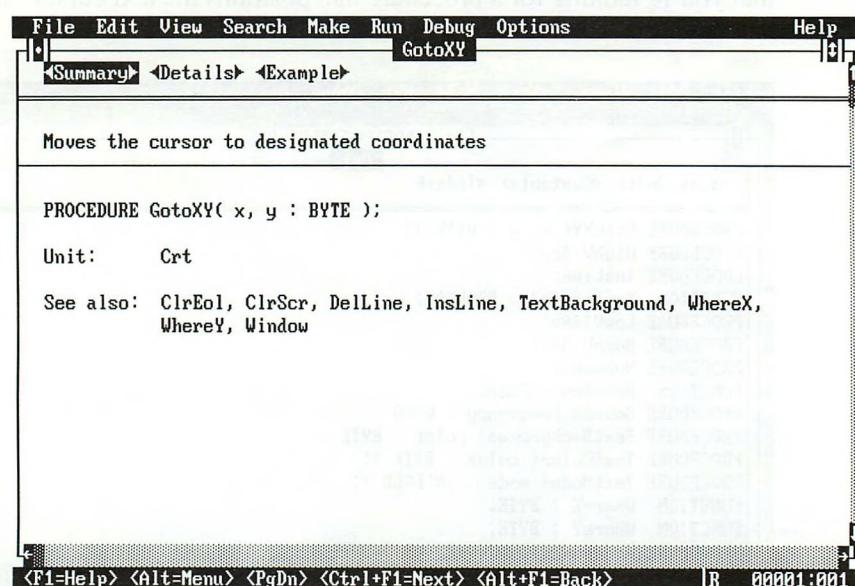


Figure 4.5 GotoXY Summary Screen

To read about **GotoXY**, tab to the **Details** button and press F1, or click the hyperlink with the Right mouse button. The following help screen appears:

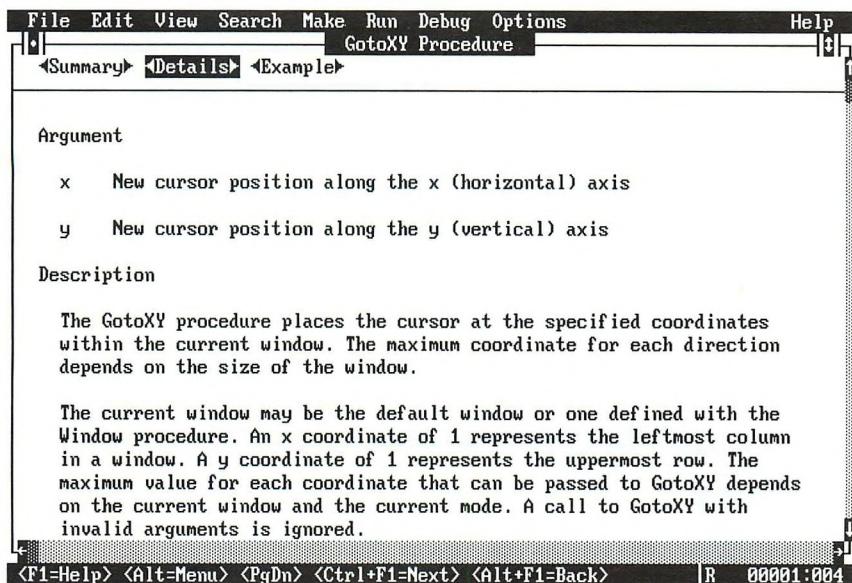


Figure 4.6 Details of GotoXY

When you started, you had only a general idea of what function you wanted, but you tracked down a specific routine through the table of contents.

Environment Help

You can get help on any of QuickPascal's menu commands. For example, open the File menu. (Be sure you have Full Menus turned on.) Suppose you notice the Merge command but you're not sure what it does. Highlight—but don't choose—the Merge command:

-  Press ALT, then F (for the File menu), and use the DOWN DIRECTION key until the Merge command is highlighted. Press F1 for help.
-  Click once with the Left mouse button to open the File menu. Use the DOWN key to highlight Merge and press F1.

The following screen appears, describing how to use the Merge command:

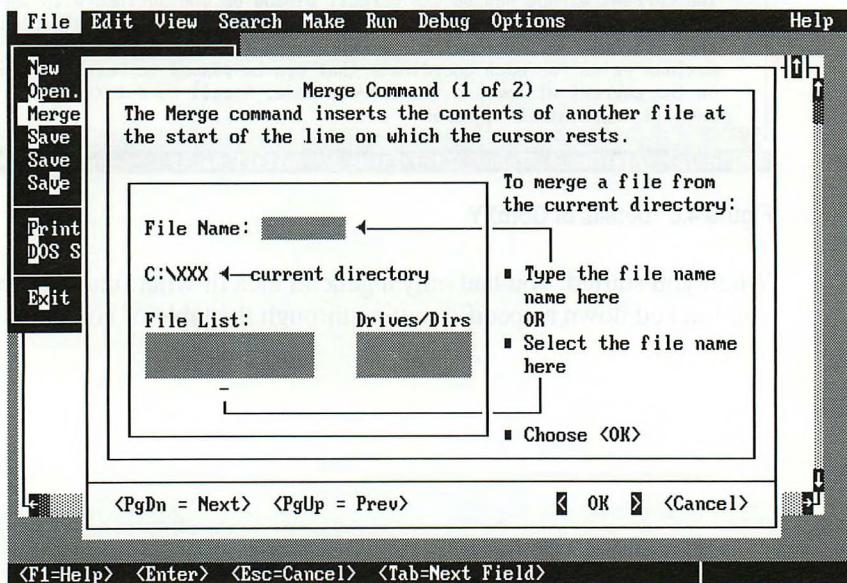


Figure 4.7 Help on the Merge Command

Using the methods described above, you can ask for help on any of the menus or menu options.

Error Help

Error help provides information about compiler and linker error messages. For example, suppose you try to build this program:

```
PROGRAM ErrExmpl;  
  
VAR  
    not_a_char : Integer  
BEGIN  
    not_a_char := 'x';  
END.
```

The following Error dialog box appears on the screen:

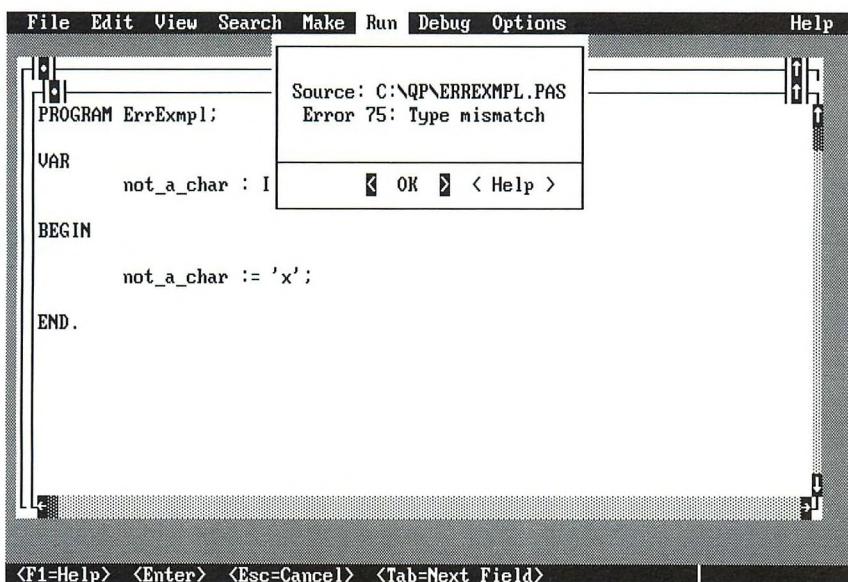


Figure 4.8 Error Dialog Box

The error message says Error 75: Type mismatch, which may or may not give you enough information to solve the difficulty. If you want more information:

- ☛ Press F1 for help or press the TAB key to highlight the Help button and press ENTER.
- ☛ Click the Help button.

The Help window opens, as shown below in Figure 4.9:

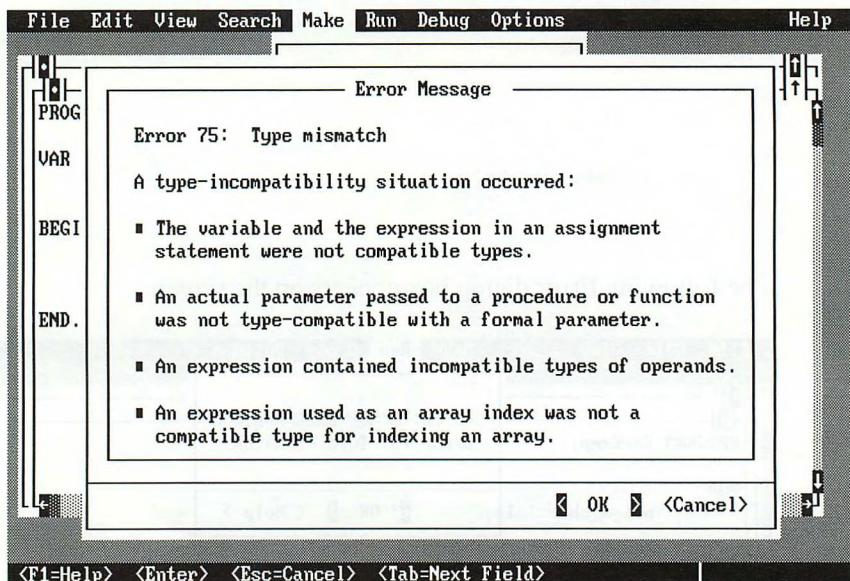


Figure 4.9 Help on an Error Message

Since the error occurred when assigning a value to a variable, the first item listed must be the cause of the error (none of the others deal with assignments).

When you're ready to return to the Source window, press CTRL+F4 to close the Help window.

Help On Help

For a quick guide to the available help functions, open the Help menu and choose Help On Help. Or use the shortcut keys, SHIFT+F1.

Help On Help describes briefly how to use the help system. From within the Help On Help screen, you can use hyperlinks to see the Contents and Index. Those screens can guide you to other subjects, including ASCII tables, lists of Pascal keywords, lists of data types, and much more.

Where to Go from Here

You've now installed QuickPascal on your system and worked through a couple of sample programming sessions. What you do next depends on your level of experience. You can experiment, or you can further investigate the other book in this package, *Pascal by Example*.

Read README.DOC

The README.DOC file lists all known corrections and additions to the printed manuals. Before you continue, please read this file. You can view it from within the QP Advisor or you can use the QuickPascal editor (type `QP README.DOC` to read the file).

For All Programmers

The QuickPascal environment has been designed with you in mind. You'll find writing, compiling, and debugging QuickPascal programs faster and easier than ever. To learn more about the editor, compiler, and debugger, run the QP Express tutorial on the distribution disk labeled "QuickPascal Express" (on either 5 1/4-inch disks or on 3 1/2-inch disks).

Place the "QuickPascal Express" disk in your disk drive, change to that drive, type `EXPRESS`, and press ENTER. The QP Express program contains four lessons: How to Use This Tutorial; Getting Around in QuickPascal; Load, Compile, and Run in QuickPascal; and Debugging in QuickPascal.

For New Programmers

The book *Pascal by Example* was written for beginning programmers as well as programmers new to Pascal. If you have never programmed in Pascal before or need a refresher course, start with the first five chapters of *Pascal by Example*. The introduction lists several books that newcomers to Pascal may also find helpful.

Another good source of information about how Pascal works is the QP Advisor. Declarations, explanations, and examples for any keyword, procedure, or function are immediately available at the press of a key. You may ask for help about a specific topic or browse through the table of contents.

For Pascal Programmers New to QuickPascal

Depending on your previous Pascal experience, you may wish to skim the contents of *Pascal by Example*, or read from Chapter 8 on.

Even if you already have a strong background in Pascal, you'll probably find the appendixes to *Pascal by Example* useful. They describe the **Dos**, **Crt**, **MSGraph**, **Printer** and **System** units supplied with QuickPascal, in addition to the special directives and options available for customizing the compiler.

If you know some Pascal but do not consider yourself an expert, try working through the examples in Chapters 5–15 of *Pascal by Example*, picking topics that interest you and apply to your programs. Those chapters cover units, user-defined data types, working with screen input and output, arrays, strings, graphics, a variety of file-handling techniques, and object-oriented programming.

Microsoft Corporation
16011 NE 36th Way
Box 97017
Redmond, WA 98073-9717

Microsoft®