About This Book

This book, *Inside Macintosh: Networking,* describes the AppleTalk protocols and the application programming interfaces to them. AppleTalk is a network system including hardware and software that supports communication over a variety of data-link types. Using AppleTalk, applications and processes can transfer and exchange data and share resources. The central part of the AppleTalk software consists of a number of protocols arranged in layers, with each protocol offering different services.

To familiarize you with the functions that each of these protocols provide so that you can determine which protocols to use for your application, this book includes an overview of the AppleTalk protocols. This book describes how to write a networked application that uses the AppleTalk application programming interfaces to send and receive data. It describes how to use different methods to send data, such as establishing a sustained connection across which you can transfer streams of data or transferring data in small, discrete units called packets.

To gain an understanding of AppleTalk as a whole and a perspective of the types of services that each AppleTalk protocol provides, see the chapter "Introduction to AppleTalk." This chapter explains some basic networking concepts and how they apply to AppleTalk. It describes how addressing is implemented in AppleTalk networks and how this affects your application. It also explains how you can use each of the AppleTalk protocols for specific application requirements, and finally, it discusses a feature that is common to all routines across AppleTalk protocol interfaces: how to use either of two methods, synchronous or asynchronous, to specify when control is returned to your program after you call a routine.

To learn how to obtain information about the AppleTalk drivers and the networking environment and how to send packets to other applications and processes on your own node, see the chapter "AppleTalk Utilities."

To determine how to register your application with AppleTalk so that it is visible on the network and available for other applications and processes to contact and also how to obtain the addresses of other applications and processes so that you can contact them, see the chapter "Name-Binding Protocol (NBP)."

To obtain zone location information for the node that is running your application or other applications on an AppleTalk network, see the chapter "Zone Information Protocol (ZIP)."

To provide support for a networked application that establishes and maintains a peer-oriented session connection between your application and its partner on the network and that allows the applications to send streams of data to each other, see the chapter “AppleTalk Data Stream Protocol (ADSP).” This chapter also discusses how you can establish a secure connection that provides for user authentication and data encryption.

To provide support for a transaction-based session application in which one end of the connection controls the session and issues a transaction request that the other end carries out, see the chapter “AppleTalk Transaction Protocol (ATP).”

To gain access to the underlying AppleTalk transport protocol that allows you send discrete packets of data across the network without imposing on your application the additional overhead required to set up and maintain a session, see the chapter “Datagram Delivery Protocol (DDP).” To use DDP, you must provide socket-listener code that you must write in assembly language.

To provide complete coverage of the AppleTalk protocols, this book includes in the chapters “AppleTalk Session Protocol (ASP)” and “AppleTalk Filing Protocol (AFP)” a discussion of two higher-level protocols that are not commonly used by application program developers: AppleTalk Session Protocol (ASP) and AppleTalk Filing Protocol (AFP). ASP allows you to establish an asymmetrical session between an ASP workstation application and an ASP server application. The primary use of ASP is to provide services for the AppleTalk Filing Protocol (AFP) that, in turn, provides all of the services necessary to access an AppleTalk AppleShare server. AFP allows a workstation on an AppleTalk network to access and manipulate files on an AFP file server, such as an AppleShare server. Because you can use the native file system to access an AFP server from a workstation, in most cases you should not need to use AFP directly.

To register your application with the LAP Manager so that you will be notified when an AppleTalk transition event occurs that can affect your application, and to define a transition event that your application causes to occur that can affect other applications, see the chapter “Link-Access Protocol (LAP) Manager.” This chapter also describes how to install a protocol handler as a client of the LAP Manager if your application processes 802.2 Type 1 packets.

To learn how to write data directly to an Ethernet, token ring, or Fiber Distributed Data Interface (FDDI) driver instead of using the AppleTalk protocol stack, see the chapter “Ethernet, Token Ring, and Fiber Distributed Data Interface.” This chapter also describes how to read data directly from an Ethernet driver.

To implement a special-purpose application that receives and processes AppleTalk packets in a custom manner instead of passing them directly on to a higher-level AppleTalk protocol for processing, see the chapter “Multinode Architecture.”

Because the AppleTalk network system includes both hardware and software—and because the software includes not only the AppleTalk protocol stack and the programming interfaces to it, but also file servers, print servers, internet routers, drivers for circuit card or network interface controllers, and so forth—the information in this book constitutes only a small part of the body of literature documenting AppleTalk.

For a detailed description of the AppleTalk protocol specifications, see *Inside AppleTalk,* second edition. For a complete description of the LAP Manager, EtherTalk, and other AppleTalk connections, see the *Macintosh AppleTalk Connections Programmer's Guide*. To learn how to install and operate an AppleTalk internet, see the *AppleTalk Internet Router Administrator's Guide* and the *AppleTalk Phase 2 Introduction and Upgrade Guide.* For an introduction to the hardware and software of an entire AppleTalk network, see *Understanding Computer Networks* and the *AppleTalk Network System Overview.* For information on designing circuit cards and device drivers for Macintosh computers, see *Designing Cards and Drivers for the Macintosh Family,* second edition.

What to Read

If you are new to AppleTalk, you should begin with the chapter "Introduction to AppleTalk." This chapter describes some basic networking concepts that pertain to AppleTalk, and it summarizes each of the AppleTalk protocols and features, suggesting possible uses for them.

The chapter also includes a section that provides an overview of the two execution modes that you can use to execute routines that belong to the AppleTalk protocol programming interfaces. Even if you are already familiar with AppleTalk, you should read this section.

Each of the remaining chapters is devoted to a separate AppleTalk protocol or feature. Most of the chapters are self-contained; unless otherwise stated, there are no dependencies on preceding or following chapters. However, in some cases you may find it helpful to familiarize yourself with the information in other chapters that address related protocols. In most cases, your application will use more than one protocol.

The higher-level protocols are described first, followed by the lower-level protocols and the interfaces to the hardware device drivers, and ending with the chapter that describes multinode architecture.

Chapter Organization

Most chapters in this book follow a standard general structure. For example, the chapter "Name-Binding Protocol (NBP)" contains these major sections:

- "About NBP." This section provides an overview of the Name-Binding Protocol and its features.
- "Using NBP." This section describes how to use the most common NBP functions, gives related user interface information, provides code samples, and supplies additional information. For example, the section describes how to register your application with NBP so that users and other applications can locate and contact your application. It also describes how to look up another application's address based on its name and how to cancel a pending NBP request that you have made.
- "NBP Reference." This section provides a complete reference to NBP by describing the constants, data structures, and routines that you use to gain access to the NBP services. Each routine description follows a standard format that gives the routine declaration; a description of every parameter; the routine result, if any; and a list of errors, warnings, and notices. Most routine descriptions give additional information about using the routine and include cross-references to related information elsewhere. Many of the AppleTalk programming interface routines use parameter blocks to pass information to and receive it from the software driver that implements the protocol. The parameter block data type is described in the data structures section, and any parameter block fields that are common to all the routines that use the parameter block are defined in that section. Fields particular to a routine, but not common to all routines, are described along with the routine to which they pertain.
- "Summary of NBP." This section shows the Pascal, C, and assembly-language interfaces for the constants, data types, and routines associated with NBP. It also lists the result codes.

Conventions Used in This Book

This book uses various conventions to present certain types of information. For example, parameter blocks are presented in a certain format so that you can scan them quickly.

Special Fonts

All code listings, reserved words, and the names of data structures, constants, fields, parameters, and functions are shown in Courier (`this is Courier`).

When new terms are introduced, they are in ***boldface.*** These terms are also defined in the glossary.

Types of Notes

There are several types of notes used in this book.

Note

A note formatted like this contains information that is interesting but possibly not essential to an understanding of the main text. The wording in the tag may say something more descriptive, such as "Calling `ReadPacket` and `ReadRest` when LocalTalk is the data link." (This example appears on page 7-19.) Notes with descriptive titles contain useful information about a particular aspect of the feature being described. ◆

IMPORTANT

A note like this contains information that is especially important. (An example appears on page 7-10.) ▲

▲ **WARNING**

Warnings like this indicate potential problems that you should be aware of as you design your application. Failure to heed these warnings could result in system crashes or loss of data. (An example appears on page 7-18.) ▲

Assembly-Language Information

Inside Macintosh provides information about the registers for specific routines like this:

Registers on entry

A0	Contents of register A0 on entry

Registers on exit

D0	Contents of register D0 on exit

In addition, *Inside Macintosh* presents information about the fields of a parameter block in this format:

Parameter block

→	`input1`	`Ptr`	Input parameter.
←	`output1`	`Ptr`	Output parameter.
↔	`inAndOut`	`Integer`	Input/output parameter.

Numerical Formats

Hexadecimal numbers are preceded by a dollar sign ($).

The numerical values of constants are shown in decimal, unless the constants are flag or mask elements that can be summed, in which case they are shown in hexadecimal.

Development Environment

The system software routines described in this book are available using Pascal, C, or assembly-language interfaces. How you access these routines depends on the development environment you are using. This book shows system software routines in their Pascal interface using the Macintosh Programmer's Workshop (MPW).

All code listings in this book are shown in Pascal. They suggest methods of using various routines and illustrate techniques for accomplishing particular tasks. However, Apple Computer, Inc., does not intend for you to use these code samples in your applications.

Developer Products and Support

APDA is Apple's worldwide source for over three hundred development tools, technical resources, training products, and information for anyone interested in developing applications on Apple platforms. Customers receive the quarterly *APDA Tools Catalog* featuring all current versions of Apple development tools and the most popular third-party development tools. Ordering is easy; there are no membership fees, and application forms are not required for most of our products. APDA offers convenient payment and shipping options, including site licensing.

To order products or to request a complimentary copy of the *APDA Tools Catalog*, contact

APDA
Apple Computer, Inc.
P.O. Box 319
Buffalo, NY 14207-0319

Telephone	800-282-2732 (United States) 800-637-0029 (Canada) 716-871-6555 (International)
Fax	716-871-6511
AppleLink	APDA
America Online	APDA
CompuServe	76666,2405
Internet	APDA@applelink.apple.com

If you provide commercial products and services, call 408-974-4897 for information on the developer support programs available from Apple.