SYMANTEC

THINK's
Lightspeed Pascal™

The Professional's Choice

USER'S MANUAL

For Macintosh Plus,
Macintosh SE, and
Macintosh II computers

THINK's
Lightspeed Pascal

The Professional's Choice

USER'S MANUAL

Credits

User's Manual: Philip Borenstein

Software: John McEnerney, David Neal, and Peter Maruhnic

Product Manager: Diana Bury

Symantec Corporation
10201 Torre Avenue
Cupertino, CA 95014
408/253-9600

Technical Support: 617/275-1739

ResEdit, RMaker, and Macsbug are copyrighted programs of Apple Computer, Inc. licensed to Symantec Corp. to distribute for use only in combination with THINK's Lightspeed Pascal. Apple software shall not be copied onto another diskette (except for archive purposes) or into memory unless as part of execution of THINK's Lightspeed Pascal. When THINK's Lightspeed Pascal has completed execution, Apple Software shall not be used by any other program.

APPLE COMPUTER, INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED REGARDING THE ENCLOSED SOFTWARE PACKAGE, IT'S MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED IN SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY PROVIDES YOU WITH SPECIFIC LEGAL RIGHTS. THERE MAY BE OTHER RIGHTS THAT YOU MAY HAVE THAT VARY FROM STATE TO STATE.

30 Day Money-Back Guarantee
If within 30 days of purchase this product does not perform in accordance with our claims, you may return it to your dealer for a refund. If you purchased it directly from Symantec Corporation, a Returned Merchandise Authorization (RMA) must be assigned. To obtain an RMA number, please call (408) 253-9600 between 8:00am and 5:00pm Pacific time. You will be given an RMA number and instructions on where to return the product. No returns will be accepted without an RMA number.

THINK's
Lightspeed Pascal

PART ONE

Getting Started

Contents

5 Tutorial: ObjectDraw DA

THREE USING THINK PASCAL

6 Editing

7 Working with Projects

8 Running Programs

Welcome
1

Introduction

Welcome to THINK's Lightspeed Pascal (THINK Pascal). This chapter tells you about THINK Pascal, what's in your THINK Pascal package, what equipment you need, and what you need to know to start writing Pascal programs on the Macintosh.

Before you begin

Be sure to fill out and return the registration card that came with your THINK Pascal package. Registering insures that you'll get the technical support you need, and that you'll be notified of revisions and upgrades to THINK Pascal.

If you don't read manuals

To get started quickly, read this chapter, the next chapter, and do one of the tutorials.

If you're an experienced THINK Pascal user

If you already use THINK's Lightspeed Pascal, you'll be pleased with the new features in this release. Read "Notes for Experienced Users" at the end of this chapter to learn what's new in THINK Pascal.

Topics covered in this chapter

- What is THINK's Lightspeed Pascal
- What you need
- What's in the package
- What's in the manual
- What you should know
- Notes for experienced users

What is THINK's Lightspeed Pascal?

THINK's Lightspeed Pascal is a unique development environment for the Macintosh. It features a very fast compiler, a faster linker, an integrated text editor designed specifically for Pascal syntax, an auto-make facility, advanced debugging tools, and a project organizer that holds all the pieces together. Because the editor, the compiler, and the linker are all components of the same application, THINK Pascal knows when edited source files need to be recompiled. If you edit the interface section of a unit, the auto-make facility recompiles only the source files that depend on it.

With THINK Pascal you can build Macintosh applications, desk accessories, device drivers, and any kind of code resource. The standard Pascal libraries include the standard I/O functions like `writeln` so you can use Pascal programs that were written for other computers.

THINK Pascal lets you run your program as you work on it. Your program runs as if you had opened it from the Finder. You can use the debugging tools built into THINK Pascal to make sure your program runs correctly. The debugger lets you set breakpoints, step through your code, examine variables, and change their values — even structured variables like records and arrays— while your program is running.

THINK Pascal is fast. So fast that it will change the way you program. Not only are the compiler and the linker many times faster than other development systems, they are part of an integrated package that also includes an editor, debuggers, an automatic "project manager" that keeps track of your edits, and only recompiles source files that have changed since the program was last built.

THINK's Lightspeed Pascal is a remarkable product that delivers the efficiency and power you need. THINK Pascal will seem natural and obvious — the way things ought to be.

A development environment that works with you

In traditional development environments, the edit-compile-link-run loop takes so long that it's impractical to make a small change just to see how it works. Because it takes so long to compile and link, programmers tend to make several changes at once to get more "bang for the buck." Unfortunately, when your program crashes, you have to figure out which one of the small changes actually caused the crash. The "bang" gets very expensive.

Because THINK Pascal is so fast, you can make a small change, compile and link your program, and run it to see the effect before traditional development environments even start linking. And because your program is running in the controlled THINK Pascal environment, you don't have to worry about most of the common crashes. THINK Pascal is right there to catch you so you can start fixing your code without restarting your Macintosh.

In traditional development environments, you have to keep track of various language objects such as source files, object files, a link-control file, an executable image, and administrative files. THINK Pascal takes care of all this bookkeeping for you. In THINK Pascal, you have only your source files, library files, and a **project document**. The project document serves as an on-line project administrator for your program development. THINK Pascal keeps track of whatever changes you make to your source files, automatically compiling and linking wherever necessary, so all you have to do is edit your program and run.

A rich set of debugging tools

Not only does THINK Pascal help you develop your program faster, it also helps you debug your program faster. THINK Pascal has a rich set of powerful debugging tools.

The integrated editor catches syntax errors and pretty-prints (formats) your program as you type it in. THINK Pascal lets you run your program a statement at a time or in slow motion. You can stop your program any time it's running, or you can set breakpoints anywhere in your program.

The Observe and LightsBug windows let you examine and change the values of your variables — even structured variables like records and arrays. The Instant window lets you try out small pieces of code to see how they'd work in your program.

Advanced features let you stop at Macintosh Toolbox routines, and you can even use a low level debugger like TMON or Macsbug.

What You Need

THINK Pascal works best when you have at least 1 megabyte (Mb) of RAM and a hard disk. It will also run with two 800K floppy drives.

How much RAM?

You can run THINK Pascal on a Macintosh Plus, Macintosh SE, or Macintosh II. You can run THINK Pascal on a Macintosh 512Ke if you've upgraded it to at least 1Mb of RAM.

How much disk space?

The complete THINK Pascal system takes up about 800K on your disk, not including your own files. The actual size of your system may be smaller, depending on the kinds of programs you work on.

Although you can use THINK Pascal with two 800K floppy drives, it works much better when you use a hard disk.

Which System and Finder?

Use the latest System and Finder. At press time, this is System Tools 6.0 (System 6.0/Finder 6.1). THINK Pascal requires at least System Tools 5.0 (System 4.2/Finder 6.0).

What's in the Package

Your THINK Pascal package consists of two double sided floppies and this manual.

What's in the Manual

This manual is organized in five sections: Getting Started, Learning THINK Pascal, Using THINK Pascal, Reference, and Appendices. Each chapter begins with an introduction that describes what's in the chapter, followed by a list of the major topics covered in the chapter.

Getting Started

This is the section you're reading. It contains this chapter and the installation instructions. Even if you don't read manuals, be sure to read the installation instructions in Chapter 2.

Learning THINK Pascal

This section contains three tutorials. The first one, "Bullseye," shows you how to write a program in THINK Pascal and how to use the basic debugging tools.

The second tutorial, "ObjectDraw," shows you how to build a Macintosh application in THINK Pascal. It covers some of the more advanced topics like using resource files and libraries.

The third tutorial, "ObjectDraw DA," shows you how to build a desk accessory in THINK Pascal. It covers even more advanced topics like segmentation.

Using THINK Pascal

This section contains ten chapters that describe the different components of THINK Pascal.

Editing, Chapter 6, describes the THINK Pascal editor which is specifically designed to help you write Pascal programs.

Working With Projects, Chapter 7, is about projects, the organizing tool that keeps track of all your source files and object code. You'll learn how to work with source files, how to add them to your project, and how to segment your project.

Running Programs, Chapter 8, shows you how to run a program in THINK Pascal. You'll learn how to compile and link your program, and what to do when something goes wrong.

Debugging Programs, Chapter 9, introduces you to the basic set of debugging tools in THINK Pascal. These tools let you run your program line by line, set breakpoints anywhere in your program, and observe the values of your variables.

Units and Libraries, Chapter 10, teaches you how to create units and libraries. Units and libraries are collections of Pascal code that help you write modular programs.

Using Predefined Routines, Chapter 11, tells you how to use the procedures and functions built into THINK Pascal. It also tells you how to call the Macintosh Toolbox routines.

Building Projects, Chapter 12, shows you how to build the four kinds of programs you can write in THINK Pascal: applications, desk accessories, device drivers, and code resources.

Assembly Language, Chapter 13, gives you all the information you need to know to write assembly language routines that work with THINK Pascal.

LightsBug, Chapter 14, introduces you to LightsBug, THINK Pascal's powerful debugging tool. LightsBug lets you get a closer look at your program. You can examine and change the values of your variables (even arrays and records), look at any part of memory, and examine the heap.

Compiler Directives, Chapter 15, shows you how to use compiler directives — instructions to the compiler — to generate different code for different situations.

Reference

This section contains two reference chapters.

THINK Pascal Menus, Chapter 16, describes the THINK Pascal menu commands.

Language Reference, Chapter 17, describes the Pascal language implemented by THINK Pascal. This is a very long chapter, and you don't have to read all of it. The important sections are sections 9 and 10 which describe the built in input/output routines and the standard routines.

Appendices

This section contains six appendices.

ANS Pascal Compatibility, Appendix A, describes THINK Pascal's compatibility with the ANS Pascal standard.

Porting to THINK Pascal, Appendix B, describes the limitations and non-standard features of THINK Pascal that you need to be aware of when you port programs from other Pascals to THINK Pascal.

BackupProject, Appendix C, describes a utility that makes backups of all the files in your project.

RMaker Reference, Appendix D, shows you how to use the resource compiler.

Error Messages, Appendix E, lists and describes all the error messages that THINK Pascal generates.

Macsbug Reference, Appendix F, shows you how to use Macsbug, a low level debugger.

If you are interested in a specific topic, consult the index in the back of the manual.

Conventions in the manual

The names of menus and commands are in **bold face**.

When a technical term or key word is introduced, it also appears in bold face.

Names of files, code fragments, resource names, function names, and variables appear in "`typewriter face`."

All numbers are decimal. Hexadecimal numbers are written in Pascal notation: $3EFA.

In this manual, the term **Toolbox routine** means any routine in ROM. The Macintosh ROM actually consists of two kinds of routines: Operating System routines and Toolbox routines. Operating System routines deal with low-level aspects of the machine like the file manager, the event posting mechanism, device management, etc. The Toolbox deals with high-level aspects like the drawing environment, the window mechanism, menus, dialogs, etc.

What You Should Know

This manual assumes you know how to use your Macintosh, and that you already know or are at least learning, how to program in Pascal. If you're just getting started in Pascal, THINK Pascal is a great platform.

If you're planning to write Macintosh applications, you should be familiar with the Macintosh Toolbox as described in *Inside Macintosh*. The Toolbox is the set of operating system and user interface routines that make a Macintosh a Macintosh. This manual won't show you how to write Macintosh applications or how to use the Macintosh Toolbox. There are several books that teach you how to build applications — see "Learning to write Macintosh programs" below. You can also look at the sample programs in your THINK Pascal package for examples of Macintosh applications.

One of the best ways to learn both Pascal and Macintosh programming at the same time is with Just Enough Pascal, an on-line programming tutorial. Just Enough Pascal is a desk accessory that works with THINK Pascal to teach you Pascal, Macintosh programming, and THINK Pascal right at your Macintosh.

Learning Pascal

Pascal is the most popular language for learning how to program, so you'll find several books that teach programming in Pascal. Some of the Pascal books are written specifically for use with THINK's Lightspeed Pascal or its cousin, Macintosh Pascal.

The standard reference for the Pascal programming language is the third edition of *Pascal User Manual and Report* (Springer-Verlag) by Kathleen Jensen and Niklaus Wirth, revised by Andrew Mickel and James Miner. The *User Manual and Report* is fairly technical, and it's designed for people who already know fundamental programming concepts.

Scott Kronick's *Macintosh Pascal Illustrated: The Fear and Loathing Guide* (Addison-Wesley) is an amusing and unorthodox introduction to programming the Macintosh in Pascal. His book covers most of the things you need to know to get started.

Oh! THINK's Lightspeed Pascal (W. W. Norton) by George Beekman and Michael Johnson is a companion to *Oh! Pascal* (W. W. Norton) by Doug Cooper and Michael Clancy. Together, these books are a good introduction to Pascal and THINK Pascal.

Another good book if you're learning Pascal is *Macintosh Pascal* (Houghton-Mifflin), by Robert Moll and Rachel Folsom. It is particularly helpful because of the many similarities between Macintosh Pascal and THINK Pascal.

If want to learn Object Pascal, see Kurt Schmucker's *Object-Oriented Programming for the Macintosh* (Hayden). This book contains a great deal of information about object-oriented programming in general and some details about using Object Pascal.

Learning to write Macintosh programs

If you're new to programming the Macintosh, you might find yourself overwhelmed by the complexity of the Macintosh Toolbox and unfamiliar programming techniques. When the Macintosh was released in 1984, there was very little technical information available to casual programmers, and even commercial developers had a hard time figuring out how to get things to work correctly. The Macintosh is even more complex today than it was in 1984, but now there are more places you can go for information.

There are now several good books that introduce you to programming the Macintosh and teach you some of the finer points of using the Macintosh Toolbox. No matter which books you choose to help you get started, *Inside Macintosh* is indispensable.

Inside Macintosh Volumes I-V (Addison-Wesley) is the official reference that describes the more than 600 Macintosh Toolbox routines. You might be able to get by without it for a