

This chapter describes the Apple Open Collaboration Environment and the Macintosh system software components that compose the AOCE APIs. It provides an introduction to the capabilities of the AOCE software and some of the ways you can use AOCE technology to enhance your application or to solve specific problems in workflow and collaboration among people and computers.

Whereas the other chapters of this book are intended for programmers and application developers with a working knowledge of the C programming language, this chapter is also of interest to anyone who wants to gain a deeper understanding of the capabilities and uses of the AOCE technology and the PowerTalk system software. To read this chapter you should be familiar with the fundamentals of programming for the Macintosh computer and the basic concepts of interapplication communications. For introductions to these topics, see *Inside Macintosh: Overview* and *Technical Introduction to the Macintosh Family*. It will also be helpful if you have spent some time using the PowerTalk software on your own computer.

This chapter begins with a brief statement about the relationship of the AOCE software to the rest of the Macintosh Operating System. It then presents several scenarios describing possible uses of the AOCE technology to solve collaboration and workflow problems for a mythical company. The third section of this chapter describes each of the components of the AOCE system software in some detail. The final section introduces some basic concepts used throughout the remaining chapters in this book.

About AOCE System Software

The AOCE system software adds to the capabilities of the Macintosh Operating System and of the Finder. To the Macintosh Operating System, the AOCE software adds a transport-independent messaging service that applications can use for mail and for interapplication communications. Unlike the Program-to-Program Communications (PPC) Toolbox, the AOCE Interprogram Messaging (IPM) service does not require that both the sending and receiving applications be simultaneously connected to a network. In fact, IPM does not require that the communication ends be connected to a network at all: it operates over modems and can work over any other message-transport mechanism for which a developer cares to write an interface.

Although collaboration among users and applications implies the existence of some messaging service, the AOCE system software supports collaboration in other ways as well: it provides catalogs, which store not only addresses but any sort of data you wish to put in them; it authenticates the identities of the sender, routers, and receivers of messages; and it allows users and applications to guarantee the identity of the sender of a message and the integrity of the data in a message by affixing a digital signature to the data.

The AOCE software extends the capabilities of the Finder by adding a universal mailbox capable of holding every sort of electronic mail received by the user: E-mail, voicemail, faxes, and so forth. It also provides a familiar folder-based interface that allows users to

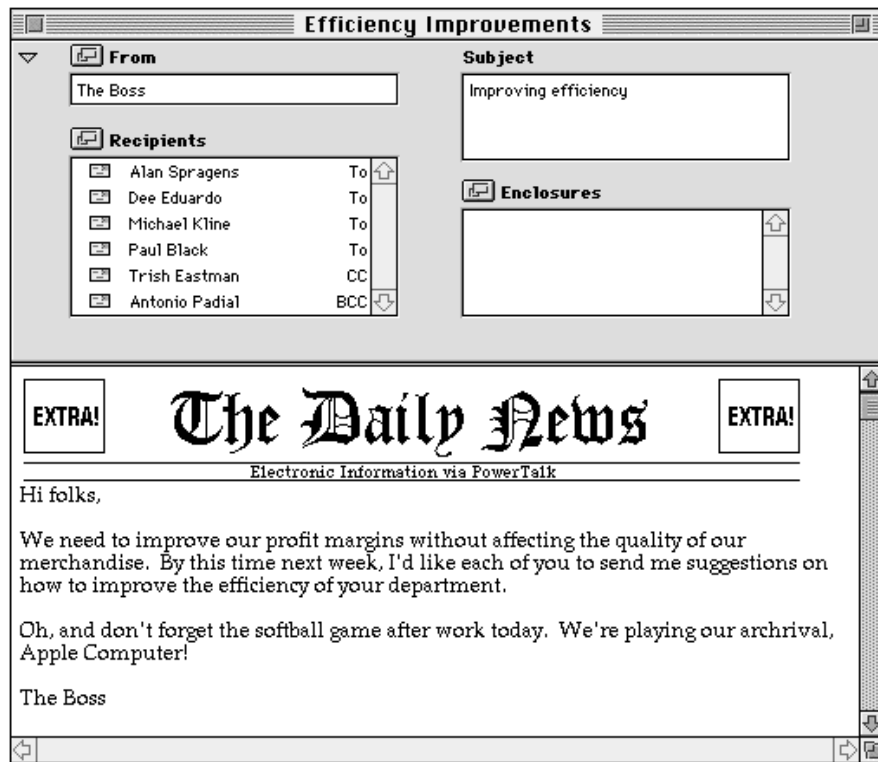
Introduction to the Apple Open Collaboration Environment

browse the contents of AOCE catalogs. Developers can extend both the types of mail that the mailbox can receive and the types of catalog data that the Finder can display.

The following sections illustrate these uses of the AOCE software and describe the components of the AOCE software in more detail.

Some Uses for AOCE Software

To understand some of the ways you can use AOCE software to improve productivity and workflow efficiency, consider a hypothetical situation. Suppose that the president and CEO of the River Change Systems company wants to improve the efficiency of her company's operations. She uses her favorite word processor, SurfWriter, to write a memo on her Macintosh computer to all of her department heads soliciting suggestions. Rather than printing the memo and using the company mail service to deliver it, she simply adds a PowerTalk mailer to the memo (Figure 1-1), turning it into a PowerTalk letter. She then drags the addresses for her department heads from her personal catalog on her Macintosh desktop, drops them into the mailer, and chooses Send from the Mail menu to send the letter over her company's AppleTalk network. Not all of the department heads at the River Change Systems company have SurfWriter on their Macintosh computers, but those who don't can read the memo using the AppleMail letter application that is provided with the PowerTalk system software.

Figure 1-1 Letter containing an AOCE mailer

Here are some of the ideas that the department heads come up with.

The Company Store Catalog

The River Change Systems company has over 12,000 employees worldwide, and they make extensive use of the company store, buying gift and office items for themselves and friends. The company store stocks hundreds of items and prints a full-color catalog four times a year as well as numerous notices and announcements of sales and special events. Overseas employees of the company sometimes have to wait weeks for the delivery of their catalogs, occasionally missing out on limited-stock items. Although the store grosses nearly \$500,000 a year, it barely breaks even because of its low profit margins.

The manager of the company store suggests a PowerTalk-based system that will replace the company store's printed catalog and integrate it with the store's inventory-tracking system. This is how the system works:

The paper catalog is replaced by a server-based AOCE catalog. The basis for the catalog is the store's existing inventory database, which was created using the SurfDB database application. A catalog service access module (CSAM) in each user's computer interfaces the SurfDB database server to that user's PowerTalk system. Each item in the catalog

Introduction to the Apple Open Collaboration Environment

appears as an icon in a folder on the employee's desktop when he or she logs on to the PowerTalk system. Overseas employees, who are connected via microwave links to the company's extended AppleTalk network, have the same access to the store's catalog as anyone else. Even employees working at home or using Macintosh PowerBook computers while traveling can dial in to the network by using the AppleTalk Remote Access application.

When an employee double-clicks an item in the catalog, a window appears that contains a stack of PowerTalk information pages displaying information about the item, including a picture (Figure 1-2). Because the catalog draws its information from the store's inventory database, the catalog includes information about whether the item is in stock, how many are available, whether it is backordered if not currently available, and—depending on the type of item—such data as what sizes and colors are in stock. One of the information pages in the stack is an order form.

Figure 1-2 Catalog-item information page

The image shows two overlapping windows from a Macintosh-style graphical user interface. The top window, titled "Recording Albums", displays a list of six items with their names and kinds. The bottom window, titled "Wisdom of Youth", shows detailed information for a specific album, including a photo of a young boy, the full title, artist, format, and a comments section.

Recording Albums		
6 items (all)	61.6 MB in disk	13.4 MB available
Name	Kind	
<input type="radio"/> Collected Wisdom	album collection	
<input type="radio"/> Great Sayings of the Boss	album	
<input type="radio"/> River Change Company S...	album	
<input type="radio"/> The Boss' Greatest Hits	album	
<input type="radio"/> Wisdom of the Boss	album	
<input checked="" type="radio"/> Wisdom of Youth	album	

Wisdom of Youth

General Info ▼

Full title: Wisdom of Youth
Artist: Sam
Format: ☒ CD ☐ Cassette ☐ Vinyl
Comments:
 Clever and pithy sayings by the boss' 3-year-old son, Sam

When employees want to order items from the catalog, they fill out the order form on the screen. To fill in the name and address fields, they can drag the information from a personal catalog or a PowerTalk information card on their desktop and drop it on the order form. They can do the same thing to add an address of a third party for delivery of a gift. For payment, employees can authorize deductions from their paychecks or can use credit cards. In either case, they must affix a digital signature to the order form to guarantee the authenticity of the order.

When the employee clicks the Send button on the order form, the code resource associated with the information page sends the signed order form to the company store and updates the database to indicate the sale. It also sends a message to the company's accounting database informing it of the sale, the amount due, and the form of payment. In the case of a payroll deduction, the accounting database automatically reduces the employee's paycheck by the indicated amount and marks the account as paid. If the employee used a credit card, the database sends a message to the credit card bank requesting payment.

From the time the employee sends the order, no human interaction is necessary to complete the financial transaction. Once the payment has been approved, the accounting database application sends a PowerTalk message to the company store warehouse, where the order, shipping list, and mailing label are automatically printed out.

In addition to this catalog-based system, the company store uses PowerTalk to send announcements of sales and special events to all the employees. The company store marketing director uses SurfWriter to create the announcement, adds a PowerTalk mailer, and sends it to several group addresses that include all the employees of the company. Each announcement is sent in both the SurfWriter native format and the AppleMail image format, so each employee can read the letter whether or not that employee has the SurfWriter application.

Purchasing

The manager of the purchasing department suggests the use of PowerTalk to automate the routing of purchasing requests. An employee wishing to make a purchase opens the company's forms application and chooses New Form from the File menu. This opens a catalog-selection dialog box that lets the user select the proper form from a PowerTalk catalog. The application displays the form, and the employee fills it in and adds a digital signature.

When the employee clicks a Send button at the bottom of the form, the application automatically looks up the AppleTalk address of the employee's manager in a PowerShare catalog and sends the form to that manager. The form appears as a forms application file in the manager's PowerTalk mailbox. The manager reads the form, approves or disapproves it, adds a digital signature, and clicks Send. At each routing step the forms application sends dated messages in AppleMail letter format to the requestor and to the purchasing department indicating that a request has been made, who made it, who has last forwarded it, and to whom it has most recently been sent. If at any point the request is denied, the forms application sends a letter, with a copy of the form attached, informing the original requestor that the request has been denied, when it was denied, and who denied it.

When the fully approved form arrives at purchasing, the requested item can be purchased. If the vendor is also using PowerTalk, the purchasing department can forward the purchase order over the network or by modem and arrange for the electronic transfer of funds from one bank to the other. The entire transaction can be completed without printing any paper. What's more, a digital signature guaranteeing the originator and integrity of the request has been added at each step. If the request is

Introduction to the Apple Open Collaboration Environment

delayed at any point in the routing, the purchasing department can check the latest routing report to determine who is holding it up and for how long.

Sales

The large sales staff of the River Change Systems company have been losing sales when they miss messages either because they are on the road or because a salesperson neglects to check all of the different types of electronic messages that can come in each day: voicemail, faxes, AppleLink, and internal E-mail. In addition, sales staff need some way to check inventory and enter orders while on the road.

By adding PowerTalk to their systems, each salesperson can gather all of his or her messages in a single mailbox, which the salesperson can check whether in the office or on the road. In addition, by using a PowerTalk CSAM to access the company's inventory database, the salesperson can check inventory and place orders at any time. Each salesperson can also use a personal PowerTalk catalog stored on his or her notebook computer to keep track of accounts and to store addresses, phone numbers, and personal information about clients.

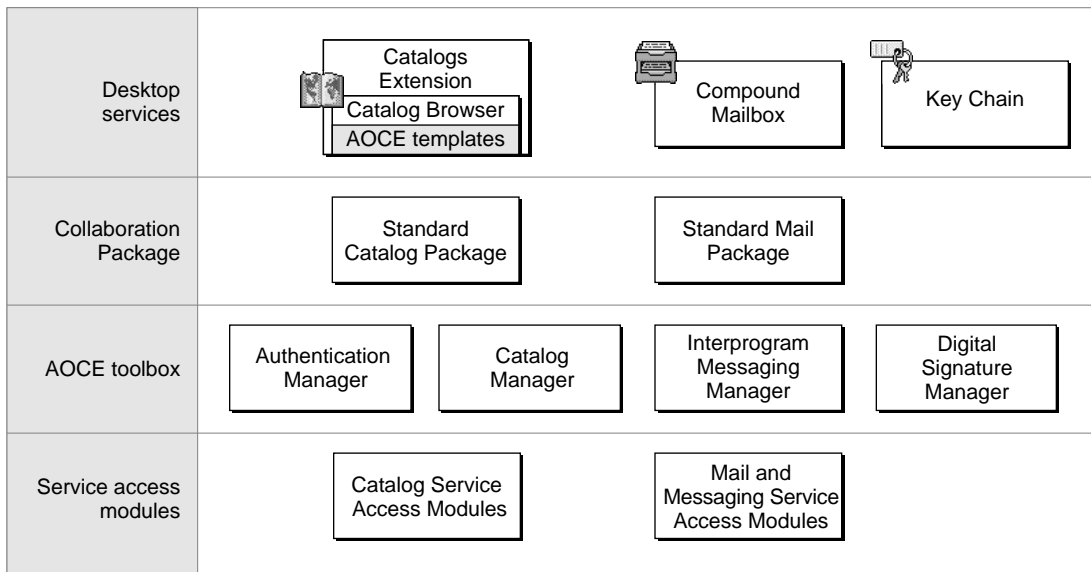
The Components of the AOCE Software

Rather than suggesting more uses for the AOCE technology, at this point it might be more useful to describe the components that constitute the AOCE software and to show how each of the features or functions in the foregoing examples is implemented with AOCE APIs.

At the heart of AOCE technology are three basic services: messaging, authentication, and catalogs. These are enhanced by an independent module, Digital Signatures, plus an extensible version of the Finder. The AOCE software provides APIs for each of these components: the Interprogram Messaging Manager, the Authentication Manager, the Catalog Manager, the Digital Signatures Manager, and the AOCE template mechanism for extending the Finder. In addition, there are two high-level AOCE APIs that make it very easy to add mail and catalog services to existing applications: the Standard Mail Package and the Standard Catalog Package.

To allow the AOCE system software to work with external databases and messaging systems, the AOCE technology also includes interfaces for catalog service access modules (CSAMs) and mail and messaging service access modules (MSAMs).

Figure 1-3 shows all of the components of the AOCE software.

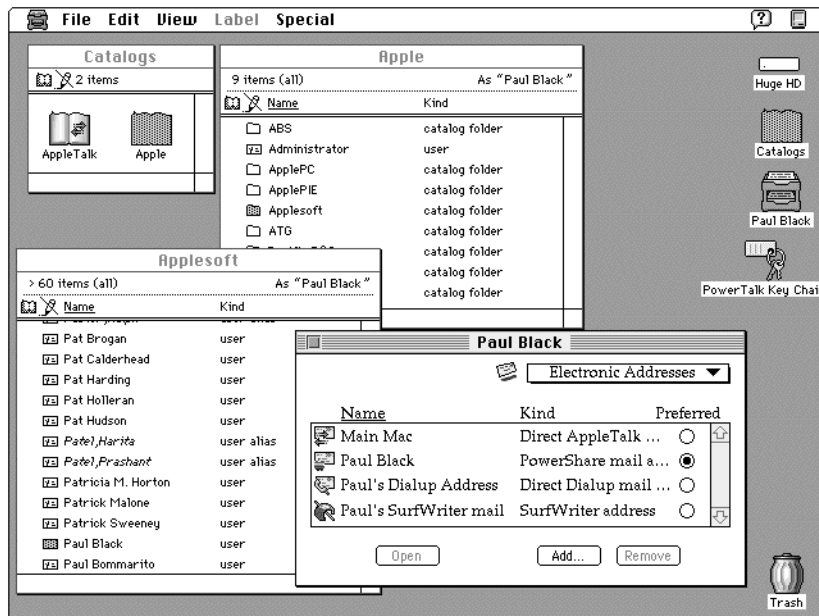
Figure 1-3 The components of the AOCE software

Desktop Services

The PowerTalk system software includes extensions to the desktop services offered by System 7. The user adds and configures mail and catalog services through the PowerTalk Key Chain. If you provide a CSAM or MSAM, you will also have to provide resources for use by the Key Chain as described in the chapter “Service Access Module Setup” in *Inside Macintosh: AOCE Service Access Modules*. All of the forms of electronic mail coming in from the messaging systems that the user has added to his or her Key Chain appear in the compound mailbox, which appears on the user’s desktop as the Mailbox icon. There is no application interface to the mailbox. To learn how to interface an external mail or messaging system to the PowerTalk messaging system, see *Inside Macintosh: AOCE Service Access Modules*.

The AOCE Catalogs Extension (CE) to the Finder includes the Catalog Browser and AOCE templates. The **Catalog Browser** is a Finder extension that allows a user to search through an AOCE catalog by opening folders on the desktop. An **AOCE catalog** is a hierarchically arranged store of data. AOCE catalogs include PowerTalk server-based catalogs, personal catalogs, and external databases interfaced to the AOCE catalog system through a catalog service access module (CSAM). Personal catalogs are stored as hierarchical file system (HFS) files on the user’s computer. AOCE catalogs are described in more detail in “Catalogs, Records, and Attributes” on page 1-14.

Viewed through the catalog browser, an AOCE catalog appears to be a folder, analogous to a volume in HFS, that contains *catalog folders*, analogous to HFS folders, and *records*, analogous to HFS files. Each record can contain one or more blocks of data, called *attribute values*. Figure 1-4 shows a desktop with catalogs, catalog folders, and record icons displayed.

Figure 1-4 The Catalogs Extension to the Finder in use on a desktop

Just as with HFS files, the user can open an AOCE catalog record to view its contents. Unlike HFS, however, which calls on the application that created the file to open it, it's the Finder itself that opens a catalog record. When the user opens a record on the desktop, the Catalogs Extension (CE) to the Finder displays a special window called an *information page window*. The information page window for a record might contain a list of attribute values; if so, the user can open each attribute value in the list to see what it contains, displayed in another information page window.

An information page window contains one or more *information pages*, each of which contains information about a record or about one of its attributes. This information can include a list of attributes in a record, information derived from attributes, and controls such as buttons and checkboxes. Information pages can also contain editable text and pictures. When the user makes changes to the information in an information page and closes the information page, the CE makes corresponding changes to the attribute values in the record. Figure 1-2 on page 1-6 shows an information page window.

Because there are no restrictions on the type of data that you can put in a record, the AOCE software includes a mechanism for designing and implementing new information pages to display new types of data (or for displaying old types of data in new ways). To create new information pages and to interface them with AOCE catalog records and attributes, you write resource files known as *AOCE templates*.

In the example in "The Company Store Catalog" beginning on page 1-5, AOCE templates are used extensively by the company store catalog to display the information in the store's database. On the user's desktop, the store's database appears as an AOCE catalog folder, containing a number of catalog folders with labels such as "Rubber ducks," "Other bath toys," and "Recording albums." Each such folder contains one

AOCE catalog record for each item in the company store catalog; for example, the “Rubber ducks” folder contains a little over 40 records, each containing information about a different variety of rubber duck.

When an employee opens a catalog-item record, an information page displays information that is stored as attributes in the record: a picture of the item, its price, the number in stock, and so forth. The order-form information page contains text-entry fields and controls that let the employee order the item. For each of these information pages, the company’s programmers have supplied a set of AOCE templates to define the contents and appearance of the information page and to implement the functions that it performs.

For detailed information about AOCE templates, see the chapter “AOCE Templates” in this book. For more information about the structure of AOCE catalogs, see the chapter “Catalog Manager” in this book.

Collaboration Package

The AOCE Collaboration Package consists of two high-level APIs: the Standard Mail Package and the Standard Catalog Package. The Standard Mail Package provides two interfaces: the send-letter routines and the mailer routines.

Standard Mail Package

The send-letter routines send a file created by an application to a user’s mailbox. In the example in “Purchasing” beginning on page 1-7, the forms application uses the send-letter routines to send purchase-order status reports to the originator of the purchase order and to the purchasing department.

The mailer routines allow you to add a mailer to any of your application’s documents. A mailer appears as a new region at the top of your document’s window (or you can put one in its own window, if you wish) that provides addressing and subject fields. The user can also add a digital signature to a letter containing a mailer. Adding a mailer to a document turns the document into a letter. The AOCE software delivers a letter you mail this way into the recipients’ PowerTalk mailboxes. When the recipient double-clicks the letter, the Finder opens the application that was used to create it. If the recipient does not have that application, and if you chose to send an image version or a standard AppleMail format version of the document, the user can still open the document using the AppleMail application.

In the example, the president of the River Change Systems company uses a SurfWriter application document with a PowerTalk mailer to send a memo to all department heads, and the company store uses this application to send announcements to employees. In both cases, the letter includes an image or AppleMail format version so that recipients who do not have SurfWriter can still open it.

Standard Catalog Package

The Standard Catalog Package makes it easy for you to provide dialog boxes that let the user browse and search PowerTalk catalogs from within your application. In the example in “Purchasing” on page 1-7, the forms application provides a dialog box to the user to allow the selection of a form from a PowerTalk catalog that contains forms. Because you can restrict the dialog box to display only records of specific types, the catalog can contain many types of records in addition to those containing forms, but the forms application presents only the selection of forms to the user.

Digital Signature Manager

The Digital Signature Manager allows you to add a digital signature to a file or any portion of a file. A *digital signature* is an encrypted number that is associated with a particular set of data. The digital signature guarantees the identity of the individual or entity (for example, the company) that signed the data, and it ensures the integrity of the data. A digital signature cannot be forged, and the signed data can not be altered without invalidating the signature.

Users can use a mailer to add a digital signature to a document in an application that uses the Standard Mail Package to add mailers to its documents. They can also use the DigiSign utility program to sign any file. You can use the Digital Signature Manager to allow users to sign any data. In the example in “The Company Store Catalog” beginning on page 1-5, the AOCE template that provides data for the order-form information page calls the Digital Signature Manager to sign the order form to authorize a credit card purchase or payroll deduction. In “Purchasing” on page 1-7, the forms application requires the person submitting, approving, or disapproving the purchase order to add a digital signature. Because the signature is being added from within the application, the application itself must call the Digital Signature Manager.

Collaboration Toolbox

The fundamental services of AOCE are handled by three system software managers: the Authentication Manager, the Catalog Manager, and the Interprogram Messaging (IPM) Manager. Although you can call each of these managers independently to make use of its services, they work together to support a variety of server-based and workstation-based collaboration services. Together with the Digital Signature Manager, these managers are referred to in this book as the *AOCE toolbox*. The Authentication, Catalog, and IPM Managers are sometimes referred to as the *Collaboration toolbox*.

Authentication Manager

The Authentication Manager provides services to the other Collaboration toolbox managers and to the PowerShare servers. The Authentication Manager allows each end of a communications connection to determine that the other end is who it claims to be. Every time a user provides a password to log on to PowerTalk or to connect to a

PowerShare collaboration server, the Authentication Manager provides credentials to the AOCE software that verify the identity of the user.

Very few applications need to call the Authentication Manager directly. However, if you want to authenticate the connection ends of an AppleTalk Secure Data Stream Protocol (ASDSP) connection or if you want to authenticate some other type of connection outside of the AOCE messaging service, you can use Authentication Manager functions to do so. In any case, it is important that you understand the concepts of authentication and the credentials (known as *identities*) returned to your program by the Authentication Manager; these concepts are discussed in “Authentication and Authentication Identities” beginning on page 1-17. ASDSP is described in the chapter on ADSP in *Inside Macintosh: Networking*.

Catalog Manager

The AOCE Catalog Manager provides an interface to AOCE catalogs (see “Desktop Services” on page 1-9 for a general description of AOCE catalogs). The AOCE Catalog Manager lets you

- get information about AOCE catalogs and catalog nodes
- create, open, and close personal catalogs
- manage the organization of an AOCE catalog
- manage the contents of an AOCE catalog
- get information about access controls for a catalog and the contents of a catalog

The AOCE Finder extension allows users to browse catalogs, and the Standard Catalog Package lets you display dialog boxes that let the user browse catalogs and search for specific records. If you want to get information from a catalog or make changes to a catalog directly from your application or without user interaction, you can use the functions provided by the Catalog Manager.

In the example “Purchasing” on page 1-7, the forms application uses the Catalog Manager to find in a PowerShare catalog the name and address of the next person to whom the form should be routed. No user interaction is therefore necessary to route the form.

Interprogram Messaging Manager

The Interprogram Messaging (IPM) Manager provides a messaging service that is used by the AOCE software and that you can use for your own purposes. The IPM Manager works with servers (such as the PowerShare mail servers) and without servers. It can send messages over an AppleTalk network, over other networks (through mail and messaging service access modules, or MSAMs), and even over telephone lines through modems. The IPM Manager provides *store-and-forward messaging*. In store-and-forward messaging, the receiver does not have to be available to receive the message at the time it is sent; the IPM Manager stores the message and sends it on when the receiver is available.

Introduction to the Apple Open Collaboration Environment

Whereas you can use the Standard Mail Package to add mail capabilities to your application and to send any file as an attachment to a letter, you can use the IPM Manager to send mail or non-mail messages. The distinction between mail and non-mail messages is that mail involves the transmission of a message that is intended to be read by a person, whereas non-mail messages are intended to be used by an application.

In the example “The Company Store Catalog” beginning on page 1-5, when the employee clicks the Send button on the order form, the code resource associated with the information page uses the IPM Manager to send the signed order form to the company store and to send a message to the company’s accounting database informing it of the sale, the amount due, and the form of payment. The signed order form is intended to be read by people and so is sent as electronic mail, but the message to the accounting database contains commands intended to be used only by the database application and so is sent as a non-mail message.

Service Access Modules

Developers can provide software modules, called service access modules (SAMs) that link the AOCE Collaboration toolbox to external mail and messaging services or databases. Two types of mail and messaging SAMs can be written: server-based MSAMs and personal MSAMs. A server-based MSAM resides on the same computer as a PowerShare mail server and makes an external mail or messaging system available to all users of that PowerShare server. A personal MSAM resides on an individual user’s computer and makes an external mail or messaging system available through the AOCE software for only that user. In either case, the user sends and receives mail from the external system through the PowerTalk mailbox, and applications that send or receive AOCE messages can send and receive these messages via the external messaging system.

All catalog service access modules (CSAMs) are in the form of device drivers located on the user’s computer. A CSAM allows a non-AOCE database to appear to users and applications to be an AOCE catalog. In the example “The Company Store Catalog” beginning on page 1-5, each user of the company catalog has a CSAM that interfaces the SurfDB database server to that user’s PowerTalk system.

AOCE Concepts

This section describes some of the basic concepts and structures that appear frequently in this book. All of these topics are discussed in more detail in the succeeding chapters in this book, but they are all crucial to an understanding of how the AOCE software works and how you can use it, and you might find an early introduction to these topics helpful.

Catalogs, Records, and Attributes

As discussed in “Desktop Services” beginning on page 1-9, an AOCE catalog is a hierarchically arranged store of data. The root level of a catalog appears on the desktop

as a folder with the name of the catalog. The root-level catalog folder can contain any number of folders (also known as catalog nodes or *dNodes*) and records, and each contained folder can itself contain folders and records. (Note that in some AOCE catalogs, such as personal catalogs, the root-level folder contains only records, not other folders.) The catalog records contain the data and take the place in the catalog hierarchy occupied by files in the hierarchical file system (HFS). Accordingly, records cannot contain folders or other records.

The data in an AOCE catalog record is stored as *attribute values*, which cannot exceed 65,536 bytes in size. Each attribute value has an *attribute value tag*, which specifies the format of the data in the attribute value. Attribute values are grouped according to *attribute type*. When you add an attribute value to a record, the Catalog Manager assigns it an *attribute creation ID*, which is unique within the record. The combination of an attribute type, attribute value, attribute value tag, and attribute creation ID is referred to as an *attribute*.

Although an attribute value is the smallest unit of data storable in a catalog, an AOCE template can parse the data in an attribute value and display any portion of that data in a manner meaningful to the user. When the user edits the data in an information page and closes the information page, the CE uses the data-parsing pattern in the template to determine how to update the attribute value and saves the new value in the record.

You can call Catalog Manager functions to create and delete records and to add, modify, and delete attributes within records. You can call Standard Catalog Package functions to provide dialog boxes that allow users to browse and search catalogs. You can write AOCE templates that tell the Finder how to display the contents of records and attributes.

Messaging and Message Queues

The AOCE Interprogram Messaging Manager provides store-and-forward messaging; that is, when you send a message, the IPM Manager stores the message until it is able to forward it to its destination. Although PowerShare mail servers and other AOCE-compatible messaging servers can store messages on the server computer so that it is never necessary for the sending computer and receiving computer to be online simultaneously, the IPM Manager also provides a store-and-forward messaging service for messages not sent through servers. To achieve this end, the IPM Manager maintains an output queue for messages on the computer of the sending application and an input queue on the computer of the receiving application.

Suppose, for example, that you used a PowerTalk mailer to send a SurfWriter application document to someone at an AppleTalk address. If the recipient's computer was not connected to the network at the time you sent the message, the IPM Manager would place the message in your output queue. The IPM Manager would then check AppleTalk periodically to determine whether the recipient computer was present on the network. When the recipient's computer comes online, the IPM Manager in the sender's computer forwards the letter, and the IPM Manager in the recipient's computer places it in the input queue. Because in this example the message is a letter, it appears in the recipient's mailbox. Note that it is not necessary for the application used to create the message—

Introduction to the Apple Open Collaboration Environment

SurfWriter in this case—to be running on either computer at the time the IPM Manager forwards the letter from the sender’s output queue to the recipient’s input queue.

The IPM Manager provides default input queues for mail and for non-mail messages. In addition, your application can create additional input queues on the local computer for its own purposes. Managing such queues is somewhat complicated, however, and the AOCE software provides no protocol for determining the name of a nonstandard input queue; thus, you should create your own queues only when you have a clear need to do so.

The administrator of an AOCE messaging server can create additional input queues on the server computer and typically creates one queue for each user who has an account on that server.

There is no facility for creating additional output queues; the IPM Manager opens and maintains a single output queue on each user’s computer for all mail and messaging purposes.

Because several applications might be using the same default input mail and messaging queues for different purposes, the IPM Manager provides the ability to filter the message list by a variety of criteria when you enumerate a queue.

Queues are described in detail in the chapter “Interprogram Messaging Manager” in this book.

Addressing Mail and Messages

Each AOCE message or letter must be delivered to a specific input queue. In many cases, the AOCE software takes care of addressing the message or letter for you, as when the user drags an address from a catalog and drops it in a mailer. In other cases, an AOCE function returns an address to which you can send a message or letter. For example, you can use the Standard Catalog Package to let the user select a user record. The Standard Catalog Package returns to you the catalog specifier (DSSpec structure) that identifies the record the user selects. You can then provide that catalog specifier to the IPM Manager as the address for a message. The IPM Manager extracts the name and address of the default messaging queue for that user from the specified record and routes the message accordingly.

This last example illustrates the use of *indirect addressing*. Indirect addressing lets you specify the entity (the user or application, for example) to which you want the message sent without specifying—or even having to know—the actual queue name or the location of the queue. You do this by specifying a catalog record (and, optionally, the attribute within that record) that contains the address. (If you use the user record type and do not specify the attribute, the IPM Manager uses the default mail or messaging queue, as in the preceding example.)

You can also use indirect addressing to send a letter or message to a group. In the simplest use of group addresses, the record you specify as an address is a group record containing the record identifiers of the user records of the members of the group. The AOCE software takes care of resolving the addresses and routing the message. You can

define your own record type in which to store group members' addresses and resolve group addresses yourself, but that requires a great deal more work on your part.

You can also use *direct addressing*, specifying the actual queue name plus the AppleTalk address, modem string, or other exact information needed by the IPM Manager to deliver the message. To use direct addressing, you must have prior knowledge of the queue name and location of the recipient, or you must develop your own protocol, outside of the AOCE APIs, for determining this information.

Addressing is discussed in detail in the chapter “Interprogram Messaging Manager” in this book.

Authentication and Authentication Identities

The AOCE Authentication Manager works together with the PowerShare collaboration servers to verify the identity of the requestor of a messaging or catalog service. Each AOCE message header includes an authentication field that you can use to determine whether the message or letter has been authenticated. If the value of this field is `true`, you can have a very high level of confidence that the name in the sender field is genuine.

In order for a message (or letter) to be authenticated, the sender of the message and every PowerShare server used to route the message has to have provided a valid password when logging on to the AOCE system. In addition, the recipient of the message has to provide a password before opening the mailbox or reading messages from an input queue. In the case of mail, the user can determine whether a letter is authenticated by looking for a seal next to the name of the sender in the In Tray. In Figure 1-5, for example, the letter titled “Dialup Address” is authenticated, whereas the letter titled “Re> AOCE Templates review...” is not.

Figure 1-5 An In Tray with certified (that is, authenticated) and uncertified letters

✓	Subject	Sender	<u>Date Sent</u>	Location	Priority
	Fwd> Re> Re: Report handlin...	Michael Bayer	12/3/93, 2:04 PM	remote	normal
	Hi	Main Mac	11/28/93, 8:42 AM	local	normal
✓	Hi	Main Mac	11/22/93, 10:04 AM	local	normal
✓	Dialup GM Candidate	Carol Lee	11/19/93, 1:43 PM	remote	normal
	Re>> Dialup Address	Carol Lee	11/10/93, 3:34 PM	remote	normal
✓	Dialup Address	Carol Lee	11/10/93, 11:39 AM	remote	normal
✓	Re> AOCE Templates review...	Harry Chesley	11/8/93, 12:16 PM	local	normal
✓	Re>>> Address template	John Evans	11/8/93, 11:27 AM	local	normal
✓	AOCE Templates review me...	Paul Black	11/8/93, 11:26 AM	local	normal
✓	Re> Address template	John Evans	11/8/93, 10:49 AM	local	normal

Note that only messages sent entirely through PowerShare servers can be authenticated by the AOCE system; messages sent through MSAM servers or by way of a serverless connection (over telephone lines, for example) cannot be authenticated by the AOCE

Introduction to the Apple Open Collaboration Environment

software. However, you can use Authentication Manager functions to implement your own authentication system for such messages. Note also that authentication guarantees only the identity of the sender, not the integrity of the data in the message itself. To guarantee that the content of a message has not been altered, use the Digital Signature Manager (see “Digital Signature Manager” on page 1-12).

When an entity (user, server, or application) provides a valid name and password to the Authentication Manager, the Authentication Manager returns a number known as an *authentication identity*. Each entity requesting a service from the IPM Manager, Catalog Manager, Standard Mail Package, or Standard Catalog Package then has to provide the authentication identity each time it requests a messaging, mail, or catalog service.

The IPM Manager uses the authentication identity to authenticate a message or letter. The Standard Mail Package goes a step further, using the authentication identity to determine the sender of a letter and to fill in the From field in the mailer. The Catalog Manager and Standard Catalog Package use the authentication identity for a different purpose entirely; they use the identity to determine whether the requestor of a catalog service has the authority to make a specific request. For example, the user who “owns” a user record (that is, the user to whom that record was assigned by the catalog administrator) normally has the authority to change personal data within that record. Your application, by providing that user’s authentication identity to the Catalog Manager, could then alter attribute values in that user record on behalf of that user. You could not, however, alter data in some other person’s user record, because the Catalog Manager would not grant you access to do so based on the authentication identity you supplied.

The Catalog Manager provides access controls for catalogs, dNodes, records, and attribute types. Access controls are described in detail in the chapter “Catalog Manager” in this book.

If you have used the PowerTalk software, you are probably already familiar with the concept of the PowerTalk Key Chain. Each time the user adds a service to the Key Chain, the AOCE software saves in a special personal catalog an encrypted form of the name and password the user provides. In this book this special catalog is referred to as the *PowerTalk Setup catalog*. The AOCE Authentication Manager takes the user’s name and password and creates a number called the *local identity*. In most cases, it is the local identity that you provide to an AOCE toolbox function as the authentication identity when requesting a service.

Suppose a user has accounts on two different PowerShare collaboration servers. The user adds the name and password for each of these accounts to the PowerTalk Key Chain. When the user turns on the computer in the morning and opens the PowerTalk mailbox, the PowerTalk software requests the user’s access code, which is the password that provides access to the Setup catalog. The user’s PowerTalk software then uses the information in the Setup catalog to obtain authentication identities, referred to in this book as *specific identities*, to obtain services from the PowerShare servers listed in the Key Chain.

When you need to provide an authentication identity to an AOCE function, you can provide either a local identity or a specific identity. When you provide the local identity,

Introduction to the Apple Open Collaboration Environment

the AOCE software uses the information in the Setup catalog to obtain the specific identity it needs to carry out the request. In most cases, you should provide the local identity to any AOCE function that requires an authentication identity. You should provide a specific identity only when it is absolutely necessary to do so, such as when you are using a background interapplication messaging service that is not visible to the user (and that has therefore not been added to the Key Chain), or when the user is not the owner of the computer and does not know the owner's access code.

The `SDPPromptForID` function (described in the chapter "Standard Catalog Package" in this book) displays a dialog box requesting a name and password and returns an authentication identity (either local or specific). If the user has already logged on, you can use the `AuthGetLocalIdentity` function (see the chapter "Authentication Manager" in this book) to obtain the local identity.

