

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

UBISOFT, INC.,
Petitioner,

v.

UNILOC USA, INC. AND UNILOC LUXEMBOURG S.A.,
Patent Owners.

U.S. Patent No. 6,489,974

DECLARATION OF SCOTT BENNETT, Ph.D.
21 July 2017

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. BACKGROUND AND QUALIFICATIONS.....	1
III. PRELIMINARIES.....	3
IV. OPINIONS REGARDING INDIVIDUAL DOCUMENTS.....	11
V. ATTACHMENTS	13
VI. CONCLUSION	14

I, Scott Bennett, hereby declare under penalty of perjury:

I. INTRODUCTION

1. I have personal knowledge of the facts and opinions set forth in this declaration, I believe them to be true, and if called upon to do so, I would testify competently to them. I have been warned that willful false statements and the like are punishable by fine or imprisonment, or both.

2. I am a retired academic librarian working as a Managing Partner of the firm Prior Art Documentation LLC at 711 South Race Street, Urbana, IL, 61801-4132. Attached as Appendix A is a true and correct copy of my Curriculum Vitae describing my background and experience. Further information about my firm, Prior Art Documentation Services LLC, is available at www.priorartdocumentation.com.

3. I have been retained by Erise IP, PA, to authenticate and establish the dates of public accessibility of certain documents in an *inter partes* review proceedings for U.S. Patent No. 6,489,974. For this service, I am being paid my usual hourly fee of \$91/hour. My compensation in no way depends on the substance of my testimony or the outcome of this proceeding.

II. BACKGROUND AND QUALIFICATIONS

4. I was previously employed as follows:

- University Librarian, Yale University, New Haven, CT, 1994-2001;

- Director, The Milton S. Eisenhower Library, The Johns Hopkins University, Baltimore, MD, 1989-1994;
- Assistant University Librarian for Collection Management, Northwestern University, Evanston, IL, 1981-1989;
- Instructor, Assistant, and Associate Professor of Library Administration, University of Illinois at Urbana-Champaign, Urbana, IL, 1974-1981; and
- Assistant Professor of English, University of Illinois at Urbana-Champaign, 1967-1974.

5. Over the course of my work as a librarian, professor of English, researcher, and author of nearly fifty scholarly papers and other publications, I have had extensive experience with catalog records and online library management systems built around Machine-Readable Cataloging (MARC) standards. I also have substantial experience in authenticating printed documents and establishing the date when they were accessible to researchers.

6. In the course of more than fifty years of academic life, I have myself been an active researcher. I have collaborated with many individual researchers and, as a librarian, worked in the services of thousands of researchers at four prominent research universities. Over the years, I have read some of the voluminous professional literature on the information seeking behaviors of

academic researchers. And as an educator, I have a broad knowledge of the ways in which students in a variety of disciplines learn to master the bibliographic resources used in their disciplines. In all of these ways, I have a general knowledge of how researchers work.

III. PRELIMINARIES

7. *Scope of this declaration.* I am not a lawyer and I am not rendering an opinion on the legal question of whether any particular document is, or is not, a “printed publication” under the law.

8. I am, however, rendering my expert opinion on the authenticity of the documents referenced herein and on when and how each of these documents was disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, could have located the documents before 13 December 1998.

9. I am informed by counsel that an item is considered authentic if there is sufficient evidence to support a finding that the item is what it is claims to be. I am also informed that authenticity can be established based on the contents of the documents themselves, such as the appearance, contents, substance, internal patterns, or other distinctive characteristics of the item, taken together with all of the circumstances. I am further informed that an item is considered authentic if it

is at least 20 years old, in a condition that creates no suspicion of its authenticity, and in a place where, if authentic, it would likely be.

10. I am informed by counsel that a given reference is publicly accessible upon a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable diligence, can locate it. I have also been informed by counsel that materials available in a library constitute printed publications if they are cataloged and indexed (such as by subject) according to general library practices that make the references available to members of the interested public.

11. *Materials considered.* In forming the opinions expressed in this declaration, I have reviewed the documents and attachments referenced herein. These materials are records created in the ordinary course of business by publishers, libraries, indexing services, and others. From my years of experience, I am familiar with the process for creating many of these records, and I know these records are created by people with knowledge of the information in the record. Further, these records are created with the expectation that researchers and other members of the public will use them. All materials cited in this declaration and its attachments are of a type that experts in my field would reasonably rely upon and refer to in forming their opinions.

12. *Persons of ordinary skill in the art.* I am told by counsel that the subject matter of these proceedings relate to (1) the central management of license information and application programs, including user and administrator preferences, and (2) the display of a notification icon when a background application in a multitasking environment ceases executing.

13. I have been informed by counsel that a “person of ordinary skill in the art at the time of the inventions” is a hypothetical person who is presumed to be familiar with the relevant field and its literature at the time of the inventions. This hypothetical person is also a person of ordinary creativity, capable of understanding the scientific principles applicable to the pertinent field.

14. I am told by counsel that persons of ordinary skill in this subject matter or art would have had at least an undergraduate degree in computer science, computer engineering, or a related field or an equivalent number of years of working experience. In addition, a POSITA would have at least one to two years of experience in networking environments, including at least some experience with management of application programs in a network environment..

15. It is my opinion that such a person would have been engaged in academic research, learning through study and practice in the field and possibly through formal instruction the bibliographic resources relevant to his or her research. In the 1980s and 1990s such a person would have had access to a vast

array of long-established print resources in electrical/computer engineering and computer science as well as to a rich and fast changing set of online resources providing indexing information, abstracts, and full text services for electrical/computer engineering and computer science.

16. *Library catalog records.* Some background on MARC formatted records, OCLC, WorldCat, and OCLC's Connexion is needed to understand the library catalog records discussed in this declaration.

17. Libraries world-wide use the MARC format for catalog records; this machine readable format was developed at the Library of Congress in the 1960s.

18. MARC formatted records provide a variety of subject access points based on the content of the document being cataloged. All may be found in the MARC Fields 6XX. For example, MARC Field 600 identifies personal names used as subjects and the MARC Field 650 identifies topical terms. A researcher might discover material relevant to his or her topic by a search using the terms employed in the MARC Fields 6XX.

19. The MARC Field 040, subfield a, identifies the library or other entity that created the original catalog record for a given document and transcribed it into machine readable form. The MARC Field 008 identifies the date when this first catalog record was entered on the file. This date persists in all subsequent uses of the first catalog record, although newly-created records for the same document,

separate from the original record, will show a new date. It is not unusual to find multiple catalog records for the same document.

20. WorldCat is the world's largest public online catalog, maintained by the Online Computer Library Center, Inc., or OCLC, and built with the records created by the thousands of libraries that are members of OCLC. WorldCat provides a user-friendly interface for the public to use MARC records; it requires no knowledge of MARC tags and codes. WorldCat records appear in many different catalogs, including the Statewide Illinois Library Catalog. The date a given catalog record was created (corresponding to the MARC Field 008) appears in some detailed WorldCat records as the Date of Entry.

21. Whereas WorldCat records are very widely available, the availability of MARC formatted records varies from library to library.

22. When an OCLC participating institution acquires a document for which it finds no previously created record in OCLC, or when the institution chooses not to use an existing record, it creates a record for the document using OCLC's Connexion, the bibliographic system used by catalogers to create MARC records. Connexion automatically supplies the date of record creation in the MARC Field 008.

23. Once the MARC record is created by a cataloger at an OCLC participating member institution, it becomes available to other OCLC participating

members in Connexion and also in WorldCat, where persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, can locate it.

24. When a book has been cataloged, it will normally be made available to readers soon thereafter—normally within a few days or (at most) within a few weeks of cataloging.

25. *Internet Archive*. The Internet Archive is a non-profit digital library founded in 1996.

26. The Internet Archive maintains an archive of webpages collected from the Internet using software called a crawler. Crawlers automatically create a snapshot of webpages as they existed at a certain point in time. The WayBack Machine is an application using a crawler created by the Internet Archive to search its archive of Web page URLs and to represent, graphically, the date of each crawler capture.

27. The Internet Archive, now with about 50 petabytes of data, collects only Web material that is publicly available. Some sites are “not archived because they were password protected, blocked by robots.txt, or otherwise inaccessible to our automated systems. Site owners might have also requested that their sites be excluded from the WayBack Machine” (see the WayBack Machine FAQ, https://archive.org/about/faqs.php#The_Wayback_Machine).

28. Many Internet Archive captures made by the WayBack Machine have a banner at the top with the capture date prominently displayed. Other dates when captures of the same URL have been made are indicated to the right and left of the date provided in the banner. Some captures may lack this banner. In any case, the URL for the capture begins with the identification of the Internet Archive page (e.g., <http://web.archive.org/web/>) followed by information that dates and time stamps the capture as follows: year in yyyy, month in mm, day in dd, time code in hh:mm:ss (e.g., 20041208081749, or 8 December 2004 at 8:17:49 a.m.). These elements are then followed by the URL of the original capture site.

29. Internet Archive captures often include links to other, related documents. Sometimes these links have become inactive. Where they remain active, the WayBack Machine is programmed to produce the archived file with the closest available date (not the closest available prior date) to the page upon which the link appeared and was clicked.

30. The Internet Archive is a resource that is well known to library professionals and is used by many such professionals.

31. *Indexing.* A researcher may discover material relevant to his or her topic in a variety of ways. One common means of discovery is to search for relevant information in an index of periodical and other publications. Having found relevant material, the researcher will then normally obtain it online, look for

it in libraries, or purchase it from the publisher, a bookstore, a document delivery service, or other provider. Sometimes, the date of a document's public accessibility will involve both indexing and library date information. Date information for indexing entries is, however, often unavailable. This is especially true for online indices.

32. Indexing services use a wide variety of controlled vocabularies to provide subject access and other means of discovering the content of documents. The formats in which these access terms are presented vary from service to service.

33. Online indexing services commonly provide bibliographic information, abstracts, and full-text copies of the indexed publications, along with a list of the documents cited in the indexed publication. These services also often provide lists of publications that cite a given document. A citation of a document is evidence that the document was publicly available and in use by researchers no later than the publication date of the citing document.

34. Prominent indexing services include:

35. Google Scholar. Google Scholar indexes the texts and metadata of scholarly publications across a wide range of disciplines. It includes most peer-reviewed online academic journals, conference papers, theses, technical reports, and other material. Google does not publish the size of the Google Scholar database, but researchers have estimated that it contained approximately 160

million items in 2014 (Enrique Oduña-Malea, et al., “About the size of Google Scholar: playing the numbers,” Granada: EC3 Working Papers, 1B: 23 July 2014, available at <https://arxiv.org/ftp/arxiv/papers/1407/1407.6239.pdf>).

IV. OPINIONS REGARDING INDIVIDUAL DOCUMENTS

Document 1. Inside Macintosh. Volume VI. Reading, MA: Addison-Wesley Publishing Company, 1991.

1. Authentication

36. Document 1 is a book published by Addison-Wesley in 1991.

Attachment 1a is a true and accurate copy of the book’s cover, preliminary material, title page, title page verso, warranty statement, table of contents, and list of figures, tables and listings from the Northern Illinois University Library.

Attachment 1b is a true and accurate copy of that library’s catalog record for Document 1, showing the book’s location and availability.

37. Attachment 1a is in a condition that creates no suspicion about its authenticity. Specifically, the contents pages in Document 1 are not missing any intermediate pages, the text on each page appears to flow seamlessly from one page to the next, and there are no visible alterations to the document. Attachment 1a was found within the custody of a library – a place where, if authentic, it would likely be found.

38. EX1002 in this proceeding, provided by counsel, is another copy of Document 1. I have compared the relevant pages of EX1002 to Attachment 1a and

find them to be substantively identical. EX1002 and Attachment 1a both have the same ISBN number, 0-201-57755-0.

39. I conclude, based on finding Document 1 in a library and on finding library catalog records for Document 1, that Document 1 is an authentic document and that Attachment 1a and EX1002 are authentic copies of Document 1.

2. **Public accessibility**

40. Attachment 1c is a true and correct copy of a Statewide Illinois Library catalog record for Document 1, showing this book held by 52 libraries world-wide. Attachment 1c also indicates that Document 1 was cataloged or indexed in a meaningful way—including being cataloged by subject. The date of entry in Attachment 1c is 5 November 1991.

41. Attachment 1d is a true and correct copy of the University of North Carolina at Asheville Library catalog record (from the Western North Carolina Library Network), in MARC format, for Document 1. In Attachment 1d, the MARC Field 040, subfield a, indicates that this catalog record for Document 1 was created by the Allen County Public Library (OCLC code = IMF). The MARC Field 008 indicates this catalog record was created on 5 November 1991.

Allowing for some time between the cataloging of Document 1 and its arrival on library shelves, where it would be publicly available, I conclude that Document 1 was accessible to the public interested in the art, and that an ordinarily skilled

researcher, exercising reasonable diligence, would have had no difficulty finding Document 1 in at least one library by December 1991.

42. Attachment 1e is a true and correct copy of Google Scholar list of publications citing Document 1. Attachment 1f is a true and correct copy of the title page, title page verso, table of contents, and bibliography for one of these documents, Dave Mark, *Learn C++ on the Macintosh* (Reading, MA: Addison-Wesley Publishing Company, 1993). Document 1 is the 8th item in the list of references in this book's bibliography, pp. 431-432. I conclude that Document 1 was in actual use by researchers by 1993.

3. **Conclusion**

43. Based on the evidence presented here—book publication, library records, and citation—it is my opinion that Document 1 is an authentic document that was publicly available in at least one library by December 1991. I further conclude that Document 1 was in actual use by researchers by 1993.

ATTACHMENTS

44. The attachments attached hereto are true and correct copies of the materials identified above. Helen Sullivan is a Managing Partner in Prior Art Documentation Services LLC (see <http://www.priorartdocumentation.com/hellen-sullivan/>). One of her primary responsibilities in our partnership is to secure the bibliographic documentation used in attachments to our declarations.

45. Ms. Sullivan and I work in close collaboration on the bibliographic documentation needed in each declaration. I will sometimes request specific bibliographic documents or, more rarely, secure them myself. In all cases, I have carefully reviewed the bibliographic documentation used in my declaration. My signature on the declaration indicates my full confidence in the authenticity, accuracy, and reliability of the bibliographic documentation used.

46. Each Attachment has been marked with an identifying label on the top of each page. However, no alterations other than these noted labels appear in these attachments, unless otherwise noted. All attachments were created on 4-13 July 2017 and all URLs referenced in this declaration were available 9 July 2017.

V. CONCLUSION

47. In summary, I have concluded that Document 1, discussed above, is an authentic document that was publicly accessible before January 14, 1993.

48. I reserve the right to supplement my opinions in the future to respond to any arguments that Patent Owner or its expert(s) may raise and to take into account new information as it becomes available to me.

49. I declare that all statements made herein of my knowledge are true, and that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements

and the like so made are punishable by fine or imprisonment, or both, under
Section 1001 of Title 18 of the United States Code.

Executed this 21st day of July, 2017 in Urbana, Illinois.

A handwritten signature in black ink, appearing to read "Scott Bennett". The signature is written in a cursive style with a large, prominent "S" at the beginning.

Scott Bennett

Appendix A

SCOTT BENNETT
Yale University Librarian Emeritus

711 South Race
Urbana, Illinois 61801-4132
2scottbb@gmail.com
217-367-9896

EMPLOYMENT

Retired, 2001. Retirement activities include:

- Managing Partner in Prior Art Documentation Services, LLC, 2015-. This firm provides documentation services to patent attorneys; more information is available at <http://www.priorartdocumentation.com>
- Consultant on library space design, 2004-. This consulting practice is rooted in a research, publication, and public speaking program conducted since I retired from Yale University in 2001. I have served more than 50 colleges and universities in the United States and abroad with projects ranging in likely cost from under \$50,000 to over \$100 million. More information is available at <http://www.libraryspaceplanning.com/>
- Senior Advisor for the library program of the **Council of Independent Colleges**, 2001-2009
- Member of the Wartburg College Library Advisory Board, 2004-
- Visiting Professor, Graduate School of Library and Information Science, **University of Illinois at Urbana-Champaign**, Fall 2003

University Librarian, **Yale University**, 1994-2001

Director, The Milton S. Eisenhower Library, **The Johns Hopkins University**, Baltimore, Maryland, 1989-1994

Assistant University Librarian for Collection Management, **Northwestern University**, Evanston, Illinois, 1981-1989

Instructor, Assistant and Associate Professor of Library Administration, **University of Illinois at Urbana-Champaign**, 1974-1981

Assistant Professor of English, **University of Illinois at Urbana-Champaign**, 1967-1974

Woodrow Wilson Teaching Intern, **St. Paul's College**, Lawrenceville, Virginia, 1964-1965

EDUCATION

University of Illinois, M.S., 1976 (Library Science)
Indiana University, M.A., 1966; Ph.D., 1967 (English)
Oberlin College, A.B. magna cum laude, 1960 (English)

HONORS AND AWARDS

Morningside College (Sioux City, IA) Doctor of Humane Letters, 2010

American Council of Learned Societies Fellowship, 1978-1979; Honorary Visiting Research Fellow, Victorian Studies Centre, **University of Leicester**, 1979; **University of Illinois** Summer Faculty Fellowship, 1969

Indiana University Dissertation Year Fellowship and an **Oberlin College** Haskell Fellowship, 1966-1967; **Woodrow Wilson** National Fellow, 1960-1961

PROFESSIONAL ACTIVITIES

American Association for the Advancement of Science: Project on Intellectual Property and Electronic Publishing in Science, 1999-2001

American Association of University Professors: University of Illinois at Urbana-Champaign Chapter Secretary and President, 1975-1978; Illinois Conference Vice President and President, 1978-1984; national Council, 1982-1985, Committee F, 1982-1986, Assembly of State Conferences Executive Committee, 1983-1986, and Committee H, 1997-2001 ; Northwestern University Chapter Secretary/Treasurer, 1985-1986

Association of American Universities: Member of the Research Libraries Task Force on Intellectual Property Rights in an Electronic Environment, 1993-1994, 1995-1996

Association of Research Libraries: Member of the Preservation Committee, 1990-1993; member of the Information Policy Committee, 1993-1995; member of the Working Group on Copyright, 1994-2001; member of the Research Library Leadership and Management Committee, 1999-2001; member of the Board of Directors, 1998-2000

Carnegie Mellon University: Member of the University Libraries Advisory Board, 1994

Center for Research Libraries: Program Committee, 1998-2000

Johns Hopkins University Press: Ex-officio member of the Editorial Board, 1990-1994; Co-director of Project Muse, 1994

Library Administration and Management Association, Public Relations Section, Friends of the Library Committee, 1977-1978

Oberlin College: Member of the Library Visiting Committee, 1990, and of the Steering Committee for the library's capital campaign, 1992-1993; President of the Library Friends, 1992-1993, 2004-2005; member, Friends of the Library Council, 2003-

Research Society for Victorian Periodicals: Executive Board, 1971-1983; Co-chairperson of the Executive Committee on Serials Bibliography, 1976-1982; President, 1977-1982

A Selected Edition of W.D. Howells (one of several editions sponsored by the MLA Center for Editions of American Authors): Associate Textual Editor, 1965-1970; Center for Editions of American Authors panel of textual experts, 1968-1970

Victorian Studies: Editorial Assistant and Managing Editor, 1962-1964

Wartburg College: member, National Advisory Board for the Vogel Library, 2004-

Some other activities: Member of the **Illinois State Library** Statewide Library and Archival Preservation Advisory Panel; member of the **Illinois State Archives** Advisory Board; member of a committee advising the **Illinois Board of Higher Education** on the cooperative management of research collections; chair of a major collaborative research project conducted by the **Research Libraries Group** with support from Conoco, Inc.; active advisor on behalf of the **Illinois Conference AAUP** to faculty and administrators on academic freedom and tenure matters in northern Illinois.

Delegate to **Maryland Governor's Conference on Libraries and Information Service**; principal in initiating state-wide preservation planning in Maryland; principal in an effort to widen the use of mass deacidification for the preservation of library materials through cooperative action by the **Association of Research Libraries** and the **Committee on Institutional Cooperation**; co-instigator of a campus-wide information service for **Johns Hopkins University**; initiated efforts with the **Enoch Pratt Free Library** to provide information services to Baltimore's Empowerment Zones; speaker or panelist on academic publishing, copyright, scholarly communication, national and regional preservation planning, mass deacidification.

Consultant for the **University of British Columbia** (1995), **Princeton University** (1996), **Modern Language Association**, (1995, 1996), **Library of Congress** (1997), **Center for Jewish History** (1998, 2000-), **National Research Council** (1998); Board of Directors for the **Digital Library Federation**, 1996-2001; accreditation visiting team at **Brandeis University** (1997); mentor for **Northern Exposure to Leadership** (1997); instructor and mentor for ARL's **Leadership and Career Development Program** (1999-2000)

At the **Northwestern University Library**, led in the creation of a preservation department and in the renovation of the renovation, for preservation purposes, of the Deering Library book stacks.

At the **Milton S. Eisenhower Library**, led the refocusing and vitalization of client-centered services; strategic planning and organizational restructuring for the library; building renovation planning. Successfully completed a \$5 million endowment campaign for the humanities collections and launched a \$27 million capital campaign for the library.

At the **Yale University Library**, participated widely in campus-space planning, university budget planning, information technology development, and the promotion of effective teaching and learning; for the library has exercised leadership in space planning and renovation, retrospective conversion of the card catalog, preservation, organizational development, recruitment of minority librarians, intellectual property and copyright issues, scholarly communication, document delivery services among libraries, and instruction in the use of information resources. Oversaw approximately \$70 million of library space renovation and construction. Was co-principal investigator for a grant to plan a digital archive for Elsevier Science.

Numerous to invitations speak at regional, national, and other professional meetings and at alumni meetings. Lectured and presented a series of seminars on library management at the **Yunnan University Library**, 2002. Participated in the 2005 International Roundtable for Library and Information Science sponsored by the **Kanazawa Institute of Technology** Library Center and the Council on Library and Information Resources.

PUBLICATIONS

“Putting Learning into Library Planning,” *portal: Libraries and the Academy*, 15, 2 (April 2015), 215-231.

“How librarians (and others!) love silos: Three stories from the field “ available at the Learning Spaces Collaboratory Web site, <http://www.pkallsc.org/>

“Learning Behaviors and Learning Spaces,” *portal: Libraries and the Academy*, 11, 3 (July 2011), 765-789.

“Libraries and Learning: A History of Paradigm Change,” *portal: Libraries and the Academy*, 9, 2 (April 2009), 181-197. Judged as the best article published in the 2009 volume of *portal*.

“The Information or the Learning Commons: Which Will We Have?” *Journal of Academic Librarianship*, 34 (May 2008), 183-185. One of the ten most-cited articles published in JAL, 2007-2011.

“Designing for Uncertainty: Three Approaches,” *Journal of Academic Librarianship*, 33 (2007), 165–179.

“Campus Cultures Fostering Information Literacy,” *portal: Libraries and the Academy*, 7 (2007), 147-167. Included in Library Instruction Round Table Top Twenty library instruction articles published in 2007

“Designing for Uncertainty: Three Approaches,” *Journal of Academic Librarianship*, 33 (2007), 165–179.

“First Questions for Designing Higher Education Learning Spaces,” *Journal of Academic Librarianship*, 33 (2007), 14-26.

“The Choice for Learning,” *Journal of Academic Librarianship*, 32 (2006), 3-13.

With Richard A. O’Connor, “The Power of Place in Learning,” *Planning for Higher Education*, 33 (June-August 2005), 28-30

“Righting the Balance,” in *Library as Place: Rethinking Roles, Rethinking Space* (Washington, DC: Council on Library and Information Resources, 2005), pp. 10-24

Libraries Designed for Learning (Washington, DC: Council on Library and Information Resources, 2003)

“The Golden Age of Libraries,” in *Proceedings of the International Conference on Academic Librarianship in the New Millennium: Roles, Trends, and Global Collaboration*, ed. Haipeng Li (Kunming: Yunnan University Press, 2002), pp. 13-21. This is a slightly different version of the following item.

“The Golden Age of Libraries,” *Journal of Academic Librarianship*, 24 (2001), 256-258

“Second Chances. An address . . . at the annual dinner of the Friends of the Oberlin College Library November 13 1999,” Friends of the Oberlin College Library, February 2000

“Authors’ Rights,” *The Journal of Electronic Publishing* (December 1999),
<http://www.press.umich.edu/jep/05-02/bennett.html>

“Information-Based Productivity,” in *Technology and Scholarly Communication*, ed. Richard Ekman and Richard E. Quandt (Berkeley, 1999), pp. 73-94

“Just-In-Time Scholarly Monographs: or, Is There a Cavalry Bugle Call for Beleaguered Authors and Publishers?” *The Journal of Electronic Publishing* (September 1998),
<http://www.press.umich.edu/jep/04-01/bennett.html>

“Re-engineering Scholarly Communication: Thoughts Addressed to Authors,” *Scholarly Publishing*, 27 (1996), 185-196

“The Copyright Challenge: Strengthening the Public Interest in the Digital Age,” *Library Journal*, 15 November 1994, pp. 34-37

“The Management of Intellectual Property,” *Computers in Libraries*, 14 (May 1994), 18-20

“Repositioning University Presses in Scholarly Communication,” *Journal of Scholarly Publishing*, 25 (1994), 243-248. Reprinted in *The Essential JSP. Critical Insights into the World of Scholarly Publishing. Volume 1: University Presses* (Toronto: University of Toronto Press, 2011), pp. 147-153

“Preservation and the Economic Investment Model,” in *Preservation Research and Development. Round Table Proceedings, September 28-29, 1992*, ed. Carrie Beyer (Washington, D.C.: Library of Congress, 1993), pp. 17-18

“Copyright and Innovation in Electronic Publishing: A Commentary,” *Journal of Academic Librarianship*, 19 (1993), 87-91; reprinted in condensed form in *Library Issues: Briefings for Faculty and Administrators*, 14 (September 1993)

with Nina Matheson, “Scholarly Articles: Valuable Commodities for Universities,” *Chronicle of Higher Education*, 27 May 1992, pp. B1-B3

“Strategies for Increasing [Preservation] Productivity,” *Minutes of the [119th] Meeting [of the Association of Research Libraries]* (Washington, D.C., 1992), pp. 39-40

“Management Issues: The Director’s Perspective,” and “Cooperative Approaches to Mass Deacidification: Mid-Atlantic Region,” in *A Roundtable on Mass Deacidification*, ed. Peter G. Sparks (Washington, D.C.: Association of Research Libraries, 1992), pp. 15-18, 54-55

“The Boat that Must Stay Afloat: Academic Libraries in Hard Times,” *Scholarly Publishing*, 23 (1992), 131-137

“Buying Time: An Alternative for the Preservation of Library Material,” *ACLS Newsletter*, Second Series 3 (Summer, 1991), 10-11

“The Golden Stain of Time: Preserving Victorian Periodicals” in *Investigating Victorian Journalism*, ed. Laurel Brake, Alex Jones, and Lionel Madden (London: Macmillan, 1990), pp. 166-183

“Commentary on the Stephens and Haley Papers” in *Coordinating Cooperative Collection Development: A National Perspective*, an issue of *Resource Sharing and Information Networks*, 2 (1985), 199-201

“The Editorial Character and Readership of *The Penny Magazine*: An Analysis,” *Victorian Periodicals Review*, 17 (1984), 127-141

“Current Initiatives and Issues in Collection Management,” *Journal of Academic Librarianship*, 10 (1984), 257-261; reprinted in *Library Lit: The Best of 85*

“Revolutions in Thought: Serial Publication and the Mass Market for Reading” in *The Victorian Periodical Press: Samplings and Soundings*, ed. Joanne Shattock and Michael Wolff (Leicester: Leicester University Press, 1982), pp. 225-257

“Victorian Newspaper Advertising: Counting What Counts,” *Publishing History*, 8 (1980), 5-18

“Library Friends: A Theoretical History” in *Organizing the Library's Support: Donors, Volunteers, Friends*, ed. D.W. Krummel, Allerton Park Institute Number 25 (Urbana: University of Illinois Graduate School of Library Science, 1980), pp. 23-32

“The Learned Professor: being a brief account of a scholar [Harris Francis Fletcher] who asked for the Moon, and got it,” *Non Solus*, 7 (1980), 5-12

“Prolegomenon to Serials Bibliography: A Report to the [Research] Society [for Victorian Periodicals],” *Victorian Periodicals Review*, 12 (1979), 3-15

“The Bibliographic Control of Victorian Periodicals” in *Victorian Periodicals: A Guide to Research*, ed. J. Don Vann and Rosemary T. VanArsdel (New York: Modern Language Association, 1978), pp. 21-51

“John Murray's Family Library and the Cheapening of Books in Early Nineteenth Century Britain,” *Studies in Bibliography*, 29 (1976), 139-166. Reprinted in Stephen Colclough and Alexis Weedon, eds., *The History of the Book in the West: 1800-1914*, Vol. 4 (Farnham, Surrey: Ashgate, 2010), pp. 307-334.

with Robert Carringer, “Dreiser to Sandburg: Three Unpublished Letters,” *Library Chronicle*, 40 (1976), 252-256

“David Douglas and the British Publication of W. D. Howells' Works,” *Studies in Bibliography*, 25 (1972), 107-124

as primary editor, W. D. Howells, *Indian Summer* (Bloomington: Indiana University Press, 1971)

“The Profession of Authorship: Some Problems for Descriptive Bibliography” in *Research Methods in Librarianship: Historical and Bibliographic Methods in Library Research*, ed. Rolland E. Stevens (Urbana: University of Illinois Graduate School of Library Science, 1971), pp. 74-85

edited with Ronald Gottesman, *Art and Error: Modern Textual Editing* (Bloomington: Indiana University Press, 1970)--also published in London by Methuen, 1970

“Catholic Emancipation, the *Quarterly Review*, and Britain’s Constitutional Revolution,” *Victorian Studies*, 12 (1969), 283-304

as textual editor, W. D. Howells, *The Altrurian Romances* (Bloomington: Indiana University Press, 1968); introduction and annotation by Clara and Rudolf Kirk

as associate textual editor, W. D. Howells, *Their Wedding Journey* (Bloomington: Indiana University Press, 1968); introduction by John Reeves

“A Concealed Printing in W. D. Howells,” *Papers of the Bibliographic Society of America*, 61 (1967), 56-60

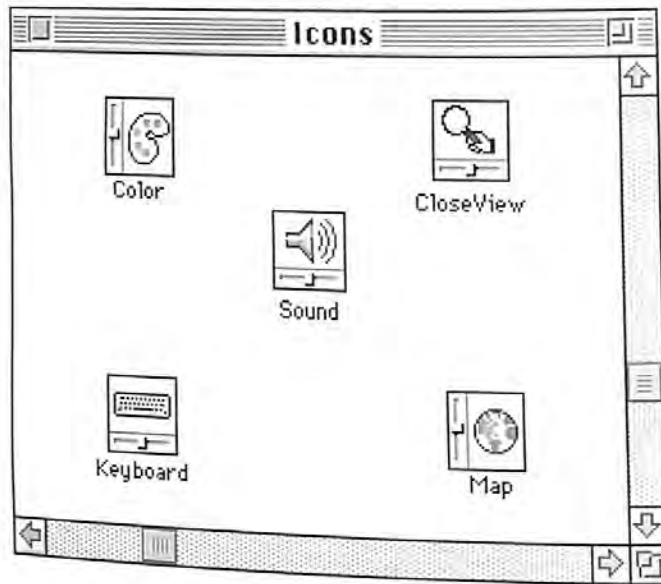
editor, *Non Solus*, A Publication of the University of Illinois Library Friends, 1974-1981

editor, Robert B. Downs Publication Fund, University of Illinois Library, 1975-1981

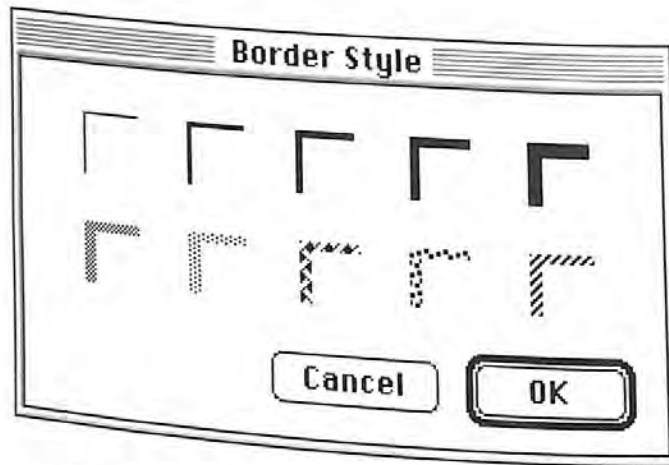
Reviews, short articles, etc. in *Victorian Studies*, *Journal of English and German Philology*, *Victorian Periodicals Newsletter*, *Collection Management*, *Nineteenth-Century Literature*, *College & Research Libraries*, *Scholarly Publishing Today*, *ARL Newsletter*, *Serials Review*, *Library Issues*, *S[ociety for] S[cholarly] P[ublishing] Newsletter*, and *Victorian Britain: An Encyclopedia*

Icon	Name
	Arabic
	Canada
	Cyrillic
	Cyrillic transliterated
	Denmark
	Faeroe Islands
	Germany
	Hebrew
	Japanese Katakana
	Japanese Romaji
	Korean
	Netherlands, period decimal separator (previously )
	Netherlands, comma decimal separator
	Roman (U.S.)
	Spain
	Swiss French
	Swiss German
	Swiss Italian
	Turkey
	Turkish, U.S. modified
	United Kingdom (previously )
	United States

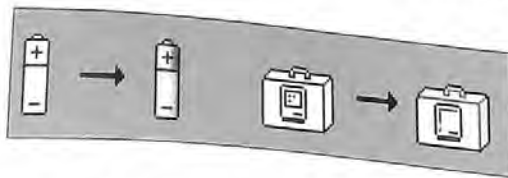
Color Plate I. Examples of keyboard icons



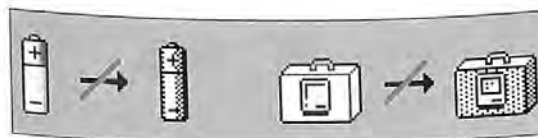
Color Plate II. A colorized window



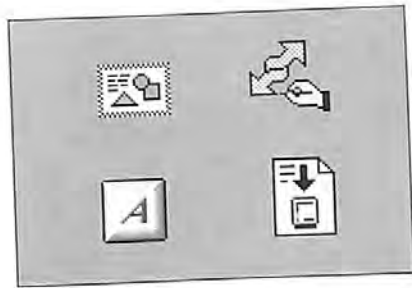
Color Plate III. A colorized movable modal dialog box



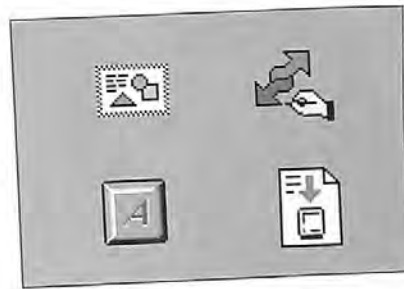
Color Plate IV. Design for black-and-white monitors first



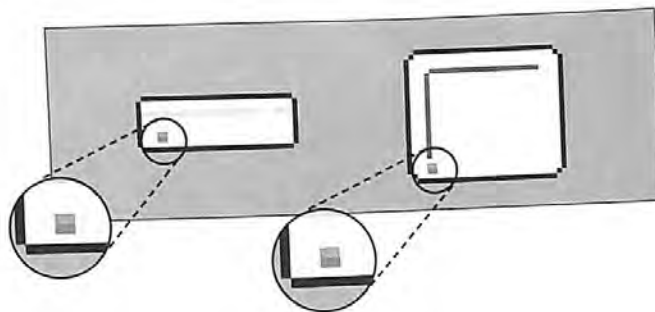
Color Plate V. Don't mimic color effects in black-and-white designs



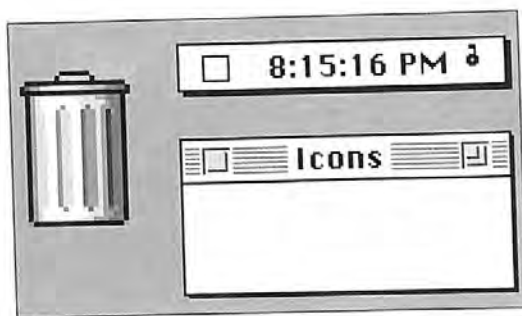
Color Plate VI. Use light colors for large areas for large areas



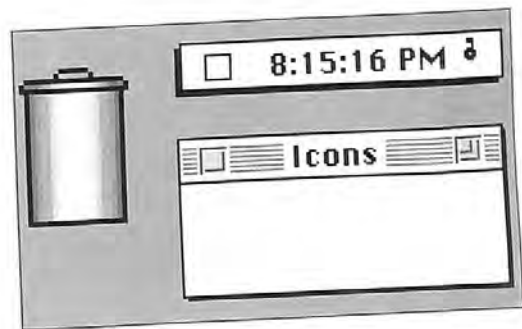
Color Plate VII. Don't use bright colors for large areas



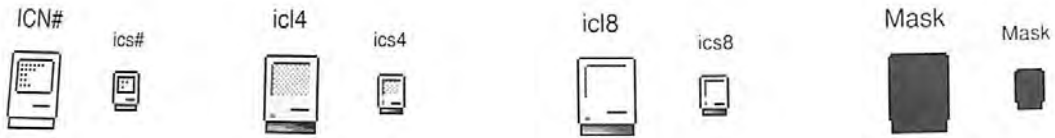
Color Plate VIII. Use bright colors for details (enlarged for detail)



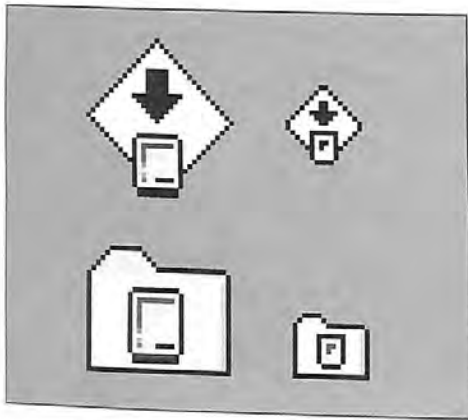
Color Plate IX. A consistent light source (enlarged for detail)



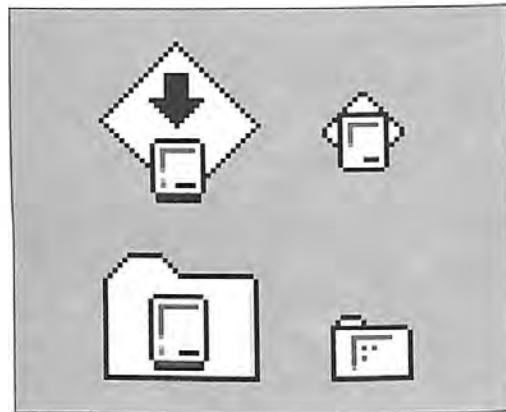
Color Plate X. Inconsistent light sources (enlarged for detail)



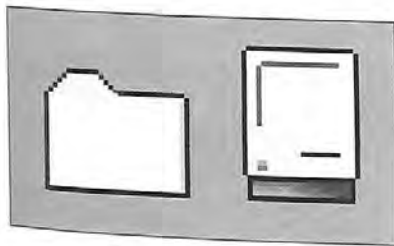
Color Plate XI. An icon family



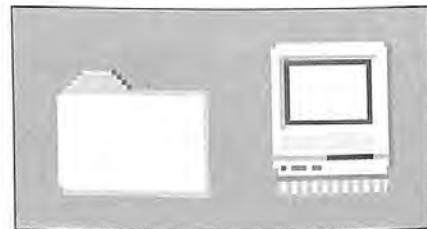
Color Plate XII. Consistently designed small icons (enlarged for detail)



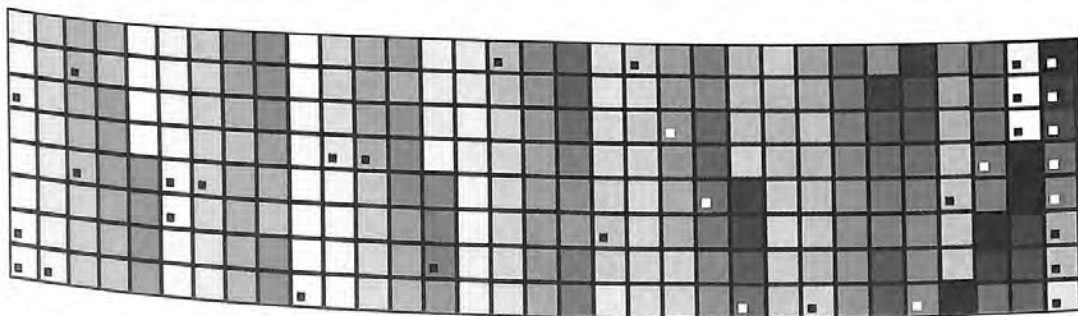
Color Plate XIII. Inconsistently designed small icons (enlarged for detail)



Color Plate XIV. Icons with a black outline (enlarged for detail)



Color Plate XV. Icons without a black outline (enlarged for detail)



Color Plate XVI. Apple icon colors (as marked)

Before anti-aliasing



Correctly anti-aliased

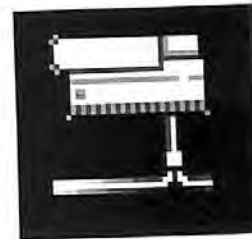
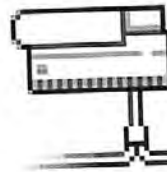


Color Plate XVII. Correct anti-aliasing (enlarged for detail)

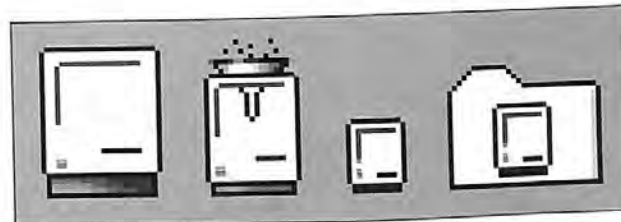
Before anti-aliasing



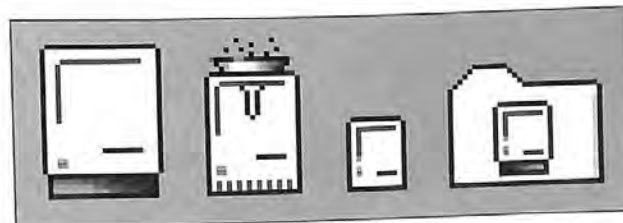
Incorrectly anti-aliased



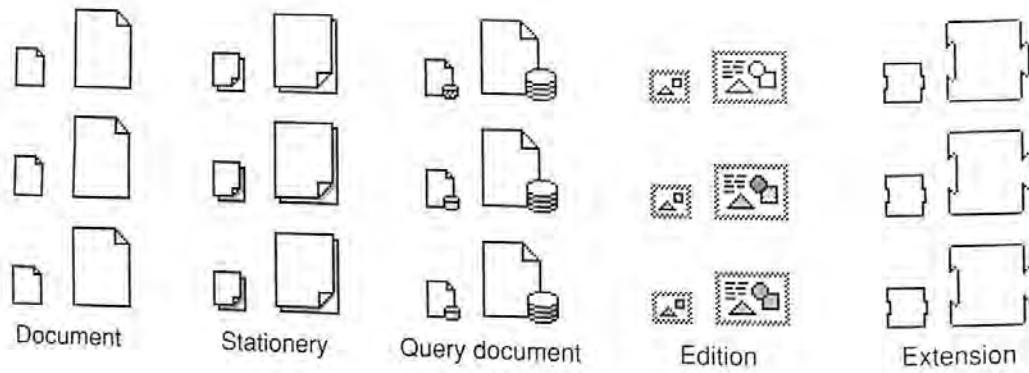
Color Plate XVIII. Incorrect anti-aliasing (enlarged for detail)



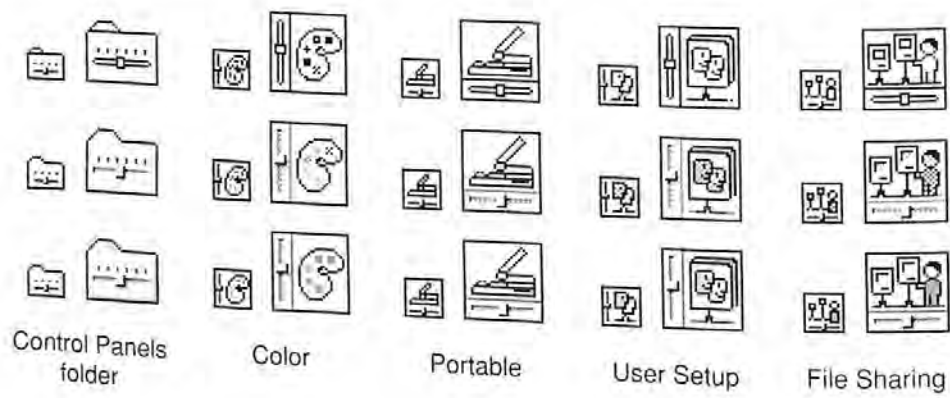
Color Plate XIX. Consistent use of icon elements (enlarged for detail)



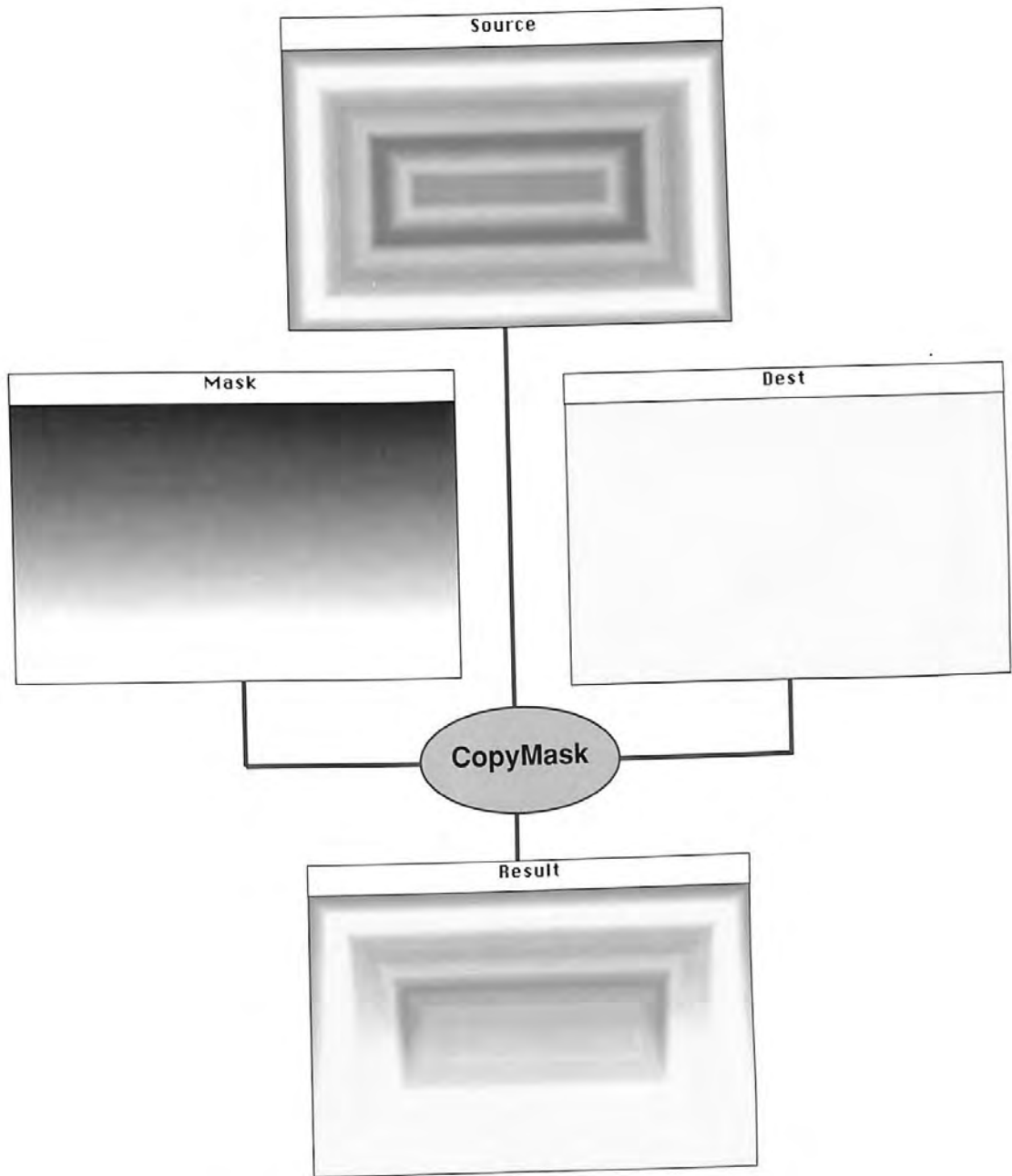
Color Plate XX. Inconsistent use of icon elements (enlarged for detail)



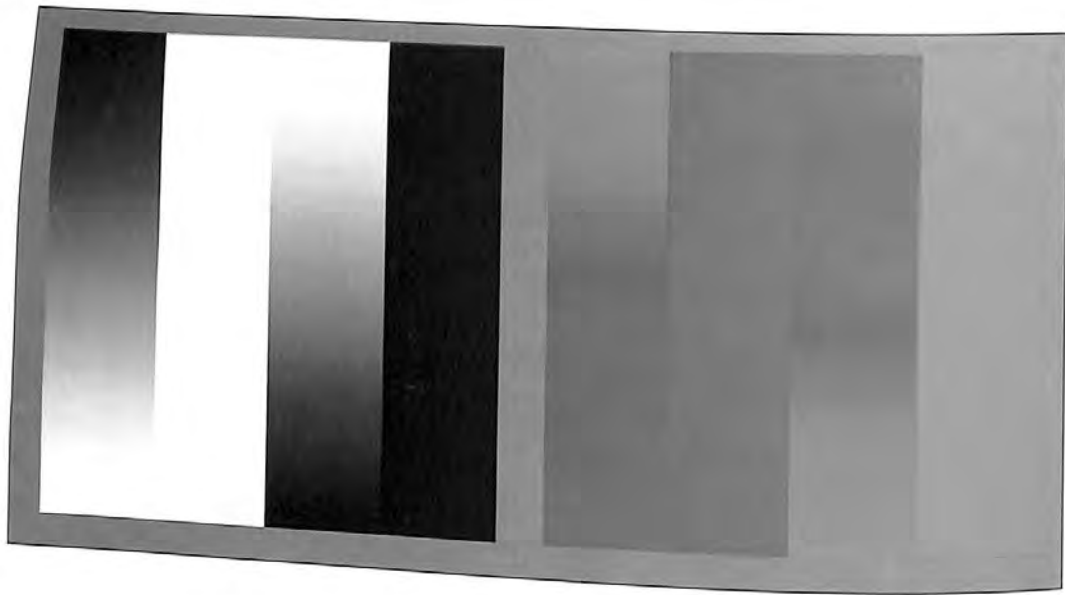
Color Plate XXI. Default system icon families



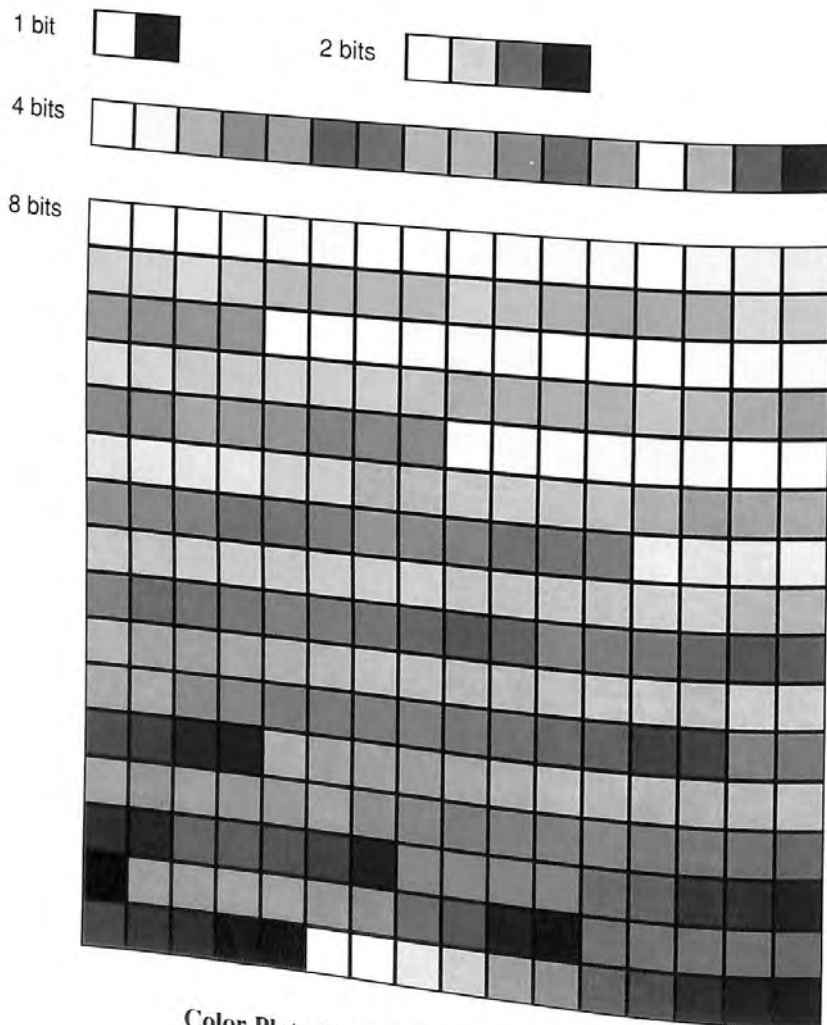
Color Plate XXII. Examples of control panel icons



Color Plate XXIII. Copying with a pixel map as a mask.



Color Plate XXIV. Colorizing



Color Plate XXV. Default color tables

Inside Macintosh® Volume VI



Addison-Wesley Publishing Company, Inc.

Reading, Massachusetts Menlo Park, California New York
Don Mills, Ontario Wokingham, England Amsterdam
Bonn Sydney Singapore Tokyo Madrid
Paris Seoul Milan Mexico City Taipei San Juan

🍏 APPLE COMPUTER, INC.

This manual is copyrighted by Apple or by Apple's suppliers, with all rights reserved. Under the copyright laws, this manual may not be copied, in whole or in part, without the written consent of Apple Computer, Inc. This exception does not allow copies to be made for others, whether or not sold, but all of the material purchased may be sold, given, or lent to another person. Under the law, copying includes translating into another language.

The Apple logo is a registered trademark of Apple Computer, Inc. Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

© Apple Computer, Inc., 1991
20525 Mariani Avenue
Cupertino, CA 95014-6299
(408) 996-1010

Apple, the Apple logo, APDA, AppleLink, AppleShare, AppleTalk, Apple IIGS, A/UX, EtherTalk, HyperCard, HyperTalk, ImageWriter, LaserWriter, LocalTalk, Macintosh, MPW, MultiFinder, SANE, and TokenTalk are registered trademarks of Apple Computer, Inc.

Apple Desktop Bus, Balloon Help, Finder, KanjiTalk, Moof, QuickDraw, ResEdit, TrueType, and Zhong-Wen Talk are trademarks of Apple Computer, Inc.

Helvetica and Times are registered trademarks of Linotype Company.

ITC Zapf Dingbats is a registered trademark of International Typeface Corporation.

MacPaint is a registered trademark of Claris Corporation.

NuBus is a trademark of Texas Instruments.

PostScript is a registered trademark, and Illustrator is a trademark, of Adobe Systems Incorporated.

Sony is a registered trademark of Sony Corporation.

UNIX is a registered trademark of UNIX System Laboratories, Inc.

Simultaneously published in the United States and Canada.

ISBN 0-201-57755-0 (book)
ISBN 0-201-57776-3 (boxed edition)
1 2 3 4 5 6 7 8 9-MU-9594939291
First printing April, 1991

Inside Macintosh
Volume VI

LIMITED WARRANTY ON MEDIA AND REPLACEMENT

ALL IMPLIED WARRANTIES ON THIS MANUAL, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO NINETY (90) DAYS FROM THE DATE OF THE ORIGINAL RETAIL PURCHASE OF THIS PRODUCT.

Even though Apple has reviewed this manual, **APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS MANUAL IS SOLD "AS IS," AND YOU, THE PURCHASER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS MANUAL, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

xxix **Figures, Tables, and Listings**

Preface

- P-3 About *Inside Macintosh*
- P-8 An Overview of the Chapters in Volume VI
- P-16 A Road Map to Volume VI

1 Introduction to the System Software Version 7.0 Environment

- 1-3 About This Chapter
- 1-4 About the System Software Version 7.0 Environment
- 1-6 The Cooperative Multitasking Environment
- 1-7 Interapplication Communication
- 1-10 Sharing Data Among Applications
- 1-12 Sending Events Between Applications
- 1-13 Exchanging Message Blocks Between Programs
- 1-14 Remote Data Access
- 1-15 Enhanced User Interface
- 1-16 Sound
- 1-16 TrueType Fonts
- 1-16 Graphics
- 1-18 File Management
- 1-19 Memory Management
- 1-19 Temporary Memory
- 1-20 24-Bit and 32-Bit Addressing
- 1-20 Process Management
- 1-21 Timing Services
- 1-21 Compatibility
- 1-21 Worldwide Development
- 1-23 Communication Over a Network
- 1-23 Hardware Interfaces
- 1-24 Overview of Chapters in This Volume

2 User Interface Guidelines

- 2-3 About This Chapter
- 2-3 User Interface Design Principles
- 2-4 Worldwide Software Development
- 2-5 Cultural Values
- 2-5 Resources
- 2-6 Language Differences
- 2-6 Text Display and Text Editing
- 2-8 Default Alignment of Interface Elements
- 2-8 Keyboards
- 2-11 Fonts
- 2-12 User Documentation
- 2-13 Terminology
- 2-14 The Version 7.0 Environment
- 2-14 User Feedback
- 2-15 Background Notification

2-16	Color Design for Version 7.0
2-16	General Color Design Guidelines
2-18	The Icon Family
2-18	Black-and-White Icons
2-19	Small Icons
2-19	Color Icons
2-20	Consistent Use of Icons
2-21	Customized Icons
2-22	Windows
2-22	Window Positions
2-23	The Zoom Box and Window Behavior
2-23	Dialog Boxes
2-24	Modal Dialog Box Behaviors
2-24	Movable Modal Dialog Boxes
2-25	Keyboard Navigation in Dialog Boxes
2-26	Button Labels
2-29	Dialog Box Layout
2-30	Dialog Box Messages
2-30	Standard File Dialog Boxes
2-31	Save Changes Dialog Box
2-32	Menus
2-32	File Menu
2-32	Edit Menu
2-33	Font Menu
2-34	Help Menu
2-34	Keyboard Equivalents
2-36	Pop-Up Menus
2-36	Standard Pop-Up Menus
2-36	Type-In Pop-Up Menus
2-37	More User Interface Information

3 Compatibility Guidelines

3-3	About This Chapter
3-5	About Compatibility
3-6	Using Memory Wisely
3-6	Using Assembly Language
3-7	Accessing Hardware
3-7	Using Low-Memory Global Variables
3-7	Determining Whether a Trap Is Available
3-9	Running in System Software Version 7.0
3-10	Allowing Multiple Applications
3-11	Supporting Required Apple Events
3-12	Removing Font Size Restrictions
3-13	Operating With Virtual Memory
3-13	Enabling Menus During a Modal Dialog
3-15	Coexisting With the System Menus
3-15	Creating Movable Modal Dialog Boxes
3-16	Creating Pop-Up Menus
3-19	Manipulating Dialog Item Lists
3-19	Counting Items in a Dialog Item List
3-20	Appending Items to a Dialog Item List
3-22	Shortening a Dialog Item List

- 3-22 Localizing Macintosh Programs
 - 3-23 General Guidelines
 - 3-23 Localizing With the Toolbox
- 3-24 Running Macintosh Programs Under A/UX
 - 3-25 How the A/UX Toolbox Works
 - 3-26 Using the A/UX Toolbox
- 3-27 A/UX Compatibility Guidelines
- 3-28 About the Gestalt Manager
- 3-30 Using the Gestalt Manager
 - 3-30 Determining Features of the Operating Environment
 - 3-32 Determining Whether Gestalt Is Available
 - 3-33 Interpreting Gestalt Responses
 - 3-35 Interpreting Responses to Environmental Selectors
 - 3-41 Interpreting Responses to Informational Selectors
 - 3-42 Adding Gestalt Selectors
 - 3-44 Modifying Gestalt Selectors
 - 3-45 Specifying Gestalt Selector Functions
- 3-45 Gestalt Manager Routines
 - 3-46 Getting Information About the Operating Environment
 - 3-46 Adding Selector Codes
 - 3-47 Modifying Selector Codes
- 3-48 Summary of the Gestalt Manager
- 3-54 Summary of the Window Manager
- 3-54 Summary of the Control Manager
- 3-55 Summary of the Dialog Manager

4 The Edition Manager

- 4-3 About This Chapter
- 4-3 About the Edition Manager
- 4-4 Publishers, Subscribers, and Editions
- 4-11 Using the Edition Manager
 - 4-12 Receiving Apple Events From the Edition Manager
 - 4-14 Creating the Section Record and Alias Record
 - 4-17 Saving a Document Containing Sections
 - 4-19 Opening and Closing a Document Containing Sections
 - 4-21 Reading and Writing a Section
 - 4-22 Formats in an Edition
 - 4-23 Opening an Edition
 - 4-24 Format Marks
 - 4-24 Reading and Writing Edition Data
 - 4-25 Closing an Edition
 - 4-25 Creating a Publisher
 - 4-28 Creating the Edition Container
 - 4-30 Opening an Edition Container to Write Data
 - 4-31 Creating a Subscriber
 - 4-35 Opening an Edition Container to Read Data
 - 4-35 Choosing Which Edition Format to Read
 - 4-37 Using Publisher and Subscriber Options
 - 4-40 Publishing a New Edition While Saving or Manually
 - 4-41 Subscribing to an Edition Automatically or Manually
 - 4-41 Canceling Sections Within Documents
 - 4-42 Locating a Publisher Through a Subscriber

4-43	Renaming a Document Containing Sections
4-43	Displaying Publisher and Subscriber Borders
4-47	Text Borders
4-48	Spreadsheet Borders
4-49	Object-Oriented Graphics Borders
4-50	Bitmapped Graphics Borders
4-50	Duplicating Publishers and Subscribers
4-52	Modifying a Subscriber
4-52	Relocating an Edition
4-52	Customizing Dialog Boxes
4-54	Subscribing to Non-Edition Files
4-55	Getting the Current Edition Opener
4-56	Setting an Edition Opener
4-56	Calling an Edition Opener Procedure
4-59	Opening and Closing Editions
4-59	Listing Files That Can Be Subscribed To
4-59	Reading From and Writing to Files
4-60	Calling a Format I/O Procedure
4-61	Edition Manager Routines
4-62	Initializing the Edition Manager
4-62	Creating and Registering a Section
4-65	Creating and Deleting an Edition Container
4-67	Setting and Locating a Format Mark
4-67	Reading in Edition Data
4-69	Writing out Edition Data
4-70	Closing an Edition After Reading or Writing
4-71	Displaying Dialog Boxes
4-76	Locating a Publisher and Edition From a Subscriber
4-78	Edition Container Formats
4-78	Reading and Writing Non-Edition Files
4-81	Summary of the Edition Manager

5 The Event Manager

5-3	About This Chapter
5-4	About the Event Manager
5-5	Introduction to Events
5-5	Low-Level Events
5-6	Operating-System Events
5-8	High-Level Events
5-10	Event Processing
5-10	The Event Loop
5-12	Event Masks
5-12	Switching Contexts
5-13	Specifying Memory Requirements and Scheduling Options
5-14	The Structure of a 'SIZE' Resource
5-17	Creating a 'SIZE' Resource
5-18	Using the Event Manager
5-18	Receiving Low-Level Events
5-19	Responding to Operating-System Events
5-21	Receiving High-Level Events
5-22	Identifying High-Level Event Senders and Receivers
5-23	Sending High-Level Events

- 5-26 Requesting Return Receipts
- 5-27 Responding to Events From Other Applications
- 5-28 Searching for a Specific High-Level Event
- 5-29 Event Manager Routines
- 5-29 Receiving Events
- 5-30 Sending Events
- 5-31 Receiving a Specific High-Level Event
- 5-32 Converting Process Serial Numbers and Port Names
- 5-33 Summary of the Event Manager

6 The Apple Event Manager

- 6-3 About This Chapter
- 6-4 About the Apple Event Manager
- 6-5 Introduction to Apple Events
- 6-6 Types of Apple Events
- 6-6 Components of Apple Events
- 6-8 Data Structures Within Apple Events
- 6-14 Responding to Apple Events
- 6-19 Requesting Services Through Apple Events
- 6-25 Using the Apple Event Manager
- 6-26 Accepting an Apple Event
- 6-28 Installing Entries Into the Apple Event Dispatch Tables
- 6-30 Handling the Required Apple Events
- 6-30 Required Apple Events
- 6-33 Handling the Open Application Event
- 6-33 Handling the Open Documents Event
- 6-35 Handling the Print Documents Event
- 6-36 Handling the Quit Application Event
- 6-37 Handling Apple Events Sent by the Edition Manager
- 6-38 Handling the Create Publisher Event
- 6-40 Getting Data out of an Apple Event
- 6-42 Getting Data out of a Parameter
- 6-44 Getting Data out of an Attribute
- 6-45 Getting Data out of a Descriptor List
- 6-47 Writing Apple Event Handlers
- 6-48 Replying to an Apple Event
- 6-50 Disposing of Apple Event Data Structures
- 6-51 Interacting With the User
- 6-54 Creating an Apple Event
- 6-55 Adding Parameters to an Apple Event
- 6-57 Specifying a Target Address
- 6-59 Sending an Apple Event
- 6-63 Dealing With Timeouts
- 6-64 Writing an Idle Function
- 6-66 Writing a Reply Filter Function
- 6-67 Writing and Installing Coercion Handlers
- 6-70 The Application Died Event
- 6-71 Apple Event Manager Routines
- 6-71 Creating and Managing the Apple Event Dispatch Tables
- 6-74 Dispatching Apple Events
- 6-75 Getting Parameters and Attributes From Apple Events
- 6-77 Counting the Items in Descriptor Lists

6-78	Getting Items From Descriptor Lists
6-80	Getting Data and Keyword-Specified Descriptor Records From AE Records
6-82	Requesting User Interaction
6-83	Requesting More Time to Respond to Apple Events
6-84	Suspending and Resuming Apple Event Handling
6-86	Creating Apple Events
6-87	Creating and Duplicating Descriptor Records
6-88	Creating Descriptor Lists and AE Records
6-89	Adding Items to Descriptor Lists
6-90	Adding Data and Keyword-Specified Descriptor Records to AE Records
6-91	Adding Parameters and Attributes to Apple Events
6-93	Sending Apple Events
6-96	Getting the Sizes and Descriptor Types of Descriptor Records
6-98	Deleting Descriptor Records
6-99	Deallocating Memory for Descriptor Records
6-100	Coercing Descriptor Types
6-101	Creating and Managing the Coercion Handler Tables
6-103	Creating and Managing the Special Handler Tables
6-105	Summary of the Apple Event Manager

7 The Program-to-Program Communications Toolbox

7-3	About This Chapter
7-4	About the PPC Toolbox
7-4	Ports, Sessions, and Message Blocks
7-6	Setting Up Authenticated Sessions
7-9	Using the PPC Toolbox
7-12	PPC Toolbox Calling Conventions
7-14	Specifying Port Names and Location Names
7-16	Opening a Port
7-18	Browsing for Ports Using the Program Linking Dialog Box
7-22	Obtaining a List of Available Ports
7-23	Preparing for a Session
7-24	Initiating a PPC Session
7-28	Receiving Session Requests
7-30	Accepting or Rejecting Session Requests
7-32	Exchanging Data During a PPC Session
7-32	Reading Data From an Application
7-33	Sending Data to an Application
7-34	Ending a Session and Closing a Port
7-35	Invalidating Users
7-36	PPC Toolbox Routines
7-37	The PPC Toolbox Parameter Block and Completion Routine
7-39	Initializing the PPC Toolbox
7-39	Using the Program Linking Dialog Box
7-41	Obtaining a List of Ports
7-42	Opening and Closing a Port
7-45	Starting and Ending a Session
7-49	Receiving, Accepting, and Rejecting a Session
7-53	Reading and Writing Data
7-56	Locating a Default User and Invalidating a User
7-57	Summary of the PPC Toolbox

8 The Data Access Manager

- 8-3 About This Chapter
- 8-4 About the Data Access Manager
- 8-5 The High-Level Interface
 - 8-6 Sending a Query Through the High-Level Interface
 - 8-7 Retrieving Data Through the High-Level Interface
- 8-8 The Low-Level Interface
 - 8-9 Sending a Query Through the Low-Level Interface
 - 8-9 Retrieving Data Through the Low-Level Interface
- 8-9 Comparison of the High-Level and Low-Level Interfaces
- 8-10 Using the Data Access Manager
 - 8-10 Executing Routines Asynchronously
 - 8-11 General Guidelines for the User Interface
 - 8-11 Keep the User in Control
 - 8-11 Provide Feedback to the User
 - 8-12 Using the High-Level Interface
 - 8-19 Writing a Status Routine for High-Level Functions
 - 8-26 Using the Low-Level Interface
 - 8-33 Getting Information About Sessions in Progress
 - 8-34 Processing Query Results
 - 8-34 Getting Query Results
 - 8-38 Converting Query Results to Text
 - 8-41 Creating a Query Document
 - 8-42 User Interface Guidelines for Query Documents
 - 8-42 Contents of a Query Document
 - 8-44 Query Records and Query Resources
 - 8-44 Query Records
 - 8-46 Query Resources
 - 8-47 Writing a Query Definition Function
 - 8-50 Data Access Manager Routines
 - 8-50 Asynchronous Execution of Routines
 - 8-51 Initializing the Data Access Manager
 - 8-52 High-Level Interface
 - 8-52 Handling Query Documents
 - 8-54 Handling Query Results
 - 8-56 Low-Level Interface
 - 8-56 Controlling the Session
 - 8-60 Sending and Executing Queries
 - 8-63 Retrieving Results
 - 8-65 Installing and Removing Result Handlers
 - 8-67 Summary of the Data Access Manager

9 The Finder Interface

- 9-3 About This Chapter
- 9-3 About the Finder Interface
- 9-5 Finder-Related Resources
 - 9-6 Creators, File Types, and the Signature Resource
 - 9-9 Icon Resources
 - 9-14 File Reference Resources
 - 9-16 The Bundle Resource
 - 9-20 The Size Resource

9-20	Messages When the Finder Can't Find Your Application
9-23	Version Resources
9-24	How and When the Finder Launches Your Application
9-26	Finder-Related Changes to the User Interface
9-26	Stationery Pads
9-27	Edition Icons
9-28	Customized Icons
9-29	Aliases
9-29	Resolving Alias Files
9-32	Desk Accessories
9-33	Fonts, Sounds, and Other Movable Resources
9-34	Balloon Help for Icons
9-36	Finder Information in the Volume Catalog
9-39	The System Folder and Its Related Directories
9-40	Folder Organization
9-42	Finding Directories
9-44	The Desktop Database
9-45	History of the Desktop Database
9-46	Using the Desktop Database
9-48	Desktop Manager Routines
9-49	Locating and Opening the Desktop Database
9-51	Reading the Desktop Database
9-54	Adding to the Desktop Database
9-57	Deleting Entries From the Desktop Database
9-58	Manipulating the Desktop Database Itself
9-61	Summary of the Finder Interface

10 Control Panels

10-3	About This Chapter
10-3	About Control Panels
10-4	Writing Control Panel Files
10-5	About the Monitors Control Panel
10-6	Designing an Extension for the Monitors Control Panel
10-7	The 'card' Resource
10-8	The 'mnr' Resource
10-8	The Monitor Function
10-10	Messages to the Monitor Function
10-12	The 'RECT' Resource
10-12	The 'DITL' Resource
10-13	The 'ICON' and 'cicn' Resources
10-13	The 'vers' Resources
10-13	The 'STR#' Resource
10-13	The 'gama' Resources
10-14	The 'FREF', 'BNDL', Icon Family, and Signature Resources
10-14	The 'INIT' Resource
10-14	A Sample of an Extension to the Monitors Control Panel
10-28	Including Another Control Panel Definition in a Monitors Extension File
10-29	Summary of the Extension File for the Monitors Control Panel

11 The Help Manager

- 11-3 About This Chapter
- 11-4 About the Help Manager
- 11-5 Help Balloon Display
- 11-10 Default Help Balloons for Menus, Windows, and Icons
- 11-14 Using the Help Manager
 - 11-14 Providing Text or Pictures for Help Balloons
 - 11-15 Defining the Help Balloon Content
 - 11-16 Using Clear, Concise Phrases
 - 11-16 Using Active Constructions
 - 11-17 Using Parallel Structure
 - 11-17 Using Consistent Terminology
 - 11-17 Defining the Help Balloon Position
 - 11-17 Specifying the Format for Help Balloon Content
 - 11-18 Specifying Options in Help Resources
 - 11-21 Providing Help Balloons for Menus
 - 11-33 Providing Help Balloons for Items in Dialog Boxes and Alert Boxes
 - 11-41 Providing Help Balloons for Window Content
 - 11-42 Help Balloons in Static Windows
 - 11-49 Help Balloons in Dynamic Windows
 - 11-56 Overriding Help Balloons for Application Icons
 - 11-58 Overriding Other Default Help Balloons
 - 11-61 Adding Your Own Menu Items to the Help Menu
 - 11-64 Writing Your Own Balloon Definition Function
- 11-65 Help Manager Routines
 - 11-65 Determining Whether Help Is Enabled
 - 11-66 Determining Whether a Help Balloon Is Showing
 - 11-66 Displaying and Removing Help Balloons
 - 11-67 Displaying a Help Balloon
 - 11-69 Using Your Own Menu Definition Procedure
 - 11-71 Removing a Help Balloon
 - 11-72 Using Your Own Tip Function
 - 11-73 Adding Items to the Help Menu
 - 11-74 Getting and Setting the Font Name and Size
 - 11-75 Getting and Setting Information for Help Resources
 - 11-78 Getting the Dimensions of a Help Balloon
 - 11-79 Getting the Content of a Help Balloon
- 11-83 Summary of the Help Manager

12 The Font Manager

- 12-3 About This Chapter
- 12-3 About the Font Manager
 - 12-3 Font Terminology
 - 12-6 Font Measurements
 - 12-8 Font Scaling
- 12-10 How the Font Manager Renders TrueType Fonts
- 12-14 Using the Font Manager
 - 12-15 Adding Font Sizes and Names to the Menu
 - 12-16 Storing a Font Name in a Document
 - 12-16 Using TrueType Fonts in Preference to Bitmapped Fonts
 - 12-17 Preserving the Glyph's Shape

- 12-18 Font Manager Routines
- 12-18 Choosing TrueType Fonts Over Bitmapped Fonts
- 12-19 Scaling Fonts
- 12-21 Erasing the Font Manager's Memory Caches
- 12-22 Summary of the Font Manager

13 The Resource Manager

- 13-3 About This Chapter
- 13-3 Resources
- 13-3 Resource Types
- 13-6 Resource IDs
- 13-7 Definition Procedures
- 13-7 Font Families and Scripts
- 13-10 Resources in the System File
- 13-10 System Icons
- 13-11 Document and Application Icons
- 13-12 Folder Icons
- 13-12 System Folder Icons
- 13-13 Desktop Icons
- 13-14 Standard File Package Icons
- 13-15 User Information Resources
- 13-15 Packages
- 13-16 Function Key Resources
- 13-16 Using the Resource Manager
- 13-16 Using Partial Resources
- 13-18 Creating and Opening Resource Files
- 13-18 Storing Fonts in a Resource Fork
- 13-18 Resource Manager Routines
- 13-19 Creating Resource Files
- 13-20 Opening Resource Files
- 13-21 Reading and Writing Partial Resources
- 13-24 Summary of the Resource Manager

14 Worldwide Software Overview

- 14-5 About This Chapter
- 14-6 About Worldwide Software
- 14-7 The Graphic Representation of Languages
- 14-7 Localized Versions of the Macintosh System Software
- 14-8 Multiple Script Systems and Multiple Languages on the Macintosh Computer
- 14-8 Identifying Scripts, Languages, and Regions
- 14-10 About the Script Management System
- 14-10 About the Script Manager
- 14-12 Local and Global Variables
- 14-12 Style Runs and Higher-Level Text Organization
- 14-12 Tokens
- 14-13 Date Conversion
- 14-13 Geographic Information
- 14-13 Number Conversion
- 14-13 About the International Utilities Package
- 14-14 About the International and Keyboard Resources
- 14-15 About the Macintosh Script Systems

14-16	The Script Management System and Related Worldwide Components
14-19	Worldwide Control Panels and Desk Accessory
14-19	Installing and Removing Script Systems, Keyboards, and Fonts
14-20	Using the Keyboard Menu
14-21	Selecting Keyboard Layouts
14-21	Distinguishing Scripts
14-23	Keyboards
14-25	Fonts
14-25	Localization
14-26	Sorting
14-26	Primary or Secondary Order
14-27	Expansion
14-27	Contraction
14-27	Ignorable Characters
14-27	Exceptional Words
14-28	Formats
14-28	Date and Time
14-28	Currency and Measurement
14-28	Calendars
14-29	Numbers
14-29	An Introduction to Scripts
14-30	Character Representation
14-30	Text Direction
14-31	Contextual Forms
14-32	Diacritical Marks
14-32	Uppercase and Lowercase Characters
14-33	Character Reordering
14-33	Word Demarcation
14-33	Alignment and Justification of Text
14-34	Representing Scripts on the Macintosh
14-34	Character Set Encoding
14-35	Character Input
14-35	Composition Rules
14-35	Text Manipulation
14-36	Text Rendering
14-36	Using the Script Manager
14-36	Determining the Features of the Script Manager
14-37	Initializing the Script Manager
14-37	Creating Simple Script Systems
14-37	Calling the Script Manager
14-39	Overview of the Script Manager Routines
14-39	Checking and Modifying Global and Local Variables
14-40	Accepting Implicit Script Codes
14-40	Verbs for GetScript and SetScript
14-41	Verbs for GetEnviron and SetEnviron
14-43	Checking and Setting System Variables
14-43	Setting the Active Keyboard Script
14-44	Obtaining Script Information
14-45	Obtaining Character Information
14-45	Manipulating Text
14-45	Drawing and Editing Text
14-46	Formatting Text

14-47	Modifying Text
14-47	Substituting Text
14-47	Truncating Text
14-48	Lexically Interpreting Different Scripts
14-48	Date and Time Utilities
14-48	Converting Worldwide Dates and Times
14-48	Converting Long Dates
14-48	Modifying and Verifying Date and Time Records
14-49	Reading and Storing Locations
14-49	Number Utilities
14-49	Converting to and From Canonical Number Formats
14-49	Working With Formatted Numbers
14-50	Script Manager Routines
14-50	Localizing Word Selection and Line Break Tables
14-51	Defining Word Boundaries and Line Breaks
14-55	Determining Word Selection: An Example
14-58	Optimized Word Break Tables
14-59	Truncating Text
14-60	Substituting Text
14-61	Substituting and Truncating Text
14-62	Converting Case and Stripping Diacritical Marks
14-64	Handling Justified Text
14-65	Providing for Spacing Between Multiple Style Runs
14-68	Justifying Text on the Roman Script System
14-71	Using the International Utilities Package Routines
14-72	Overview of the International Utilities Package Routines
14-72	Comparing Strings
14-73	Modifying the Standard String Comparison
14-73	Facilitating Interscript Sorting Order
14-74	Accessing the International Resources
14-74	Localizing Dates, Times, and Metric Information
14-75	International Utilities Package Routines
14-75	Script and Language Codes
14-76	Manipulating the 'itl2' and 'itl4' Resources
14-77	Specifying Resource Handles Explicitly
14-78	Determining Interscript Sorting Order
14-79	Using the International and Keyboard Resources
14-80	Enhancements to International Resources
14-81	The 'itle' Resource
14-82	The 'itlm' Resource
14-85	The 'itlb' Resource
14-86	The 'itl0' Resource
14-87	The 'itl1' Resource
14-90	The 'itl2' Resource
14-91	The 'itl2' Resource Header
14-92	The 'itl4' Resource
14-95	Keyboard Types and Modifier Bits
14-96	The 'KCHR' Resource
14-98	The 'kcs#', 'kcs4', and 'kcs8' Resources
14-98	The 'KSWP' Resource
14-99	The 'itlk' Resource
14-100	Key Caps and the 'KCAP' Resource
14-100	Dead-Key Feedback
14-101	The 'KCAP' Resource

- 14-102 Localizing to Other Languages and Regions
 - 14-102 Using Resources
 - 14-102 Text and Dialog Translation Tips
- 14-103 Adapting Text Operations
- 14-104 Using Fonts
 - 14-104 Avoiding Special Character Codes as Delimiters
 - 14-104 Using the Standard Roman Character Set
- 14-105 Adapting Keyboard Equivalents
- 14-107 Modifying the Representation of Dates, Times, and Numbers
- 14-107 Writing Software for Other Scripts
 - 14-107 Working With Fonts
 - 14-108 Working With Character Codes
 - 14-109 Working With Text Direction
 - 14-109 Synchronizing Keyboards and Fonts
 - 14-109 Handling Numbers
 - 14-110 Identifying Keywords and Tokens
 - 14-110 Possible Printing Problems
- 14-111 Summary of the Script Manager
- 14-133 Summary of the International Utilities Package
- 14-138 Summary of the International Resources

15 TextEdit

- 15-3 About This Chapter
- 15-3 About TextEdit
- 15-4 Mixed-Directional Text
 - 15-6 Highlighting
 - 15-8 Mouse-Down Regions
 - 15-9 Dual Carets
- 15-11 Caret Movement Across Direction Boundaries
- 15-11 Font and Keyboard Synchronization
- 15-13 Double-Byte Characters
- 15-14 Vertical Movement of the Caret
- 15-14 Arrow Key Actions for Selected Text
- 15-14 Caret Position at Line Ends
- 15-15 Word Selection and Line Breaks
- 15-16 Accurate Line Measurement
- 15-17 TextEdit and TrueType Fonts
- 15-17 Using TextEdit
 - 15-17 Determining the Version of TextEdit
 - 15-18 Customizing TextEdit's Features
 - 15-19 Measuring the Width of Components of a Line
 - 15-19 Defining Word Boundaries
 - 15-19 Controlling Outline Highlighting, Text Buffering, and Inline Input
 - 15-19 Setting Left Alignment for Right-to-Left Directional Scripts
 - 15-20 Using WordRedraw for Line Calculations
 - 15-20 Using the lineStarts Array to Determine Line Length
 - 15-21 Using TextEdit's Default Click Procedure
- 15-21 TextEdit Routines
 - 15-22 Outline Highlighting, Text Buffering, and Inline Input
 - 15-23 Outline Highlighting
 - 15-23 Text Buffering
 - 15-24 Inline Input

15-25	Customizing TextEdit
15-26	Replacing the End-of-Line Routine
15-26	Replacing the Measuring Routines
15-28	Replacing the Drawing Routine
15-29	Replacing the Hit Test Routine
15-29	Replacing the Word Breaking Routine
15-31	Backspacing to the Beginning of a Style
15-31	Determining the Position of an Ambiguous Offset
15-32	Toggling a Style
15-34	Determining Styles Across a Selection
15-36	Setting Styles in TextEdit's Scrap Record
15-36	Determining the Number of Styles
15-37	TextEdit Data Structures
15-40	Summary of TextEdit

16 Graphics Overview

16-3	About This Chapter
16-3	About Macintosh Graphics
16-6	The Components of Macintosh Graphics
16-6	QuickDraw
16-6	The Interface
16-8	The Video Card and Screen
16-9	The Major Data Structures
16-9	The RGB Color Record
16-10	Color Collections
16-11	The Pixel Map Record
16-12	Port Characteristics: The Color GrafPort Record
16-13	Device Characteristics: The Graphics Device Record
16-15	Graphics Initialization
16-16	The Graphics Path
16-16	How Indexed Pixels Work
16-18	How Direct Pixels Work
16-18	Determining the QuickDraw Version
16-19	What Else to Read
16-20	Summary of Graphics Data Types

17 Color QuickDraw

17-3	About This Chapter
17-3	About Color QuickDraw
17-4	Direct Pixels
17-4	Pixel Map Record Extensions
17-6	Direct Pixel Values
17-10	Writing Compatible Graphics Applications
17-11	Using Color QuickDraw
17-12	Manipulating Pixel Map Images
17-12	Copying With Masks
17-16	Colorizing
17-16	Transfer Modes
17-17	Dithering
17-17	Resizing Images
17-18	Luminance Mapping

- 17-18 Image Resolution
- 17-19 Displaying Variable-Resolution Pixel Maps and Pictures
- 17-19 Exporting Pixel Map Records
- 17-19 Converting a Bitmap to a Region
- 17-20 Determining Whether Drawing Is Complete
- 17-20 Extensions to the Version 2 Picture Format
- 17-21 Font Name
- 17-21 Line Justification
- 17-21 Direct Pixel Images
- 17-24 Sample Extended Version 2 Picture
- 17-25 Color QuickDraw Routines
- 17-25 Creating an Extended Version 2 Picture
- 17-25 Creating Regions From Bitmaps
- 17-25 Copying Pixel Map Images
- 17-26 Determining Whether QuickDraw Has Finished Drawing
- 17-26 Reporting Data Structure Changes
- 17-27 Obtaining Intermediate Colors
- 17-28 Interpreting New QDError Result Codes
- 17-28 Using a Custom Color Search Function
- 17-29 Summary of Color QuickDraw

18 The Picture Utilities Package

- 18-3 About This Chapter
- 18-3 About the Picture Utilities Package
- 18-3 Using the Picture Utilities Package
- 18-4 Getting Color Information
- 18-5 Collecting Information From Multiple Pixel Maps or Pictures
- 18-5 Storing Information: The Picture Information Record
- 18-9 Picture Utilities Package Routines
- 18-9 Collecting Information From a Single Image
- 18-10 Collecting Information From Multiple Images
- 18-12 Creating Custom Color-Sampling Methods
- 18-15 Summary of the Picture Utilities Package

19 The Color Picker Package

- 19-3 About This Chapter
- 19-3 About the Color Picker Package
- 19-4 Color Models
- 19-4 The RGB Model
- 19-6 The CMYK Model
- 19-6 The HLS and HSV Models
- 19-8 Color Models in the Dialog Box
- 19-8 Using the Color Picker Package
- 19-8 Presenting the Color Picker Dialog Box
- 19-9 Using Conversion Facilities
- 19-9 Color Picker Package Routines
- 19-10 Displaying the Color Picker Dialog Box
- 19-10 Converting Between Color Models
- 19-11 Converting Between SmallFract and Fixed Values
- 19-12 Summary of the Color Picker Package

20 The Palette Manager

- 20-3 About This Chapter
- 20-3 About the Palette Manager
- 20-4 Palettes
- 20-5 Color Usage Categories
- 20-6 Changing the Color Environment
- 20-7 Restoring the Color Environment
- 20-8 Using the Palette Manager
- 20-8 Working With Color Usage Categories
- 20-9 Courteous Colors
- 20-10 Tolerant Colors
- 20-10 Animated Colors
- 20-11 Displaying Animated Colors on Direct Devices
- 20-12 Explicit Colors
- 20-12 Inhibited Colors
- 20-13 Combined Usage Categories
- 20-14 Creating Palettes
- 20-14 Assigning Colors to a Palette
- 20-14 Creating a Palette in a Resource File
- 20-16 Assigning a Default Palette to an Application
- 20-17 Linking a Color Table to a Palette
- 20-18 Associating One Palette With Many Ports
- 20-18 Palette Manager Routines
- 20-18 Initializing the Palette Manager
- 20-19 Initializing and Allocating Palettes
- 20-19 Interacting With the Window Manager
- 20-21 Drawing With Color Palettes
- 20-23 Animating Color Tables
- 20-23 Manipulating Palettes and Color Tables
- 20-24 Manipulating Palette Entries
- 20-26 Summary of the Palette Manager

21 The Graphics Devices Manager

- 21-3 About This Chapter
- 21-3 About the Graphics Devices Manager
- 21-4 Offscreen Graphics
- 21-5 Using the Graphics Devices Manager
- 21-5 Offscreen Graphics World Flags
- 21-6 Example of Offscreen Graphics Code
- 21-7 Advanced Features of the Graphics Devices Manager
- 21-7 The Graphics Device Record
- 21-10 The 'sorn' Resource
- 21-12 Setting a Device's Pixel Depth
- 21-12 Graphics Devices Manager Routines
- 21-12 High-Level Routines
- 21-12 Creating a Graphics World
- 21-14 Working With a Graphics World's Pixel Map
- 21-16 Updating the Graphics World
- 21-18 Setting and Retrieving the Graphics World
- 21-19 Disposing of a Graphics World
- 21-19 Low-Level Routines
- 21-20 Supporting the Offscreen Graphics World

- 21-20 Managing the Graphics Device Record
- 21-25 Summary of the Graphics Devices Manager

22 The Sound Manager

- 22-3 About This Chapter
- 22-4 About the Sound Manager
- 22-6 Introduction to Sound
- 22-8 Sound Synthesizers
 - 22-8 The Square-Wave Synthesizer
 - 22-9 The Wave-Table Synthesizer
 - 22-10 The Sampled Sound Synthesizer
- 22-10 Sound Commands
- 22-12 Sound Channels
 - 22-12 Multiple Channels of Sound
 - 22-13 Sound Compression and Expansion
 - 22-15 Continuous Play From Disk
 - 22-16 Sound Recording
- 22-18 Sound Storage Formats
- 22-18 Sound Resources
 - 22-20 The Format 1 'snd ' Resource
 - 22-24 The Format 2 'snd ' Resource
- 22-24 Sound Files
 - 22-26 Chunk Organization and Data Types
 - 22-27 The Form Chunk
 - 22-28 The Format Version Chunk
 - 22-29 The Common Chunk
 - 22-31 The Sound Data Chunk
 - 22-32 Reading and Writing Sound Files
- 22-34 Using the Sound Manager
 - 22-35 Playing 'snd ' Resources
 - 22-36 Allocating Sound Channels
 - 22-38 Initializing Sound Channels
 - 22-40 Releasing Sound Channels
 - 22-40 Determining Features of Synthesizers
 - 22-41 Playing Frequencies
 - 22-43 Playing Sampled Sounds
 - 22-50 Installing Voices Into Channels
 - 22-50 Manipulating a Sound That Is Playing
 - 22-52 Flushing Sound Channels
 - 22-52 Pausing and Restarting Sound Channels
 - 22-53 Synchronizing Sound Channels
 - 22-53 Managing the CPU Load
 - 22-54 Producing an Alert Sound
 - 22-55 Compressing and Expanding Sounds
 - 22-56 Playing Sampled Sounds From Files
 - 22-56 Playing an 'snd ' Resource From Disk
 - 22-57 Playing a File From Disk
 - 22-57 Playing Selections
 - 22-58 Recording Sounds Through the Sound Input Dialog Box
 - 22-59 Recording Sounds Directly From a Device
 - 22-62 Defining a Sound Input Completion Routine
 - 22-63 Defining an Interrupt Routine
 - 22-63 Getting and Setting Sound Input Device Information

22-68	Obtaining Information About Sound Features
22-69	Obtaining Information About Available Sound Features
22-69	Obtaining Version Information
22-70	Obtaining Information About a Single Sound Channel
22-71	Obtaining Information About All Sound Channels
22-72	Using Double Buffers
22-74	Setting Up Double Buffers
22-76	Writing a Doubleback Procedure
22-77	Specifying Callback Routines
22-79	Sound Manager Routines
22-79	Playing Sound Resources
22-81	Allocating and Releasing Sound Channels
22-82	Linking Synthesizers to Sound Channels
22-82	Sending Commands to a Sound Channel
22-87	Obtaining Information
22-88	Playing From Disk
22-90	Managing Double Buffers
22-91	Compressing and Expanding Audio Data
22-92	Recording Sounds
22-94	Manipulating Sound Input Devices
22-94	Opening and Closing Sound Input Devices
22-94	Recording Sounds Directly From Sound Input Devices
22-98	Manipulating Device Settings
22-99	Constructing Sound Resource and File Headers
22-101	Registering Sound Input Devices
22-102	Converting Between Milliseconds and Bytes
22-103	Summary of the Sound Manager

23 The Time Manager

23-3	About This Chapter
23-3	About the Time Manager
23-4	Time Manager Versions
23-4	The Original Time Manager
23-5	The Revised Time Manager
23-6	The Extended Time Manager
23-8	Other Time-Related Facilities
23-9	The TickCount Function
23-9	The Delay Function
23-9	The Vertical Retrace Manager
23-10	Using the Time Manager
23-11	Installing and Activating Task Records
23-12	Using Application Global Variables in Tasks
23-14	Performing Periodic Tasks
23-14	Computing Elapsed Time
23-16	Time Manager Routines
23-18	Summary of the Time Manager

24 The Notification Manager

24-3	About This Chapter
24-3	About the Notification Manager
24-6	Using the Notification Manager
24-6	Creating a Notification Request

- 24-8 Defining a Response Procedure
- 24-9 Installing a Notification Request
- 24-10 Removing a Notification Request
- 24-10 Notification Manager Routines
- 24-12 Summary of the Notification Manager

25 The File Manager

- 25-3 About This Chapter
- 25-3 About the File Manager
- 25-4 Identifying Files, Directories, and Volumes
 - 25-4 File System Specifications
 - 25-5 The Evolution of File Specification Strategies
- 25-9 Limitations on MFS Disks
 - 25-9 A Simpler Safe-Save Strategy
- 25-10 New Special-Purpose Features
 - 25-10 A Quick, Thorough Catalog Search
 - 25-10 File IDs
 - 25-11 Shared Environments
 - 25-11 Remote Mounting
 - 25-12 Privilege Information in Foreign File Systems
 - 25-12 Using the File Manager
 - 25-15 Using FSSpec Records
 - 25-17 Updating Files
 - 25-20 Searching a Volume
 - 25-25 Tracking Files With File IDs
 - 25-26 Mounting Volumes Programmatically
 - 25-28 Manipulating Privilege Information in Foreign File Systems
 - 25-29 High-Level File Manager Routines
 - 25-29 Routines That Use FSSpec Records
 - 25-30 Making FSSpec Records
 - 25-31 Exchanging the Data in Two Files
 - 25-31 Functions Modified to Accept FSSpec Records
 - 25-34 Opening a Data Fork
 - 25-34 Managing HFS
 - 25-34 Functions New With HFS
 - 25-35 MFS Functions Modified to Accommodate Directory IDs
 - 25-37 Low-Level File Manager Routines
 - 25-37 Reading Volume Information
 - 25-40 Searching a Catalog
 - 25-42 Creating FSSpec Records
 - 25-43 Swapping Data Between Two Files
 - 25-44 Creating and Using File IDs
 - 25-45 Functions for Manipulating File IDs
 - 25-47 Functions Changed to Accommodate File IDs
 - 25-48 Mounting Volumes
 - 25-50 Accessing Privilege Information in Foreign File Systems
 - 25-52 Opening Data Forks
 - 25-54 Summary of the File Manager

26 The Standard File Package

- 26-3 About This Chapter
- 26-3 About the Standard File Package
- 26-4 Using the Standard File Package
- 26-5 Presenting the Default Interface
- 26-9 Customizing Your Interface
- 26-10 Customized Dialog Boxes
- 26-11 Callback Routines
- 26-20 Compatibility With Earlier Procedures
- 26-20 Standard File Package Routines
- 26-20 Saving Files
- 26-22 Opening Files
- 26-25 Summary of the Standard File Package

27 The Alias Manager

- 27-3 About This Chapter
- 27-3 About the Alias Manager
- 27-4 About Alias Records
- 27-6 Search Strategies for Resolving Alias Records
- 27-7 Fast Search
- 27-8 Exhaustive Search
- 27-8 Using the Alias Manager
- 27-8 Creating Alias Records
- 27-9 Resolving Alias Records
- 27-10 ResolveAlias
- 27-11 MatchAlias
- 27-11 Maintaining Alias Records
- 27-12 Getting Information About Alias Records
- 27-12 Customizing Alias Records
- 27-12 Alias Manager Routines
- 27-12 Creating and Updating Alias Records
- 27-15 Resolving and Reading Alias Records
- 27-20 Filtering Possible Targets
- 27-21 Summary of the Alias Manager

28 Memory Management

- 28-3 About This Chapter
- 28-4 About Memory Management
- 28-6 Dividing Memory Among Multiple Applications
- 28-7 Extending an Application's Available Memory
- 28-9 Extending the Operating System's Available Memory
- 28-9 Controlling the System Memory Settings
- 28-10 About the Memory Manager
- 28-11 Using Master Pointers
- 28-12 Using Window and Control Definition Functions
- 28-12 Manipulating 24-Bit and 32-Bit Memory Addresses
- 28-13 Using the Memory Manager
- 28-14 Setting and Restoring the A5 Register
- 28-15 Manipulating A5 Without MPW
- 28-16 Memory Manager Routines
- 28-16 Setting and Restoring the A5 World
- 28-16 Manipulating Memory Addresses

28-17	About Virtual Memory
28-18	Using Virtual Memory
28-19	Holding and Releasing Memory
28-20	Locking and Unlocking Memory
28-21	Obtaining Information About Virtual Memory
28-21	Information About the System Memory Configuration
28-21	Information About Page Mapping
28-23	Deferring User Interrupt Handling
28-23	Debugger Support Under Virtual Memory
28-24	Bus Error Vectors
28-25	Special Nonmaskable Interrupt Needs
28-25	Supervisor Mode
28-25	The Debugging State
28-26	Keyboard Input
28-26	Page States
28-26	Virtual Memory Routines
28-26	Holding and Releasing Pages
28-28	Locking and Unlocking Pages
28-29	Obtaining Page-Mapping Information
28-30	Deferring User Interrupt Handling
28-30	Determining Which Debugger Functions Are Present
28-31	Determining Whether Paging Is Safe
28-31	Locking and Unlocking Memory With Caching Enabled
28-32	Entering and Exiting the Debugging State
28-33	Obtaining Keyboard Input
28-33	Determining Page State
28-33	About Temporary Memory
28-36	Using Temporary Memory
28-36	Allocating Temporary Memory
28-37	Locking Temporary Memory
28-37	Unlocking Temporary Memory
28-37	Releasing Temporary Memory
28-37	Determining Features of Temporary Memory
28-38	Temporary Memory Routines
28-38	Requesting Temporary Memory
28-39	Locking and Unlocking Temporary Memory
28-40	Freeing Temporary Memory
28-41	Summary of Memory Management

29 Process Management

29-3	About This Chapter
29-4	About Process Management
29-5	How the Process Manager Creates Processes
29-6	How the Process Manager Schedules Processes
29-7	How Your Application Specifies Scheduling Options
29-8	Using the Process Manager
29-8	Opening or Printing Files Based on Finder Information
29-8	Getting Information About Other Processes
29-12	Launching Other Applications
29-15	Specifying Launch Options
29-16	Controlling Launched Applications
29-16	Launching Desk Accessories

- 29-16 Process Manager Routines
- 29-17 Getting Process Information
- 29-20 Launching Applications and Desk Accessories
- 29-24 Summary of Process Manager Routines

30 The Slot Manager

- 30-3 About This Chapter
- 30-4 About the Slot Manager
- 30-4 Card Initialization
- 30-6 Using the Slot Manager
 - 30-6 Enabling and Disabling NuBus Cards
 - 30-7 Enabling and Disabling SResource Data Structures
 - 30-8 Searching for Disabled SResource Data Structures
 - 30-8 Restoring Deleted SResource Data Structures
- 30-10 Slot Manager Routines
 - 30-11 The Slot Manager Parameter Block
 - 30-13 Determining the Version of the Slot Manager
 - 30-13 Getting Information About SResource Data Structures
 - 30-16 Enabling, Disabling, or Restoring SResource Data Structures
 - 30-19 Summary of the Slot Manager

31 The Power Manager

- 31-3 About This Chapter
- 31-4 About the Power Manager
- 31-5 The Idle State
- 31-6 The Sleep State
- 31-7 The Sleep Queue
 - 31-8 Sleep Requests
 - 31-10 Sleep Demands
 - 31-11 Wakeup Demands
 - 31-11 Sleep-Request Revocations
- 31-12 Using the Power Manager
 - 31-12 Determining Whether the Power Manager Is Present
 - 31-13 Enabling or Disabling the Idle State
 - 31-14 Setting, Disabling, and Reading the Wakeup Timer
 - 31-15 Placing a Routine in the Sleep Queue
 - 31-16 Responding When the Sleep Queue Calls Your Routine
 - 31-18 Switching Serial Power On and Off
- 31-19 Power Manager Routines
 - 31-19 Controlling the Idle State
 - 31-20 Controlling and Reading the Wakeup Timer
 - 31-21 Controlling the Sleep Queue
 - 31-21 Controlling Serial Power
 - 31-22 Reading the Status of the Internal Modem
 - 31-23 Reading the Status of the Battery and of the Battery Charger
 - 31-25 Summary of the Power Manager

32 The AppleTalk Manager

- 32-3 About This Chapter
- 32-4 About the AppleTalk Manager
- 32-4 Changes to the AppleTalk Manager

32-5	AppleTalk Protocols
32-8	AppleTalk Device Drivers, AppleTalk Connection Files, and the LAP Manager
32-11	Using the AppleTalk Manager
32-11	Determining Whether AppleTalk Phase 2 Drivers Are Present
32-11	Deciding Which AppleTalk Protocol to Use
32-12	The .MPP Driver
32-12	Getting Information About the .MPP Driver
32-16	A New NBP Wildcard Character
32-17	The LAP Manager
32-17	The AppleTalk Transition Queue
32-18	Adding and Removing AppleTalk Transition Queue Entries
32-20	Sending Messages to the AppleTalk Transition Queue
32-21	How the AppleTalk Manager Calls Your AppleTalk Transition Queue Entry
32-24	Defining Your Own AppleTalk Transition
32-25	The LAP Manager 802.2 Protocol
32-27	Attaching and Detaching 802.2 Protocol Handlers
32-29	The .ATP Driver
32-29	Canceling All Calls to the ATPGetRequest Function
32-30	Setting the Timeout Value for the ATP Release Timer
32-31	The .XPP Driver
32-32	Using the .XPP Driver to Obtain Information About Zones
32-34	Obtaining Zone Information
32-37	AppleTalk Data Stream Protocol (ADSP)
32-38	Using ADSP
32-38	The ADSP Connection Control Block
32-40	The .DSP Parameter Block
32-43	Opening and Maintaining an ADSP Connection
32-52	Creating and Using a Connection Listener
32-56	Writing a User Routine for Connection Events
32-58	.DSP Driver Routines
32-58	Establishing and Terminating an ADSP Connection
32-69	Establishing and Terminating an ADSP Connection Listener
32-73	Maintaining an ADSP Connection
32-78	The .ENET Driver
32-79	Providing Your Own Ethernet Driver
32-79	Changing the Ethernet Hardware Address
32-79	Opening the .ENET Driver
32-81	Using a Write-Data Structure to Transmit Ethernet Data
32-83	Using the Default Ethernet Protocol Handler to Read Data
32-85	Using Your Own Ethernet Protocol Handler to Read Data
32-85	How the .ENET Driver Calls Your Protocol Handler
32-86	How Your Protocol Handler Calls the .ENET Driver
32-88	.ENET Driver Routines
32-90	Attaching and Detaching an Ethernet Protocol Handler
32-91	Writing and Reading Ethernet Packets
32-95	Adding and Removing Ethernet Multicast Addresses
32-97	Summary of the AppleTalk Manager

A-1 **A** Result Codes

B-1 **B** Routines and Their Memory Behavior

C-1	C	System Traps
D-1	D	Global Variables
E-1	E	The Standard Roman Character Set
GL-1		Glossary
IN-1		Index

Figures, Tables, and Listings

Color Plates

Color plates are immediately preceding the title page.

Color Plate I.	Examples of keyboard icons
Color Plate II.	A colorized window
Color Plate III.	A colorized movable modal dialog box
Color Plate IV.	Design for black-and-white monitors first
Color Plate V.	Don't mimic color effects in black-and-white designs
Color Plate VI.	Use light colors for large areas
Color Plate VII.	Don't use bright colors for large areas
Color Plate VIII.	Use bright colors for details
Color Plate IX.	A consistent light source
Color Plate X.	Inconsistent light sources
Color Plate XI.	An icon family
Color Plate XII.	Consistently designed small icons
Color Plate XIII.	Inconsistently designed small icons
Color Plate XIV.	Icons with a black outline
Color Plate XV.	Icons without a black outline
Color Plate XVI.	Apple icon colors
Color Plate XVII.	Correct anti-aliasing
Color Plate XVIII.	Incorrect anti-aliasing
Color Plate XIX.	Consistent use of icon elements
Color Plate XX.	Inconsistent use of icon elements
Color Plate XXI.	Default system icon families
Color Plate XXII.	Examples of control panel icons
Color Plate XXIII.	Copying with a pixel map as a mask
Color Plate XXIV.	Colorizing
Color Plate XXV.	Default color tables

Preface

P-17	Figure P-1.	A road map to Volume VI
------	-------------	-------------------------

1 Introduction to the System Software Version 7.0 Environment

1-5	Figure 1-1.	Features of the system software version 7.0 environment
1-7	Figure 1-2.	The managers constituting the interapplication communications architecture
1-9	Figure 1-3.	Using interapplication communication
1-9	Figure 1-4.	Applications using interapplication communication
1-10	Figure 1-5.	A publisher, an edition, and a subscriber
1-11	Figure 1-6.	Sharing dynamic data with other applications
1-13	Figure 1-7.	Sending events to other applications
1-14	Figure 1-8.	Requesting data from a remote database
1-17	Figure 1-9.	Comparison of TrueType and bitmapped fonts
1-22	Figure 1-10.	Using multiple scripts in a single document
1-25	Figure 1-11.	Overview of chapters in Volume VI

2 User Interface Guidelines

- 2-7 Figure 2-1. Dual carets in mixed-directional text
- 2-7 Figure 2-2. Multidirectional text correctly highlighted
- 2-9 Figure 2-3. Reversing the alignment of dialog box items
- 2-9 Figure 2-4. The Keyboard menu
- 2-12 Figure 2-5. The boundaries of a font
- 2-15 Figure 2-6. A progress indicator
- 2-15 Figure 2-7. The Application menu with a notification symbol
- 2-18 Figure 2-8. An icon family
- 2-19 Figure 2-9. A well-designed icon and its selected version
- 2-19 Figure 2-10. A poorly designed icon and its selected version
- 2-21 Figure 2-11. Default system icons in black and white
- 2-22 Figure 2-12. Examples of control panel icons
- 2-24 Figure 2-13. A movable modal dialog box
- 2-25 Figure 2-14. A Finder movable modal dialog box
- 2-26 Figure 2-15. A selected scrolling list
- 2-27 Figure 2-16. A dialog box with OK and Cancel buttons
- 2-27 Figure 2-17. A dialog box with OK instead of a Cancel button
- 2-28 Figure 2-18. A progress indicator that uses a Stop button
- 2-28 Figure 2-19. A confirmation alert box
- 2-29 Figure 2-20. The recommended spacing of buttons and text in a dialog box
- 2-30 Figure 2-21. A well-written dialog box message
- 2-31 Figure 2-22. The new standard file dialog box for opening files
- 2-32 Figure 2-23. The save changes dialog box
- 2-33 Figure 2-24. A sample Edit menu
- 2-33 Figure 2-25. A sample hierarchical menu with Edition Manager commands
- 2-34 Figure 2-26. A sample pull-down Size menu and font size dialog box
- 2-34 Figure 2-27. The Help menu
- 2-36 Figure 2-28. The appearance of a version 7.0 pop-up menu
- 2-36 Figure 2-29. An open version 7.0 pop-up menu
- 2-37 Figure 2-30. A type-in pop-up menu
- 2-37 Figure 2-31. A type-in pop-up menu with user's choice added

- 2-10 Table 2-1. Examples of keyboard icons
- 2-11 Table 2-2. Pattern substitutions for colors in keyboard icons
- 2-13 Table 2-3. Translation chart for user documentation
- 2-35 Table 2-4. Apple reserved keyboard equivalents for all systems
- 2-35 Table 2-5. Additional reserved keyboard equivalents for worldwide systems
- 2-35 Table 2-6. Other common keyboard equivalents

3 Compatibility Guidelines

- 3-12 Figure 3-1. The size menu for a bitmapped font
- 3-12 Figure 3-2. The size menu for an outline font
- 3-16 Figure 3-3. A pop-up menu in its inactive and active states
- 3-17 Figure 3-4. A pop-up control that is right-aligned
- 3-20 Figure 3-5. An initial dialog box and a list of items to append
- 3-20 Figure 3-6. The dialog box after items are overlaid
- 3-21 Figure 3-7. The dialog box after items are appended to the right
- 3-21 Figure 3-8. The dialog box after items are appended to the bottom
- 3-22 Figure 3-9. The dialog box after items are appended relative to Item 2
- 3-25 Figure 3-10. Interactions among an application, the A/UX Toolbox, and ROM code

3-26	Table 3-1.	Status of User Interface Toolbox and Macintosh Operating System libraries in the A/UX Toolbox
3-8	Listing 3-1.	Determining whether a trap is available
3-32	Listing 3-2.	Determining whether Gestalt is available
3-33	Listing 3-3.	Using Gestalt to determine the Time Manager version
3-34	Listing 3-4.	Interpreting a bit field response
3-43	Listing 3-5.	Installing a selector function into the system heap
3-44	Listing 3-6.	Defining a new Gestalt function

4 The Edition Manager

4-4	Figure 4-1.	The default edition icon
4-5	Figure 4-2.	A publisher, an edition, and a subscriber
4-5	Figure 4-3.	The publisher dialog box
4-6	Figure 4-4.	The subscriber dialog box
4-8	Figure 4-5.	A document and its corresponding editions
4-9	Figure 4-6.	Publisher and subscriber borders
4-10	Figure 4-7.	Edition Manager commands in the Edit menu
4-10	Figure 4-8.	Edition Manager commands under the Publishing menu command
4-15	Figure 4-9.	A document with a publisher and subscriber and its resource fork
4-17	Figure 4-10.	The new publisher alert box
4-26	Figure 4-11.	A sample publisher dialog box
4-32	Figure 4-12.	A sample subscriber dialog box
4-37	Figure 4-13.	The publisher options dialog box with update mode set to On Save
4-37	Figure 4-14.	The publisher options dialog box with update mode set to Manually
4-38	Figure 4-15.	The subscriber options dialog box with update mode set to Automatically
4-38	Figure 4-16.	The subscriber options dialog box with update mode set to Manually
4-44	Figure 4-17.	Edit menu with Show/Hide Borders menu command
4-45	Figure 4-18.	Publisher borders
4-46	Figure 4-19.	Subscriber borders
4-47	Figure 4-20.	A publisher with contents removed
4-48	Figure 4-21.	A publisher border within a spreadsheet document
4-49	Figure 4-22.	A publisher border with resize handles
4-50	Figure 4-23.	A publisher and subscriber with clipped graphics
4-51	Figure 4-24.	Creating multiple publishers alert box
4-51	Figure 4-25.	Saving multiple publishers alert box
4-55	Figure 4-26.	Subscribing directly to a 'PICT' file
4-13	Listing 4-1.	Accepting Section Read events and verifying if a section is registered
4-18	Listing 4-2.	Saving a document containing sections
4-20	Listing 4-3.	Opening a document containing sections
4-28	Listing 4-4.	Creating a publisher
4-30	Listing 4-5.	Writing data to an edition
4-33	Listing 4-6.	Creating a subscriber
4-36	Listing 4-7.	Reading in edition data
4-39	Listing 4-8.	Responding to action codes
4-58	Listing 4-9.	Using your own edition opener function

2 User Interface Guidelines

- 2-7 Figure 2-1. Dual carets in mixed-directional text
- 2-7 Figure 2-2. Multidirectional text correctly highlighted
- 2-9 Figure 2-3. Reversing the alignment of dialog box items
- 2-9 Figure 2-4. The Keyboard menu
- 2-12 Figure 2-5. The boundaries of a font
- 2-15 Figure 2-6. A progress indicator
- 2-15 Figure 2-7. The Application menu with a notification symbol
- 2-18 Figure 2-8. An icon family
- 2-19 Figure 2-9. A well-designed icon and its selected version
- 2-19 Figure 2-10. A poorly designed icon and its selected version
- 2-21 Figure 2-11. Default system icons in black and white
- 2-22 Figure 2-12. Examples of control panel icons
- 2-24 Figure 2-13. A movable modal dialog box
- 2-25 Figure 2-14. A Finder movable modal dialog box
- 2-26 Figure 2-15. A selected scrolling list
- 2-27 Figure 2-16. A dialog box with OK and Cancel buttons
- 2-27 Figure 2-17. A dialog box with OK instead of a Cancel button
- 2-28 Figure 2-18. A progress indicator that uses a Stop button
- 2-28 Figure 2-19. A confirmation alert box
- 2-29 Figure 2-20. The recommended spacing of buttons and text in a dialog box
- 2-30 Figure 2-21. A well-written dialog box message
- 2-31 Figure 2-22. The new standard file dialog box for opening files
- 2-32 Figure 2-23. The save changes dialog box
- 2-33 Figure 2-24. A sample Edit menu
- 2-33 Figure 2-25. A sample hierarchical menu with Edition Manager commands
- 2-34 Figure 2-26. A sample pull-down Size menu and font size dialog box
- 2-34 Figure 2-27. The Help menu
- 2-36 Figure 2-28. The appearance of a version 7.0 pop-up menu
- 2-36 Figure 2-29. An open version 7.0 pop-up menu
- 2-37 Figure 2-30. A type-in pop-up menu
- 2-37 Figure 2-31. A type-in pop-up menu with user's choice added

- 2-10 Table 2-1. Examples of keyboard icons
- 2-11 Table 2-2. Pattern substitutions for colors in keyboard icons
- 2-13 Table 2-3. Translation chart for user documentation
- 2-35 Table 2-4. Apple reserved keyboard equivalents for all systems
- 2-35 Table 2-5. Additional reserved keyboard equivalents for worldwide systems
- 2-35 Table 2-6. Other common keyboard equivalents

3 Compatibility Guidelines

- 3-12 Figure 3-1. The size menu for a bitmapped font
- 3-12 Figure 3-2. The size menu for an outline font
- 3-16 Figure 3-3. A pop-up menu in its inactive and active states
- 3-17 Figure 3-4. A pop-up control that is right-aligned
- 3-20 Figure 3-5. An initial dialog box and a list of items to append
- 3-20 Figure 3-6. The dialog box after items are overlaid
- 3-21 Figure 3-7. The dialog box after items are appended to the right
- 3-21 Figure 3-8. The dialog box after items are appended to the bottom
- 3-22 Figure 3-9. The dialog box after items are appended relative to Item 2
- 3-25 Figure 3-10. Interactions among an application, the A/UX Toolbox, and ROM code

3-26	Table 3-1.	Status of User Interface Toolbox and Macintosh Operating System libraries in the A/UX Toolbox
3-8	Listing 3-1.	Determining whether a trap is available
3-32	Listing 3-2.	Determining whether Gestalt is available
3-33	Listing 3-3.	Using Gestalt to determine the Time Manager version
3-34	Listing 3-4.	Interpreting a bit field response
3-43	Listing 3-5.	Installing a selector function into the system heap
3-44	Listing 3-6.	Defining a new Gestalt function

4 The Edition Manager

4-4	Figure 4-1.	The default edition icon
4-5	Figure 4-2.	A publisher, an edition, and a subscriber
4-5	Figure 4-3.	The publisher dialog box
4-6	Figure 4-4.	The subscriber dialog box
4-8	Figure 4-5.	A document and its corresponding editions
4-9	Figure 4-6.	Publisher and subscriber borders
4-10	Figure 4-7.	Edition Manager commands in the Edit menu
4-10	Figure 4-8.	Edition Manager commands under the Publishing menu command
4-15	Figure 4-9.	A document with a publisher and subscriber and its resource fork
4-17	Figure 4-10.	The new publisher alert box
4-26	Figure 4-11.	A sample publisher dialog box
4-32	Figure 4-12.	A sample subscriber dialog box
4-37	Figure 4-13.	The publisher options dialog box with update mode set to On Save
4-37	Figure 4-14.	The publisher options dialog box with update mode set to Manually
4-38	Figure 4-15.	The subscriber options dialog box with update mode set to Automatically
4-38	Figure 4-16.	The subscriber options dialog box with update mode set to Manually
4-44	Figure 4-17.	Edit menu with Show/Hide Borders menu command
4-45	Figure 4-18.	Publisher borders
4-46	Figure 4-19.	Subscriber borders
4-47	Figure 4-20.	A publisher with contents removed
4-48	Figure 4-21.	A publisher border within a spreadsheet document
4-49	Figure 4-22.	A publisher border with resize handles
4-50	Figure 4-23.	A publisher and subscriber with clipped graphics
4-51	Figure 4-24.	Creating multiple publishers alert box
4-51	Figure 4-25.	Saving multiple publishers alert box
4-55	Figure 4-26.	Subscribing directly to a 'PICT' file
4-13	Listing 4-1.	Accepting Section Read events and verifying if a section is registered
4-18	Listing 4-2.	Saving a document containing sections
4-20	Listing 4-3.	Opening a document containing sections
4-28	Listing 4-4.	Creating a publisher
4-30	Listing 4-5.	Writing data to an edition
4-33	Listing 4-6.	Creating a subscriber
4-36	Listing 4-7.	Reading in edition data
4-39	Listing 4-8.	Responding to action codes
4-58	Listing 4-9.	Using your own edition opener function

5 The Event Manager

- 5-6 Figure 5-1. Events in a single application environment
- 5-7 Figure 5-2. Events in a multi-application environment
- 5-9 Figure 5-3. Events in system software version 7.0

- 5-11 Listing 5-1. A simple event loop
- 5-11 Listing 5-2. Processing events
- 5-14 Listing 5-3. A template for a 'SIZE' resource
- 5-17 Listing 5-4. The Rez input for a sample 'SIZE' resource
- 5-23 Listing 5-5. Posting a high-level event by application signature
- 5-25 Listing 5-6. Using the PPCBrowser function to post a high-level event
- 5-27 Listing 5-7. Accepting a high-level event

6 The Apple Event Manager

- 6-5 Figure 6-1. An Open Documents event
- 6-8 Figure 6-2. Major components of an Open Documents event
- 6-10 Figure 6-3. A descriptor record with event class data
- 6-11 Figure 6-4. A keyword-specified descriptor record for the event class attribute of an Open Documents event
- 6-12 Figure 6-5. A descriptor list for a list of aliases
- 6-13 Figure 6-6. Data structures within an Open Documents event
- 6-15 Figure 6-7. Accepting and processing an Open Documents event
- 6-16 Figure 6-8. The Apple Event Manager calling the handler for an Open Documents event
- 6-20 Figure 6-9. Responding to an Open Documents event

- 6-69 Table 6-1. Coercion handling provided by the Apple Event Manager

- 6-26 Listing 6-1. A DoEvent procedure
- 6-27 Listing 6-2. A DoHighLevelEvent procedure for handling Apple events and other high-level events
- 6-28 Listing 6-3. Inserting entries for required Apple events into an application's Apple event dispatch table
- 6-30 Listing 6-4. Inserting entries for Apple events sent by the Edition Manager into an application's Apple event dispatch table
- 6-33 Listing 6-5. A handler for the Open Application event
- 6-34 Listing 6-6. A handler for the Open Documents event
- 6-35 Listing 6-7. A handler for the Print Documents event
- 6-36 Listing 6-8. A handler for the Quit Application event
- 6-39 Listing 6-9. A handler for the Create Publisher event
- 6-46 Listing 6-10. Extracting items from a descriptor list
- 6-47 Listing 6-11. A function that checks for a keyMissedKeywordAttr attribute
- 6-49 Listing 6-12. Adding the keyErrorString parameter to the reply Apple event
- 6-50 Listing 6-13. Adding parameters to the reply Apple event
- 6-54 Listing 6-14. Using the AEInteractWithUser function
- 6-58 Listing 6-15. Creating a target address
- 6-58 Listing 6-16. Specifying a target address in an Apple event
- 6-61 Listing 6-17. Sending an Apple event
- 6-66 Listing 6-18. An idle function

7 The Program-to-Program Communications Toolbox

- 7-5 Figure 7-1. A PPC Toolbox session between two applications
- 7-6 Figure 7-2. The Sharing Setup icon
- 7-6 Figure 7-3. The Sharing Setup control panel
- 7-7 Figure 7-4. The session termination alert box
- 7-7 Figure 7-5. The users and groups dialog box
- 7-8 Figure 7-6. The user termination alert box
- 7-8 Figure 7-7. The guest dialog box
- 7-9 Figure 7-8. The PPC Toolbox authentication process
- 7-11 Figure 7-9. Database and spreadsheet applications using the PPC Toolbox
- 7-15 Figure 7-10. Two Macintosh applications and their corresponding ports
- 7-17 Figure 7-11. The PPC Toolbox and a dictionary service application
- 7-18 Figure 7-12. The program linking dialog box
- 7-19 Figure 7-13. The program linking dialog box without a zone list
- 7-24 Figure 7-14. The user identity dialog box
- 7-25 Figure 7-15. The incorrect password dialog box
- 7-25 Figure 7-16. The invalid user name dialog box
- 7-32 Figure 7-17. Transmitting message blocks
- 7-38 Figure 7-18. The PPC Toolbox parameter blocks

- 7-10 Listing 7-1. Initializing the PPC Toolbox using the PPCInit function
- 7-17 Listing 7-2. Opening a PPC port
- 7-20 Listing 7-3. Using a port filter function
- 7-21 Listing 7-4. Browsing through dictionary service ports
- 7-22 Listing 7-5. Using the IPCListPorts function to obtain a list of ports
- 7-26 Listing 7-6. Using the StartSecureSession function to establish a session
- 7-27 Listing 7-7. Initiating a session using the PPCStart function
- 7-29 Listing 7-8. Using the PPCInform function to enable a port to receive sessions
- 7-29 Listing 7-9. Completion routine for a PPCInform function
- 7-30 Listing 7-10. Accepting a session request using the PPCAccept function
- 7-31 Listing 7-11. Completion routine for a PPCAccept function
- 7-31 Listing 7-12. Rejecting a session request using the PPCReject function
- 7-31 Listing 7-13. Completion routine for a PPCReject function
- 7-33 Listing 7-14. Using the PPCRead function to read data during a session
- 7-33 Listing 7-15. Polling the ioResult field to determine if a PPCRead function has completed
- 7-34 Listing 7-16. Using the PPCWrite function to write data during a session
- 7-34 Listing 7-17. Polling the ioResult field to determine if a PPCWrite function has completed
- 7-35 Listing 7-18. Ending a PPC session using the PPCEnd function
- 7-35 Listing 7-19. Closing a PPC port using the PPCClose function
- 7-36 Listing 7-20. Using the DeleteUserIdentity function to invalidate a user identity

8 The Data Access Manager

- 8-5 Figure 8-1. A connection with a database
- 8-7 Figure 8-2. Using high-level Data Access Manager routines
- 8-8 Figure 8-3. Using low-level Data Access Manager routines
- 8-13 Figure 8-4. A flowchart of a session using the high-level interface
- 8-27 Figure 8-5. A flowchart of a session using the low-level interface
- 8-43 Figure 8-6. A query document dialog box
- 8-43 Figure 8-7. Function of a query document

- 8-35 Table 8-1. Data types defined by the Data Access Manager
- 8-15 Listing 8-1. Using the high-level interface
- 8-23 Listing 8-2. A sample status routine
- 8-29 Listing 8-3. Sending a query fragment
- 8-30 Listing 8-4. Using the low-level interface
- 8-41 Listing 8-5. A result handler
- 8-48 Listing 8-6. A query definition function

9 The Finder Interface

- 9-10 Figure 9-1. Default large black-and-white icons
- 9-10 Figure 9-2. An application icon and its mask
- 9-11 Figure 9-3. The ResEdit view of an icon and its mask
- 9-13 Figure 9-4. Examples of document icons
- 9-18 Figure 9-5. Linking 'ICN#' and 'FREF' resources in a 'BNDL' resource
- 9-21 Figure 9-6. The default application-unavailable alert box
- 9-22 Figure 9-7. The application-unavailable alert box specifying an application's name
- 9-22 Figure 9-8. The application-unavailable alert box with a customized message
- 9-23 Figure 9-9. The application-unavailable alert box for 'TEXT' and 'PICT' documents
- 9-24 Figure 9-10. The version data in the information window
- 9-35 Figure 9-11. Default and customized help balloons for application icons
- 9-39 Figure 9-12. The System Folder and related folders
- 9-11 Listing 9-1. 'ICN#' resources for an application and its documents
- 9-14 Listing 9-2. Using file reference resources
- 9-16 Listing 9-3. Using a bundle resource
- 9-31 Listing 9-4. Using the ResolveAliasFile function to open a file
- 9-35 Listing 9-5. Creating a help balloon resource for an application icon

10 Control Panels

- 10-5 Figure 10-1. The Monitors control panel
- 10-6 Figure 10-2. An Options dialog box for a Monitors control panel
- 10-12 Figure 10-3. An Options dialog box with superuser controls
- 10-15 Listing 10-1. Sample of an extension to the Monitors control panel
- 10-22 Listing 10-2. Resources for a file that extends the Monitors control panel

11 The Help Manager

- 11-4 Figure 11-1. The Help menu for the Finder
- 11-6 Figure 11-2. A help balloon drawn with the standard balloon definition function
- 11-6 Figure 11-3. The tip and hot rectangle for a help balloon
- 11-7 Figure 11-4. Standard balloon positions and their variation codes
- 11-8 Figure 11-5. Alternate positions of a help balloon
- 11-12 Figure 11-6. Default help balloons for the window frame
- 11-13 Figure 11-7. Default help balloons for the Apple and Help menus
- 11-13 Figure 11-8. Default help balloons for application and document icons
- 11-22 Figure 11-9. Help balloons for different states of the Cut command
- 11-32 Figure 11-10. Help balloons for a changing menu item
- 11-40 Figure 11-11. A help balloon in a modal dialog box
- 11-41 Figure 11-12. Static and dynamic windows

- 11-46 Figure 11-13. A tool palette with a help balloon
- 11-47 Figure 11-14. A help balloon for a dialog box with a title
- 11-58 Figure 11-15. Default and customized help balloons for an application icon
- 11-61 Figure 11-16. The Help menu with two appended menu items

- 11-23 Listing 11-1. A partial menu help resource
- 11-27 Listing 11-2. The missing items component in a menu help resource
- 11-29 Listing 11-3. Corresponding 'hmnu' and 'STR#' resources
- 11-31 Listing 11-4. Using HMCompareItem for a changing menu item
- 11-38 Listing 11-5. A dialog item list and its help resource
- 11-46 Listing 11-6. Corresponding 'hwin' and 'hrct' resources for a tool palette
- 11-48 Listing 11-7. Specifying help for titled and untitled windows with an 'hwin' resource

- 11-55 Listing 11-8. Using HMShowBalloon to display help balloons
- 11-58 Listing 11-9. Creating a help balloon resource for an application icon
- 11-60 Listing 11-10. Overriding default help balloons
- 11-62 Listing 11-11. A sample menu help resource for items in the Help menu

12 The Font Manager

- 12-5 Figure 12-1. The standard Roman character set
- 12-6 Figure 12-2. Terms for font measurements
- 12-7 Figure 12-3. The ascent line and maximum y-value
- 12-8 Figure 12-4. A comparison of scaled TrueType and bitmapped fonts
- 12-9 Figure 12-5. A glyph stretched horizontally
- 12-9 Figure 12-6. A glyph stretched vertically
- 12-9 Figure 12-7. A glyph condensed horizontally
- 12-10 Figure 12-8. The effect of an off-curve point on two Bézier curves
- 12-11 Figure 12-9. An outline with points on and off the curve
- 12-12 Figure 12-10. A curve with consecutive off-curve points
- 12-12 Figure 12-11. An outline glyph
- 12-13 Figure 12-12. An unmodified outline glyph at a small point size
- 12-14 Figure 12-13. An instructed outline glyph
- 12-15 Figure 12-14. A sample Size menu and font size dialog box
- 12-17 Figure 12-15. The difference between a scaled glyph and a preserved glyph

- 12-16 Listing 12-1. Checking a font family ID against the font name

13 The Resource Manager

- 13-4 Table 13-1. Resource types available for your application's use
- 13-6 Table 13-2. Resource types reserved for the Operating System's use

- 13-17 Listing 13-1. Using partial resource calls

14 Worldwide Software Overview

- 14-9 Figure 14-1. The script, language, and region hierarchy
- 14-16 Figure 14-2. Types of script systems
- 14-17 Figure 14-3. The components of the Macintosh Script Management System
- 14-18 Figure 14-4. Worldwide control panels and desk accessory
- 14-19 Figure 14-5. Default icons for keyboards, fonts, and scripts
- 14-21 Figure 14-6. The Keyboard menu
- 14-22 Figure 14-7. Distinguishing scripts with resource ID ranges for script codes 0-32
- 14-24 Figure 14-8. Keyboard translation

14-29	Figure 14-9.	Scripts
14-30	Figure 14-10.	Alphabetic, syllabic, and ideographic representations of characters
14-31	Figure 14-11.	Three text directions
14-31	Figure 14-12.	Contextual forms in cursive English
14-31	Figure 14-13.	Stand-alone and contextual forms in Arabic
14-32	Figure 14-14.	A ligature in Roman text
14-32	Figure 14-15.	Ligatures in Arabic text
14-33	Figure 14-16.	Character reordering in Devanagari script (for the word <i>hindi</i>)
14-36	Figure 14-17.	Backing-store and display order
14-38	Figure 14-18.	Calling the Script Manager routines implemented by a script system
14-38	Figure 14-19.	Calling the Script Manager routines
14-52	Figure 14-20.	Forward operation of the state machine for word selection
14-53	Figure 14-21.	NFindWord header and class tables
14-54	Figure 14-22.	NFindWord state table
14-54	Figure 14-23.	Format of NFindWord action code
14-57	Figure 14-24.	Roman word selection state transitions
14-69	Figure 14-25.	Justification in Roman text
14-70	Figure 14-26.	The effects of the Roman justification routines
14-93	Figure 14-27.	The 'itl2' resource header
14-96	Figure 14-28.	Inside the 'KCHR' resource
14-99	Figure 14-29.	Format of entries in the 'KSWP' resource
14-99	Figure 14-30.	The 'itlk' resource entries
14-100	Figure 14-31.	Key Caps display of dead keys with Option key pressed
14-101	Figure 14-32.	Key Caps display of completer keys after circumflex dead key has been pressed
14-108	Figure 14-33.	Extracting blocks of Roman text
14-41	Table 14-1.	GetScript and SetScript verbs
14-42	Table 14-2.	Verbs for GetEnviron and SetEnviron
14-44	Table 14-3.	Verbs for the KeyScript procedure
14-55	Table 14-4.	U.S. word selection algorithm
14-56	Table 14-5.	Occurrence of word breaks in various character sequences
14-56	Table 14-6.	Significance of the state numbers in the Roman word selection algorithm
14-58	Table 14-7.	U.S. word select transition table for forward processing
14-66	Table 14-8.	Proportions of slop value to be distributed
14-80	Table 14-9.	The international and keyboard resources
14-84	Table 14-10.	Script, language, and region codes
14-95	Table 14-11.	The keyboard types
14-95	Table 14-12.	The keyboard modifier bits
14-97	Table 14-13.	Changes in handling 'KCHR' (0) key combinations
14-98	Table 14-14.	Changes in 'KCHR' dead-key completers
14-98	Table 14-15.	Keyboard color icon types and standard icon equivalents
14-59	Listing 14-1.	Obtaining optimized word break tables
14-61	Listing 14-2.	Substituting and truncating text
14-66	Listing 14-3.	Distributing slop value among style runs
14-83	Listing 14-4.	Multiscript mapping and sorting
14-88	Listing 14-5.	International date and time information
14-101	Listing 14-6.	Physical layout of keyboards
14-105	Listing 14-7.	Making keyboard equivalents work with multiple scripts

15 TextEdit

- 15-5 Figure 15-1. A right-to-left primary line direction
- 15-5 Figure 15-2. The display order
- 15-5 Figure 15-3. The backing-store order
- 15-6 Figure 15-4. Different levels of runs in a line of text
- 15-7 Figure 15-5. Discontinuous highlighting display
- 15-7 Figure 15-6. Highlighting mixed-directional text
- 15-8 Figure 15-7. Mouse-down region specifics
- 15-9 Figure 15-8. Dual carets in mixed-directional text
- 15-9 Figure 15-9. Pasting styled text
- 15-10 Figure 15-10. Dual carets at a direction boundary
- 15-11 Figure 15-11. Caret movement across a direction boundary
- 15-12 Figure 15-12. Font and keyboard script synchronization
- 15-15 Figure 15-13. The caret position at line end
- 15-16 Figure 15-14. Word breaks for word selection
- 15-16 Figure 15-15. A line break with multiple scripts installed
- 15-32 Figure 15-16. A character offset at a line break
- 15-33 Figure 15-17. An initial selection before TESSetStyle is called
- 15-33 Figure 15-18. The result of calling TESSetStyle to toggle with a bold style
- 15-33 Figure 15-19. The result of calling TESSetStyle to toggle with an italic style
- 15-38 Figure 15-20. The TextEdit data structures

- 15-20 Table 15-1. Constants for the just parameter of TESSetJust
- 15-34 Listing 15-1. Marking the Style menu items so they correspond to the current selection
- 15-35 Listing 15-2. Determining the font, face, size, and color of the current selection

16 Graphics Overview

- 16-4 Figure 16-1. Macintosh graphics chapters
- 16-7 Figure 16-2. QuickDraw and the graphics managers
- 16-15 Figure 16-3. Initializing Macintosh graphics
- 16-17 Figure 16-4. The indexed pixel path
- 16-18 Figure 16-5. The direct pixel path

17 Color QuickDraw

- 17-6 Figure 17-1. A 32-bit direct pixel
- 17-6 Figure 17-2. A 16-bit direct pixel
- 17-7 Figure 17-3. Converting a 16-bit direct pixel to a 32-bit direct pixel
- 17-7 Figure 17-4. Converting a 48-bit RGB color to a 32-bit direct pixel
- 17-8 Figure 17-5. Converting a 48-bit RGB color to a 16-bit direct pixel
- 17-8 Figure 17-6. Converting a 48-bit RGB color to an 8-bit indexed pixel
- 17-9 Figure 17-7. Converting a 32-bit pixel to a 48-bit RGB color
- 17-9 Figure 17-8. Converting a 16-bit pixel to a 48-bit RGB color
- 17-10 Figure 17-9. Converting an 8-bit indexed pixel to a 48-bit RGB color
- 17-13 Figure 17-10. Copying pixel maps with CopyBits
- 17-14 Figure 17-11. Copying pixel maps with CopyMask
- 17-15 Figure 17-12. Copying pixel maps with CopyDeepMask

- 17-18 Table 17-1. The default color tables for gray-scale devices
- 17-18 Table 17-2. The default color tables for color devices
- 17-20 Table 17-3. The new version 2 picture opcodes
- 17-24 Table 17-4. Version 2 picture example

19 The Color Picker Package

- 19-4 Figure 19-1. The Color Picker dialog box
- 19-5 Figure 19-2. The RGB color cube
- 19-5 Figure 19-3. Getting to pink
- 19-6 Figure 19-4. Cyan, magenta, and yellow on the color cube
- 19-7 Figure 19-5. The HLS/HSV color cone

20 The Palette Manager

- 20-9 Figure 20-1. A courteous palette
- 20-14 Listing 20-1. A palette ('pltt') resource
- 20-15 Listing 20-2. A multi-use palette

21 The Graphics Devices Manager

- 21-7 Listing 21-1. Sample offscreen graphics world

22 The Sound Manager

- 22-7 Figure 22-1. The position of the Sound Manager
- 22-7 Figure 22-2. Bypassing the command queue
- 22-8 Figure 22-3. Mixing multiple channels of sampled sound
- 22-9 Figure 22-4. A graph of a wave table
- 22-16 Figure 22-5. The Sound control panel
- 22-17 Figure 22-6. The sound recording dialog box
- 22-19 Figure 22-7. The structure of 'snd' resources
- 22-19 Figure 22-8. The location of the data offset bit
- 22-27 Figure 22-9. The general structure of a chunk
- 22-32 Figure 22-10. Interleaving stereo sample points
- 22-33 Figure 22-11. A sample AIFF-C file

- 22-14 Table 22-1. Audio compression and expansion options
- 22-25 Table 22-2. AIFF and AIFF-C capabilities
- 22-43 Table 22-3. MIDI values
- 22-45 Table 22-4. Sample rates
- 22-65 Table 22-5. Sound input device information selectors
- 22-83 Table 22-6. Sound commands
- 22-99 Table 22-7. The sound header format used by SetupSndHeader

- 22-21 Listing 22-1. A format 1 'snd' resource
- 22-21 Listing 22-2. A restructured format 1 'snd' resource
- 22-22 Listing 22-3. A format 1 'snd' resource containing sampled sound data
- 22-22 Listing 22-4. A resource specification
- 22-23 Listing 22-5. Resource specification for the Simple Beep
- 22-24 Listing 22-6. A format 2 'snd' resource
- 22-35 Listing 22-7. Playing an 'snd' resource with SndPlay
- 22-36 Listing 22-8. Using low-level Sound Manager routines
- 22-37 Listing 22-9. Creating a sound channel
- 22-41 Listing 22-10. Using the availableCmd command
- 22-41 Listing 22-11. Using the versionCmd command
- 22-42 Listing 22-12. Using the freqDurationCmd command
- 22-49 Listing 22-13. An 'snd' resource containing compressed sound data

- 22-51 Listing 22-14. Halving the frequency of a sampled sound
- 22-56 Listing 22-15. Compressing audio data
- 22-57 Listing 22-16. Playing an 'snd' resource from disk
- 22-58 Listing 22-17. Recording through the sound input dialog box
- 22-61 Listing 22-18. Recording directly from a sound input device
- 22-64 Listing 22-19. Determining the name of a sound input device
- 22-64 Listing 22-20. Determining some sound input device settings
- 22-71 Listing 22-21. Determining whether a sound channel is paused
- 22-72 Listing 22-22. Determining the number of allocated sound channels
- 22-74 Listing 22-23. Setting up double buffers
- 22-77 Listing 22-24. Defining a doubleback procedure
- 22-78 Listing 22-25. Issuing a callback command
- 22-79 Listing 22-26. Defining a callback procedure

23 The Time Manager

- 23-6 Figure 23-1. Original and revised Time Managers (drifting, unpredictable frequency)
- 23-7 Figure 23-2. The extended Time Manager (drift-free, fixed frequency)
- 23-11 Listing 23-1. Installing and activating a Time Manager task
- 23-13 Listing 23-2. Passing the address of the application's A5 world to a Time Manager task
- 23-13 Listing 23-3. Defining a Time Manager task that can manipulate global variables
- 23-14 Listing 23-4. Defining a periodic Time Manager task
- 23-15 Listing 23-5. Computing elapsed time

24 The Notification Manager

- 24-4 Figure 24-1. A notification in the Application menu
- 24-5 Figure 24-2. A sample alert box
- 24-8 Listing 24-1. Setting up a notification record

25 The File Manager

- 25-6 Figure 25-1. Identifying a file in MFS
- 25-8 Figure 25-2. Identifying a file in HFS
- 25-23 Figure 25-3. The effect of ioSearchBits on interpretation of ioSearchInfo1 and ioSearchInfo2
- 25-16 Table 25-1. How FSMakeFSSpec interprets file specifications
- 25-18 Table 25-2. The effect of FSpExchangeFiles on a catalog entry
- 25-18 Table 25-3. The effect of FSpExchangeFiles on a file control block
- 25-21 Table 25-4. Fields in ioSearchInfo1 and ioSearchInfo2 used for a file
- 25-21 Table 25-5. Fields in ioSearchInfo1 and ioSearchInfo2 used for a directory
- 25-14 Listing 25-1. Testing for PBCatSearch
- 25-15 Listing 25-2. Opening a document using the FSSpec record
- 25-19 Listing 25-3. Updating a file with FSpExchangeFiles
- 25-24 Listing 25-4. Searching a volume with PBCatSearch

26 The Standard File Package

- 26-6 Figure 26-1. The default Open dialog box
- 26-7 Figure 26-2. The default Save dialog box
- 26-7 Figure 26-3. The New Folder dialog box
- 26-7 Figure 26-4. The name conflict dialog box

- 26-8 Listing 26-1. Opening a document
- 26-10 Listing 26-2. The definition of the default Open dialog box
- 26-10 Listing 26-3. The definition of the default Save dialog box
- 26-11 Listing 26-4. The item list for the default Open dialog box
- 26-11 Listing 26-5. The item list for the default Save dialog box

27 The Alias Manager

- 27-5 Figure 27-1. Resolving a relative path

28 Memory Management

- 28-4 Figure 28-1. The memory organization in a single-application environment
- 28-5 Figure 28-2. The organization of the application partition
- 28-7 Figure 28-3. The memory organization in a multiple application environment
- 28-8 Figure 28-4. Using temporary memory
- 28-9 Figure 28-5. The Memory control panel
- 28-11 Figure 28-6. A master pointer structure in the 24-bit Memory Manager

- 28-14 Listing 28-1. Passing A5 to a notification response procedure
- 28-15 Listing 28-2. Setting up and restoring A5
- 28-22 Listing 28-3. Translating logical to physical addresses
- 28-38 Listing 28-4. Determining whether temporary memory routines are available

29 Process Management

- 29-11 Listing 29-1. Searching for a specific process
- 29-14 Listing 29-2. Launching an application

30 The Slot Manager

- 30-8 Table 30-1. Slot Manager search routines
- 30-18 Table 30-2. How the Slot Manager determines the base address used by an sResource data structure

- 30-7 Listing 30-1. Disabling and enabling sResource data structures
- 30-9 Listing 30-2. Searching for sResource data structures

31 The Power Manager

- 31-5 Figure 31-1. Relationship of an application to the Power Manager
- 31-9 Figure 31-2. How the sleep queue handles a sleep request
- 31-10 Figure 31-3. How the sleep queue handles a sleep demand

- 31-8 Table 31-1. Response of network services to sleep requests and demands

- 31-15 Listing 31-1. Adding an entry to the sleep queue
- 31-17 Listing 31-2. A sleep queue routine

32 The AppleTalk Manager

- 32-6 Figure 32-1. AppleTalk protocols
- 32-9 Figure 32-2. AppleTalk device drivers
- 32-10 Figure 32-3. AppleTalk application interfaces
- 32-26 Figure 32-4. AppleTalk Ethernet packet formats
- 32-82 Figure 32-5. An Ethernet write-data structure

- 32-27 Listing 32-1. Calling a LAP Manager L802.2 routine from assembly language
- 32-33 Listing 32-2. Using the GetZoneList function
- 32-34 Listing 32-3. Using the GetMyZone function
- 32-48 Listing 32-4. Using ADSP to establish and use a connection
- 32-54 Listing 32-5. Using ADSP to establish and use a connection listener
- 32-57 Listing 32-6. An ADSP user routine
- 32-80 Listing 32-7. Finding an EtherTalk card and opening the .ENET driver
- 32-82 Listing 32-8. Sending a data packet over Ethernet
- 32-84 Listing 32-9. Using the default Ethernet protocol handler to read data

A Result Codes

- A-1 Table A-1. Result codes

B Routines and Their Memory Behavior

- B-1 Table B-1. Routines that may move or purge memory
- B-5 Table B-2. Routines that do not move or purge memory but may not be called at interrupt time
- B-7 Table B-3. Routines that may be called at interrupt time

C System Traps

- C-1 Table C-1. System traps by trap name
- C-3 Table C-2. System traps by trap word
- C-5 Table C-3. System traps that take selectors
- C-16 Table C-4. Routines selected from system traps

D Global Variables

- D-1 Table D-1. Global variables

E The Standard Roman Character Set

- E-1 Table E-1. The standard Roman character set
- E-5 Table E-2. Unencoded PostScript characters

PREFACE

3	About <i>Inside Macintosh</i>
4	The Development Environment
5	The System Software Environment
5	The Format of a Typical Chapter
6	The Conventions Used in This Volume
7	Other Documentation
8	An Overview of the Chapters in Volume VI
8	Introduction to the System Software Version 7.0 Environment
8	User Interface Guidelines
8	Compatibility Guidelines
9	The Edition Manager
9	The Event Manager
9	The Apple Event Manager
9	The Program-to-Program Communications Toolbox
10	The Data Access Manager
10	The Finder Interface
10	Control Panels
10	The Help Manager
11	The Font Manager
11	The Resource Manager
11	Worldwide Software Overview
12	TextEdit
12	Graphics Overview
12	Color QuickDraw
12	The Picture Utilities Package
13	The Color Picker Package
13	The Palette Manager
13	The Graphics Devices Manager
13	The Sound Manager
13	The Time Manager
14	The Notification Manager
14	The File Manager
14	The Standard File Package
14	The Alias Manager
15	Memory Management
15	Process Management
15	The Slot Manager
15	The Power Manager
15	The AppleTalk Manager
16	A Road Map to Volume VI

ABOUT INSIDE MACINTOSH

Inside Macintosh is a six-volume set of books that describes how to write an application for the Apple® Macintosh® family of computers. *Inside Macintosh* is the definitive guide and reference for anyone writing software for the Macintosh computer. The first two volumes describe the routines in the Macintosh User Interface Toolbox and the Macintosh Operating System. The third volume is a summary of the Pascal interfaces for all routines described in Volumes I and II. The fourth and fifth volumes describe features and routines introduced with the Macintosh Plus, Macintosh SE, and Macintosh II computers. Volume VI describes the managers and features available in system software version 7.0.

Volume I contains the original user interface guidelines for Macintosh applications and an introduction to memory management and assembly language. It also describes QuickDraw™, the Resource Manager, the Event Manager, the Font Manager, the Window Manager, the Menu Manager, the Dialog Manager, TextEdit, and other routines relating to the user interface that you can use in your application.

Volume II describes the Macintosh Operating System, including the routines that perform file I/O, device I/O, memory management, and interrupt handling. It covers the File Manager, the Device Manager, the Printing Manager, the AppleTalk® Manager, and various drivers and utilities.

Volume III describes the Finder™ interface, provides an overview of the hardware of the Macintosh 128K and Macintosh 512K computers, and contains summaries of the Pascal interfaces for all routines described in Volumes I and II.

Volume IV describes routines introduced with the Macintosh Plus and Macintosh 512K enhanced computers. It introduces the Hierarchical File System, the SCSI Manager, the Time Manager, and the List Manager. The volume also describes changes to various managers and drivers and presents an overview of the Macintosh Plus hardware.

Volume V describes routines introduced with the Macintosh SE and Macintosh II computers. It describes Color QuickDraw, the Palette Manager, the Script Manager, the Sound Manager, the Slot Manager, the Apple Desktop Bus™, and changes made to various managers to support color. The volume also includes additional user interface guidelines and compatibility guidelines. It explains how to add color to menus, windows, and dialog boxes. It also discusses hierarchical, scrolling, and pop-up menus.

This volume, Volume VI, describes the system software version 7.0 environment, new managers available with version 7.0, new routines and data structures, new user interface guidelines, and how to take advantage of the version 7.0 environment.

Inside Macintosh, Volume VI, is also available in an on-line edition. The on-line edition provides a navigational model that lets you browse through information and it provides a search capability to quickly locate routines, data structures, and other text.

The *Inside Macintosh X-Ref* provides a comprehensive, integrated index for Volumes I through VI of *Inside Macintosh*, as well as *Programmer's Introduction to the Macintosh Family*; *Technical Introduction to the Macintosh Family*; *Designing Cards and Drivers for the Macintosh Family*, second edition; and *Guide to the Macintosh Family Hardware*, second edition. All these books are available from Addison-Wesley.

The Development Environment

The User Interface Toolbox and Macintosh Operating System routines are available using Pascal, C, or assembly-language interfaces. How you access these routines depends on the development environment you are using. This volume shows all routines in their Pascal interface using the Macintosh Programmer's Workshop (MPW[®]). All sample code listings are shown in MPW Pascal, with a few examples shown in assembly language.

The MPW development environment includes these books: *Macintosh Programmer's Workshop Development Environment, Volume 1*; *Macintosh Programmer's Workshop Development Environment, Volume 2*; *MPW Pascal: Macintosh Programmer's Workshop Pascal*; *MPW C: Macintosh Programmer's Workshop C*; and *MPW Assembler: Macintosh Programmer's Workshop Assembler*. These books are available from APDA[®] (Apple Programmers and Developers Association).

The code listings and other code in this volume were developed using MPW 3.0. They show methods of using various routines and illustrate techniques for accomplishing particular tasks. All code listings have been compiled and, in many cases, tested. However, Apple does not intend that you use these code samples in your application.

If you are programming in assembly language, pay attention to the assembly-language notes and trap macro notes. These notes provide information about saving and restoring registers, details of what each register must contain on entry to Operating System routines, what the routines return in the registers, and other information you might find helpful.

If you are programming in Pascal or C only, you can skip over the assembly-language information.

This volume occasionally uses *SurfWriter*, *WipeOut*, *store data*, *display data*, *send and receive*, *make memo*, and *spell quick* as names of sample programs for illustrative purposes; these are not actual products of Apple Computer, Inc.

APDA offers worldwide access to a broad range of programming products, resources, and information for anyone developing on Apple platforms. You'll find the most current versions of Apple and third-party development tools, debuggers, compilers, languages, and technical references for all Apple platforms. To establish an APDA account, obtain additional ordering information, or find out about site licensing and developer training programs, contact

APDA

Apple Computer, Inc.
20525 Mariani Avenue, M/S 33-G
Cupertino, CA 95014-6299

Telephone: 800-282-2732 (United States)
800-637-0029 (Canada)
408-562-3910 (elsewhere in the world)
Fax: 408-562-3971
Telex: 171-576

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

For information on registering signatures, file types, Apple events, and other technical information, contact

Macintosh Developer Technical Support
 Apple Computer, Inc.
 20525 Mariani Ave., M/S 75-3T
 Cupertino, CA 95014-6299

The System Software Environment

Inside Macintosh Volume VI focuses on system software version 7.0; however, many of the chapters in this volume contain information that is also relevant to system software version 6.0 and later. See the Compatibility Guidelines chapter for information on developing applications that can run in both system software version 6.0 and system software version 7.0.

If the Gestalt function is available, you should use it instead of the SysEnviron and Environ routines. You can use the Gestalt function to determine whether all the features your application requires are present on a particular Macintosh computer. You should not rely on the ROM version, since later system software versions can override routines in ROM. See the Compatibility Guidelines chapter for details on how to use the Gestalt function.

The Format of a Typical Chapter

Almost all chapters in Volume VI have a standard structure. For example, the Edition Manager chapter contains these sections:

- “About This Chapter” This section describes the information you can find in the chapter and includes references to related chapters.
- “About the Edition Manager” This section provides an overview of the features provided by the Edition Manager.
- Additional sections describe concepts related to the Edition Manager.
- “Using the Edition Manager” This section describes the tasks you can accomplish using the routines provided by the Edition Manager. It describes how to use the most common routines, gives related user interface information, provides code samples, and supplies additional information.
- “Edition Manager Routines” This section lists Edition Manager routines in version 7.0, with routine declarations and descriptions of every parameter for each routine.
- “Summary of the Edition Manager” This section provides the Edition Manager’s Pascal interface for version 7.0 constants, data structures, routines, and procedures, as well as relevant assembly-language information.

The Conventions Used in This Volume

This volume uses elements such as assembly-language notes, trap macro notes, note boxes, and warning boxes to set off important information. Trap macro notes and assembly-language notes are useful only if you are programming in assembly language.

All routines (with a few exceptions) have both a Pascal and assembly-language form. The summary at the end of each chapter first lists the constants, data structures, and routines provided with the MPW Pascal interface files, and then lists equivalent assembly-language information for data structures and routines for use with the MPW Assembler interface files. The constants for the MPW Assembler interface files are the same as their Pascal equivalents, so the constant names are shown only in the Pascal section of the summary. (The constants, data structure names, and routine names in the MPW C interface files are also the same as their Pascal equivalents.)

When appropriate, the declaration for a procedure or function includes relevant assembly-language information in the form of a trap macro note that immediately follows the declaration. The trap macro that corresponds to a Pascal interface routine begins with an underscore character (`_`) followed by the Pascal routine name. Trap macro notes appear in this form:

Trap macro	For register-based routines, this shows the trap macro name and describes the parameters that must be in the registers on entry to the routine and describes the values returned in the registers.
	For stack-based routines, this shows the name of the trap macro if it is different from the Pascal interface name.

Assembly-language notes appear in this form:

Assembly-language note: This gives information of interest only if you are programming in assembly language.

If you are programming in Pascal or C only, you can skip over the information in trap macro notes and assembly-language notes.

Important information is often called out in a note box:

Note: Text set off in this way presents reminders or notes related to the topic.

Information that you need to pay special attention to is shown in a warning box:

▲ **Warning:** Warnings like this alert you to situations in which you could damage software or lose data. ▲

Words that appear in **boldface** are key terms or concepts and are defined in the Glossary

All code listings use the Courier font (*this is Courier*) to indicate code from a sample program that can be compiled. The summary listings and set-off code in text also use Courier for the actual data structure names, field names, constant names, and routine names that match the names used in the MPW Pascal interface files.

Many Toolbox and Operating System routines accept a pointer to a parameter block as a parameter. For these routines, the routine description includes a list of the fields in the parameter block that are used by the routine.

A typical parameter block description looks like this:

Parameter block

[in/out]	[offset]	[field name]	[size]	[description]
→	0	input1	long	This is an input parameter
←	4	output1	word	This is an output parameter
↔	6	inAndOut	long	This is an input/output parameter
→	10	reqCount	long	Requested number of files to send
→	14	buffer	long	Pointer to data buffer
←	18	accCount	long	Actual number of files sent

The arrow in the first column indicates whether the field is an input parameter, output parameter, or both. You must supply values for all input parameters and input/output parameters. The routine returns values in output parameters and input/output parameters.

The second column indicates the offset and is useful only if you are programming in assembly language or debugging your code. The offset value is the offset in bytes from the beginning of the parameter block for each field within the structure.

The third column shows the field name as defined in the MPW Pascal interfaces, and the fourth column shows the size of that field. The size is given in bytes or indicated as *word* or *long* (for long word). Long indicates a field that occupies 4 bytes; word indicates a field that occupies 2 bytes. The size is provided for your information and is more useful if you are programming in assembly language. The final column provides a short description of the field.

Other Documentation

For specific hardware information about the Macintosh family, see *Guide to the Macintosh Family Hardware*, second edition, and *Designing Cards and Drivers for the Macintosh Family*, second edition; for additional software information, see previous volumes of *Inside Macintosh*. Also see *Macintosh Worldwide Development: Guide to System Software* for a complete description of all components of the worldwide system software. See *Human Interface Guidelines: The Apple Desktop Interface* for a complete description of the Apple human interface.

AN OVERVIEW OF THE CHAPTERS IN VOLUME VI

The following sections describe the content of each chapter in this volume and tell where to find additional information in previous volumes. Figure P-1 (at the end of the Preface) lists the chapters in Volume VI and shows which other volumes cover those topics.

Introduction to the System Software Version 7.0 Environment

The first chapter in this volume provides an overview of the features of system software version 7.0. It describes the operating environment for applications that run in version 7.0.

User Interface Guidelines

The User Interface Guidelines chapter in Volume VI reviews the user interface design principles and gives new guidelines for system software version 7.0. The chapter discusses windows, dialog boxes and movable modal dialog boxes, additions to the standard menus, terminology, and user feedback. It also gives guidelines for developing worldwide software and for designing color icons and windows.

The Finder Interface chapter in this volume provides related information on the user interface presented by the Finder. Individual chapters address specific issues related to the user interface features provided by a particular manager.

The User Interface Guidelines chapter in Volume I describes the various components of a Macintosh application and discusses the use of menus, windows, dialog boxes, scroll bars and other controls.

The User Interface Guidelines chapter in Volume IV discusses use of the arrow keys, reserved keyboard equivalents, window zooming, and the standard close box.

The User Interface Guidelines chapter in Volume V briefly discusses the use of color in your application. The chapter describes features of the standard and extended keyboards, and discusses using sound, hierarchical menus, and scrolling menus in your application.

For more information on the Apple human interface, see the *Human Interface Guidelines: The Apple Desktop Interface*.

Compatibility Guidelines

The Compatibility Guidelines chapter describes issues relating to compatibility for various managers in system software version 7.0. It also includes details on pop-up menus, movable modal dialog boxes, new routines for manipulating dialog items in a dialog box, and discusses menu access when an application displays a modal dialog box.

The chapter also shows you how to call Gestalt, the new function for determining various attributes, versions, and features of the system software.

The chapter gives guidelines you should follow to help ensure that your application is compatible across the Macintosh family of computers. It also provides information on how to make your application compatible with A/UX[®] (Apple's version of the UNIX[®] operating system) and presents a brief overview of how to write software that can be easily localized for use in other regions.

The Edition Manager

The Edition Manager chapter describes how you can let users publish and subscribe data among many documents. The Edition Manager is part of the interapplication communications (IAC) architecture in version 7.0. See the Edition Manager chapter for sample code that shows how to add publish and subscribe capabilities to your application.

The Event Manager

The Event Manager chapter in Volume VI includes information on all events, including suspend and resume events. The chapter incorporates information from *Programmer's Guide to MultiFinder* and replaces the information found there. The Event Manager chapter in this volume also describes how to send and receive high-level events.

For specific information on keyboard events, the modifier flags field of the event record, reading the keyboard and keypad, and responding to mouse events or disk-inserted events, see the Toolbox Event Manager chapter in Volume I.

You also may want to read about the Operating System Event Manager, described in Volume II. The Operating System Event Manager handles low-level, hardware-related events. The Operating System Event Manager chapter also describes how your application can post its own events in the event queue. You usually use the Event Manager to send and retrieve events. For information on the PPostEvent function, see the Operating System Event Manager chapter in Volume IV.

For information on standard keyboards, an addition to the modifier flags field in the event record, and the KeyTrans function, see the Toolbox Event Manager chapter in Volume V.

The Apple Event Manager

The Apple Event Manager chapter describes Apple events and how your application can receive and process the required set of Apple events. It also describes how to create and send Apple events.

The Program-to-Program Communications Toolbox

The Program-to-Program Communications (PPC) Toolbox chapter describes how your application can exchange message blocks with other applications. The PPC Toolbox provides low-level control of communication and is generally more suitable for code that is not event-based or desk accessories or applications that are closely integrated.

The Data Access Manager

The Data Access Manager chapter describes how your application can communicate with a database application or other data source running on a remote computer. The chapter describes how your application can use high-level or low-level routines to initiate communication with a remote data server, send commands or data to the server, and, after the server executes the commands, retrieve any requested data from the server.

The Finder Interface

The Finder Interface chapter in this volume describes how to create bundles, file references, and icons, including small icons and color icons. Code listings show how to set up the resources the Finder needs to start up your application and display your application's icons on the desktop.

The chapter also describes changes to the Finder interface—for example, the new aliases and stationery documents. It shows how to find special folders, such as the Preferences folder and Temporary Items folder. In addition, the chapter describes how fonts and sounds are visible on the desktop and how the user installs fonts and sounds by moving their icons to the System Folder icon.

The Finder Interface chapter describes the Desktop Manager, a new manager that lets your application add or remove information from the desktop database.

The Finder Interface chapter in this volume replaces the Finder Interface chapters in Volumes III and IV.

Control Panels

The Control Panels chapter in this volume describes the new behavior of control panels in system software version 7.0. If you develop video cards, you can also use the information in the chapter to create an Options dialog box for the Monitors control panel.

The Control Panel chapter in Volume V describes how to write a control panel. Read the information in the Control Panels chapter in this volume for additional information on writing a control panel in system software version 7.0. Control panels written for earlier versions of system software are compatible with version 7.0.

The Help Manager

The Help Manager chapter discusses how you can provide help balloons that supply your users with information that describes the actions, behaviors, or properties of elements of your application. The chapter explains how to create help balloons for menus, windows, icons, controls, and other elements of the user interface of your application.

The Font Manager

The Font Manager chapter in Volume VI describes how your application can take advantage of TrueType™ fonts.

The Font Manager chapter in Volume I describes how the Font Manager works with QuickDraw to draw characters. It discusses font numbers, character styles, font size, scaling factors, the ascent line, the base line, the descent line, and leading. The chapter also describes the format of a bitmapped font.

The Font Manager chapter in Volume IV discusses bitmapped fonts (of resource type 'FONT' or 'NFNT') and font families (of resource type 'FOND'). It describes a few data structures, like the font family record.

The Font Manager chapter in Volume V includes information on fractional character widths, the font search algorithm (how the Font Manager looks for a particular font), and how to specify colors for a font.

The Resource Manager

The Resource Manager chapter in Volume VI lists the standard resource types in version 7.0. The chapter also describes routines that you can use to read or write part of a resource.

The Resource Manager chapter in Volume I describes how you can store menus, fonts, icons, and other data as resources. It gives definitions and descriptions of resource files, resource forks, and data forks. It describes how to create and open resource files, how to read resources from a resource file, and how to add, remove, update, and write resources to a resource file.

The Resource Manager chapter in Volume IV describes a few routines that search only the current resource file (these routines have the numeral 1 in their routine name). It also describes two advanced functions, RsrcMapEntry and OpenRFPern.

The Resource Manager chapter in Volume V describes the RGetResource function and lists resource types, ROM resources, and resources in the System file.

Worldwide Software Overview

The Worldwide Software Overview chapter provides an introduction to scripts and script systems. It can help you design your application so that it is compatible with Macintosh computers throughout the world.

See the Worldwide Software Overview chapter for an introduction to worldwide issues, and see the User Interface Guidelines chapter for guidelines about developing your application for use around the world. See the International Utilities Package chapter in Volume I for information on displaying numbers, currency, time, and dates in the correct format for various countries around the world. *Macintosh Worldwide Development: Guide to System Software* (available from APDA) replaces the Script Manager chapter in Volume V and provides a more complete description of all components of the worldwide system software.

TextEdit

The TextEdit chapter in this volume describes how TextEdit provides support for working with different script systems. *It describes how you can use TextEdit to let the user edit and display text in multiple scripts and styles when a non-Roman script system is in use. TextEdit automatically handles text that uses more than one script, style, or direction.*

The TextEdit chapter in Volume I introduces TextEdit and explains how your application can use TextEdit routines for basic text formatting and editing.

The TextEdit chapter in Volume IV describes how TextEdit supports automatic scrolling of text.

The TextEdit chapter in Volume V explains how TextEdit lets you vary text attributes such as size, style, and font. It also describes the style record that stores the style information.

Graphics Overview

The Graphics Overview chapter provides an introduction to graphics on the Macintosh computer. The system software provides a rich set of routines that support quick drawing of objects such as circles, rectangles, and text. The Graphics Overview chapter introduces many of the concepts and data structures explained in greater detail in the chapters on Color QuickDraw, the Picture Utilities Package, the Color Picker Package, the Palette Manager, and the Graphics Devices Manager.

Color QuickDraw

The Color QuickDraw chapter in Volume VI describes how version 7.0 supports both indexed and direct specification of color. It also describes changes to the pixel map record and the PICT2 file format, and it describes a routine that lets you convert a bitmap record into a region. The information in the Color QuickDraw chapter in this volume supplements the QuickDraw chapter in Volume I and the Color QuickDraw chapter in Volume V.

The QuickDraw chapter in Volume I introduces the basic concepts of QuickDraw, including descriptions of the mathematical foundation of QuickDraw and the graphics environment that QuickDraw provides. It also describes QuickDraw routines.

The Color QuickDraw chapter in Volume V describes how Color QuickDraw provides support for drawing objects using a large number of different colors.

The Picture Utilities Package

The Picture Utilities Package chapter describes routines you can use to examine the contents of pictures and pixel maps.

The Color Picker Package

The Color Picker Package chapter in this volume describes how to present users with a standard user interface for selecting a color. This chapter replaces the Color Picker Package chapter in Volume V.

The Palette Manager

The Palette Manager chapter in this volume describes palettes, the default color tables, and how to create and use a palette to control the color environment. This chapter replaces the Palette Manager chapter in Volume V.

The Graphics Devices Manager

The Graphics Devices Manager chapter describes how you can prepare offscreen graphics and move them quickly into view. It also provides useful information if you are developing a graphics-intensive application. This chapter replaces the Graphics Devices chapter in Volume V.

The Sound Manager

The Sound Manager chapter in this volume completely replaces any previous information in *Inside Macintosh* regarding the Sound Manager. The Sound Manager chapter in Volume VI is the complete reference and guide for the use of sound. It provides an introduction to sound and describes sound synthesizers, sound channels, sound commands, sound resources, and sound files.

The chapter also describes how your application can use the Sound Manager to create and play sounds, mix and synchronize multiple channels of sound, expand and compress sound data, and play sounds continuously from disk.

See the Sound Manager chapter in this volume if you want to use any kind of sound in your application, even if you only want to use the SysBeep procedure.

The Time Manager

The Time Manager chapter in this volume describes the original Time Manager, the revised Time Manager (available in system software version 6.0.3 and later), and the extended Time Manager (available in system software version 7.0). It completely replaces the Time Manager chapter in Volume IV.

The chapter describes how to schedule a routine for later execution, how to schedule a routine to execute at periodic intervals, and how to compute elapsed time. It also describes other time-related services, such as those provided by the TickCount and Delay functions, and the Vertical Retrace Manager.

The Notification Manager

The Notification Manager chapter describes how to notify users of significant occurrences relating to your application when your application is running in the background. Device drivers, VBL tasks, Time Manager tasks, completion routines, startup code, desk accessories, and applications can use the Notification Manager.

The Notification Manager chapter in this volume replaces the information in Appendix D of the *Programmer's Guide to MultiFinder*.

The File Manager

The File Manager chapter in this volume describes how to create a file specification to identify a file, folder, or volume. It also describes how you can use the File Manager to search for and quickly find files.

The File Manager chapter in Volume IV describes the file system, including the Macintosh File System (MFS) and Hierarchical File System (HFS). The chapter provides descriptions of File Manager data structures and routines.

The chapter on File Manager Extensions in a Shared Environment in Volume V presents routines that allow your application to more easily execute in a shared environment.

The Standard File Package

The Standard File Package chapter in this volume describes the StandardGetFile and StandardPutFile procedures available in version 7.0. You can use these two procedures to present the standard user interface when a user opens or saves a file. The chapter also describes the two new procedures CustomGetFile and CustomPutFile, which let your application exercise more control over the user interface when opening and saving files.

The Standard File Package chapter in Volume I describes the original procedures that present the standard user interface for opening and saving files in earlier system software. The Standard File Package chapter in Volume IV describes modifications to the original procedures for use with the Hierarchical File System.

The Alias Manager

The Alias Manager chapter describes how to create and resolve alias records—a new data structure that describes a file, folder, or volume.

You can use alias records instead of conventional file specifications to store file or directory information. If you create an alias record, your application can use the Alias Manager to locate the file or directory when needed—even if the user has renamed it, copied it, restored it from backup, or moved it. The chapter describes the routines you can use to

Memory Management

The Memory Management chapter in Volume VI describes 32-bit addressing, virtual memory, and routines that let your application use available temporary memory. The chapter replaces the discussion of temporary memory in Chapter 3 of the *Programmer's Guide to MultiFinder*.

The Memory Manager chapter in Volume II describes the system heap zone and application heap zone, how to allocate memory blocks, and how to avoid memory fragmentation. It also discusses dereferencing a handle, lists general-purpose data types, shows the organization of memory, and gives an overview of the stack and the heap. The routine descriptions discuss how to set the heap zone size, create handles and pointers, allocate relocatable and nonrelocatable blocks, and how to free memory in the heap.

The Memory Manager chapter in Volume IV describes improvements to Memory Manager routines that are largely transparent to your application. It also describes routines that let your application set or clear flags that the Memory Manager associates with each relocatable block.

Process Management

The Process Management chapter describes how the Process Manager schedules applications for execution and manages access to shared resources. It describes routines that let your application get information about any or all running applications. The chapter replaces the discussion of launching applications found in the *Programmer's Guide to MultiFinder*.

The Slot Manager

The Slot Manager chapter in this volume describes how version 7.0 supports 32-bit addressing of NuBus™ cards. The Slot Manager chapter in Volume V gives an overview of the firmware of a slot card, explains the slot parameter block, and describes Slot Manager routines.

The Power Manager

The Power Manager chapter describes a manager used only with the Macintosh Portable in system software version 6.0.4 and later. This information is useful only if you are writing a device driver or application that might be affected when power for the various subsystems of the Macintosh Portable is shut off.

The AppleTalk Manager

The AppleTalk Manager chapter in this volume describes how version 7.0 supports various link access protocols (for example, the LocalTalk® Link Access Protocol and the EtherTalk® Link Access Protocol) that can be used for AppleTalk communication. It also describes the AppleTalk Data Stream Protocol (ADSP), a new protocol your application can use to exchange information between two equal entities.

The chapter explains how you can request that your program receive notification each time another routine opens or closes the .MPP driver or whenever another routine is about to close the .MPP driver.

The chapter also discusses how the LAP Manager lets your application control communication over non-LocalTalk networks, such as Ethernet. In addition, it provides information you can use to write your own protocol handler for Ethernet or 802.3.

The AppleTalk Manager chapters in Volumes II, IV, and V provide additional information on the device drivers and protocols associated with AppleTalk.

A ROAD MAP TO VOLUME VI

Figure P-1 shows each chapter in this volume. If you need to read related chapters in earlier volumes of *Inside Macintosh* for additional information, those other volumes are also shown. For each chapter, the volumes are shown in the order in which you should read them; the volumes shown are the only ones you need to read for information on that topic.

Volume VI Chapter	Volume Priority	Volume VI Chapter	Volume Priority
Introduction to the System Software Version 7.0 Environment	VI	Color QuickDraw	V VI
User Interface Guidelines	I IV V VI	Picture Utilities Package	VI
Compatibility Guidelines	VI	Color Picker Package	VI
Edition Manager	VI	Palette Manager	VI
Event Manager	VI I IV V	Graphics Devices Manager	VI
Apple Event Manager	VI	Sound Manager	VI
Program-to-Program Communications Toolbox	VI	Time Manager	VI
Data Access Manager	VI	Notification Manager	VI
Finder Interface	VI	File Manager	IV V VI
Control Panels	V VI	Standard File Package	I IV VI
Help Manager	VI	Alias Manager	VI
Font Manager	I IV V VI	Memory Management	II IV VI
Resource Manager	I IV V VI	Process Management	VI
Worldwide Software Overview	VI	Slot Manager	V VI
TextEdit	I IV V VI	Power Manager	VI
Graphics Overview	VI	AppleTalk Manager	II IV V VI

Preface

Figure P-1. A road map to Volume VI

1 INTRODUCTION TO THE SYSTEM SOFTWARE VERSION 7.0 ENVIRONMENT

- 3 About This Chapter
- 4 About the System Software Version 7.0 Environment
- 6 The Cooperative Multitasking Environment
- 7 Interapplication Communication
 - 10 Sharing Data Among Applications
 - 12 Sending Events Between Applications
 - 13 Exchanging Message Blocks Between Programs
- 14 Remote Data Access
- 15 Enhanced User Interface
- 16 Sound
- 16 TrueType Fonts
- 16 Graphics
- 18 File Management
- 19 Memory Management
 - 19 Temporary Memory
 - 20 24-Bit and 32-Bit Addressing
- 20 Process Management
- 21 Timing Services
- 21 Compatibility
- 21 Worldwide Development
- 23 Communication Over a Network
- 23 Hardware Interfaces
- 24 Overview of Chapters in This Volume

ABOUT THIS CHAPTER

This chapter describes the operating environment for applications that run in system software version 7.0. It also provides general information about the features available to you when you design an application to run in the system software version 7.0 environment.

Read this chapter for an overview of how your application can use the Macintosh® User Interface Toolbox and Macintosh Operating System routines in system software version 7.0 to

- share data with other applications using the Edition Manager
- communicate with other applications using the Event Manager, Apple® Event Manager, or the Program-to-Program Communications (PPC) Toolbox
- access data from other sources, including remote databases, using the Data Access Manager
- play sounds using the Sound Manager
- keep track of specific files using the Alias Manager
- perform quick searches for specific files using the File Manager
- provide on-line assistance for users with the Help Manager
- draw TrueType™ fonts using the Font Manager
- use direct devices for graphics applications using Color QuickDraw™
- function in worldwide markets using the Script Manager, International Utilities Package, and TextEdit

This chapter discusses the features and managers new to version 7.0. In addition, see the Preface, where “A Road Map to Volume VI” shows each manager discussed in this volume and illustrates a pathway through related information in previous volumes of *Inside Macintosh*.

Although Volume VI focuses on system software version 7.0, many of its chapters contain information that is also relevant to system software version 6.0 and later. See the Compatibility Guidelines chapter in this volume for information on developing applications that can run in both system software version 6.0 and system software version 7.0.

ABOUT THE SYSTEM SOFTWARE VERSION 7.0 ENVIRONMENT

System software version 7.0 extends the environment of the Macintosh computer by providing even greater support for cooperation between applications. The user interface continues to build on solid design principles and provides additional benefits; for example, in version 7.0 users can more directly manipulate icons on the desktop and users can customize the Apple menu. The Finder™, the Macintosh Operating System, and the User Interface Toolbox provide and maintain this environment.

The Finder is the system application that lets users organize and manage applications, documents, folders, and disks on the desktop. Users can choose commands from the Finder menu bar or use the mouse to perform various tasks. Because the Finder presents the standard interface that the user becomes familiar with, you need to make sure that your application performs in an expected manner in the Finder environment.

Macintosh users also expect certain standard behavior from Macintosh applications; for example, all applications should provide File and Edit menus. Macintosh applications that follow the user interface guidelines provide consistency and let users determine what action to take to perform a particular task.

In earlier Macintosh computers a user ran one application at a time. Today's Macintosh model recognizes that a user often wants to run many applications at once. System software version 7.0 provides this cooperative environment.

In system software versions 5.0 and 6.0, the MultiFinder® option provided a cooperative multitasking environment. In system software version 7.0, the features of MultiFinder are integrated into the Macintosh Operating System.

The Macintosh Operating System lets the user have several applications open at the same time and lets the user switch between them. The Operating System also gives the user constant access to the Finder. This lets a user move among open documents and applications without having to save or quit the previous document or application. This environment also allows applications to run in the background. For example, the Finder can copy files while the user is working on another task in the foreground.

The cooperative environment of the Macintosh allows multiple applications to share the CPU and other resources. You need to understand how this environment can affect your application. The next section, "The Cooperative Multitasking Environment," explains this in more detail.

An important aspect of system software version 7.0 is **interapplication communication (IAC)**, a new collection of features that help applications work together.

Copy and paste is a simple way in which Macintosh applications work together by sharing data. In system software version 7.0, applications can provide automated copy and paste features (that is, your application can automatically update the data that the user pastes into a document when the original source of information changes). Applications can extend this concept by using high-level events to request that other applications perform a particular task or return requested information. Applications and drivers that require close integration with each other can also extend this concept by reading and writing low-level message blocks.

Apple Computer, Inc. has defined a protocol for high-level events called the Apple Event Interprocess Messaging Protocol. High-level events that adhere to this protocol are called Apple events. You can help ensure effective communication with other applications by using this protocol.

Macintosh applications in system software version 7.0 can respond to incoming high-level events from other applications as well as events generated by the user, and they can also send high-level events to other applications. Better cooperation and communication between applications help users to get the most out of any one application or to use the best features from many applications—in effect, combining the features of many applications to achieve the desired result.

By including the features provided by IAC in your application, you give the users of your application even greater power, ease of use, and flexibility in accomplishing their tasks.

Figure 1-1 highlights the general areas for which system software version 7.0 provides routines. The next sections describe these topics in greater detail.

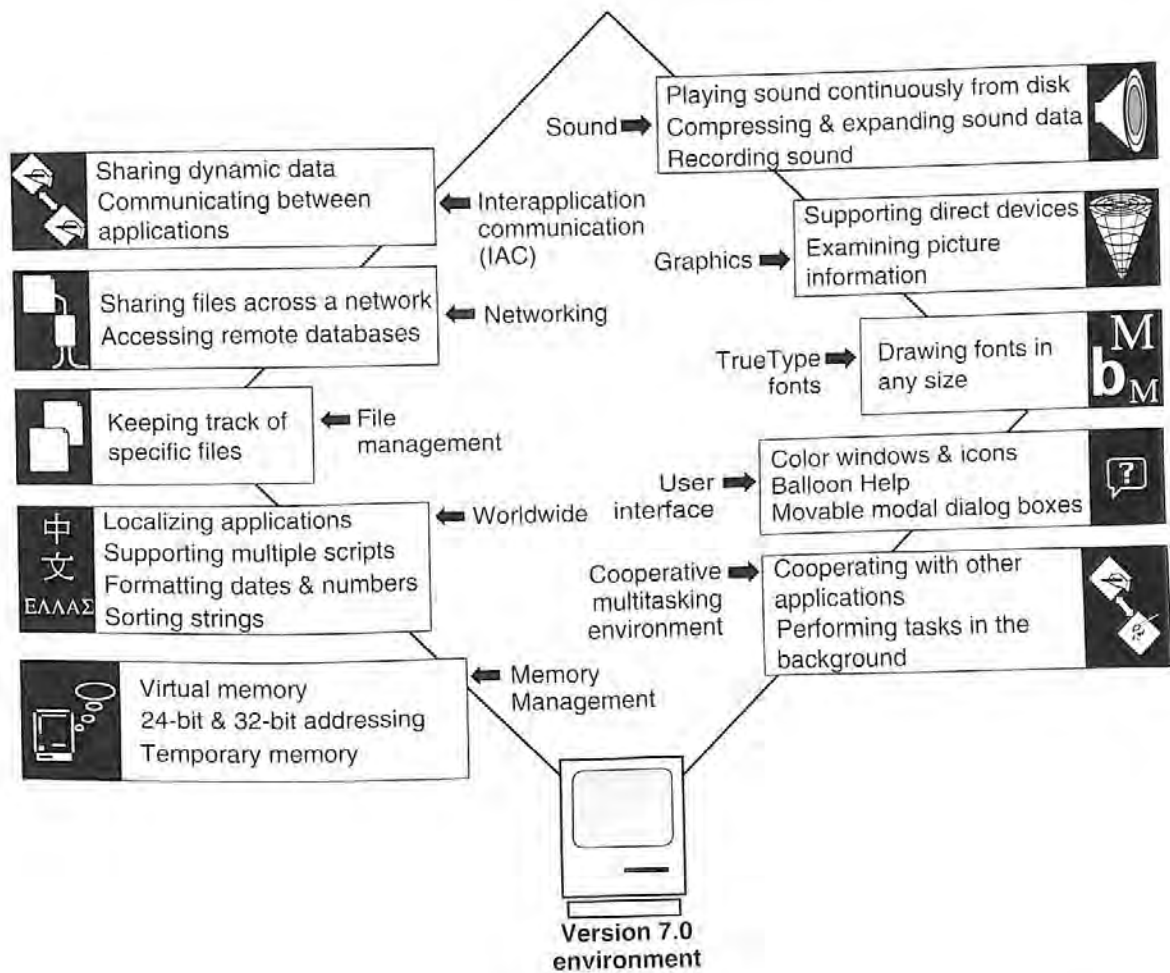


Figure 1-1. Features of the system software version 7.0 environment

The Cooperative Multitasking Environment

The cooperative multitasking environment is a standard part of system software version 7.0. The Macintosh Operating System and the Finder work together to provide this environment. MultiFinder is now transparent to the user; the user always has the capability to run more than one application at a time. Because the user may choose to run other applications in addition to your application, your application needs to be capable of existing in a shared environment.

The Operating System schedules the processing of all applications and desk accessories. When a user opens a document or application, the Operating System loads the application code into memory and schedules the application to run. The application runs at the next available opportunity. The next available opportunity usually means when the current process or application gives up the CPU. In most cases, the application runs immediately (or appears to the user to run immediately).

Once an application is executing, the CPU is available only to that application. The application can only be interrupted by hardware interrupts, and these are transparent to the application. However, to allow the user to interact with your application and others, you must periodically relinquish the CPU using the `WaitNextEvent` or `EventAvail` function. Using these event routines in your application lets the user interact with your application and also with other applications.

Although the user can have a number of open documents and applications, only one application is the active application. The **active application** is the application currently interacting with the user; its icon appears in the right side of the menu bar. The active application displays its menu bar and is responsible for highlighting the controls of its frontmost window.

When your application is the active application and the user switches to another application (by clicking in the window of a document belonging to another application, for example), the Operating System sends your application a suspend event. When your application receives a suspend event, it should prepare to suspend processing, allowing the user to switch to the other application. For example, in response to a suspend event, your application should remove the highlighting from the controls of its frontmost window and take any other necessary actions. The suspension actually occurs the next time your application calls `WaitNextEvent` or `EventAvail`.

Your application also needs to be able to resume processing when the user chooses to work with your application again. Your application receives a resume event when the user switches back to your application. In response to a resume event, your application should update the contents of its windows and highlight the controls of its frontmost window.

The Operating System preserves the environment of your application when it is suspended and restores that environment before sending it a resume event. Your application does not need to preserve or restore the operating environment in response to suspend or resume events.

When you perform user testing of your application, you might want to observe people using other applications as well as your application, to make sure that your application works well in a cooperative environment.

See the Compatibility Guidelines and the Event Manager chapters in this volume for specific information on how your application can handle suspend and resume events and how your application can take advantage of the cooperative multitasking environment.

Interapplication Communication

The interapplication communications architecture provides support for

- automated copy and paste between applications
- sending and receiving events between applications
- reading and writing blocks of data between applications

The Edition Manager, Apple Event Manager, Event Manager, and PPC Toolbox provide these features, and Figure 1-2 shows their relationships.

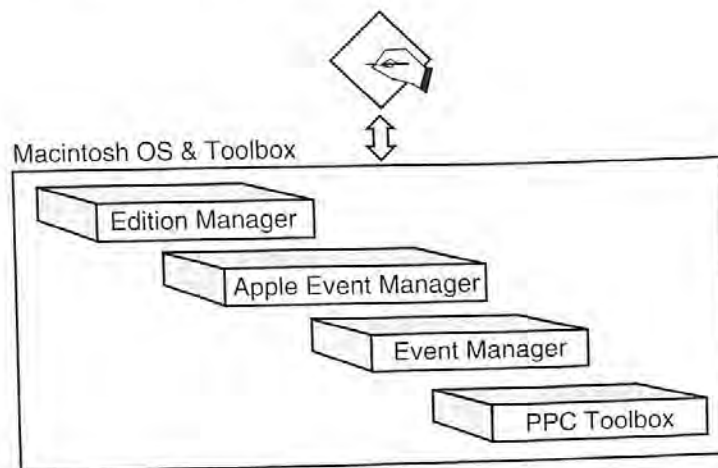


Figure 1-2. The managers constituting the interapplication communications architecture

The IAC architecture is built on communication and cooperation between applications. Apple has defined important standards to help ensure that communication between applications is effective. Using the Clipboard, applications can share static data by allowing the user to copy and paste data between documents. Using the Edition Manager, applications can support dynamic data sharing and allow users to perform automatic copy and paste between documents. Applications that support dynamic data sharing allow users to copy data from one document to another and receive automatic updating of the information when the data in the original document changes. The verbs *publish* and *subscribe* describe this form of dynamic data sharing.

You can let users publish and subscribe among many documents by using the Edition Manager and implementing the Create Publisher and Subscribe To menu commands. This is a form of high-level communication between applications; actually, the communication is indirect, as the Edition Manager provides the interface that allows applications to share dynamic data.

Your application can publish and subscribe with applications and documents on a local disk or across a network. In general, anything that you allow the user to copy or paste you should also allow the user to publish or subscribe to. See “Sharing Data Among Applications” later in this chapter for more information on using the publish and subscribe features in your application.

Using the Apple Event Manager, applications can send Apple events to each other to request services or information. These types of events are often the result of a user request, or they can be specific events that your application sends to another application. Apple events provide a standard way in which your application can communicate with many other applications. Other high-level events are for applications that choose to use a protocol other than the Apple Event Interprocess Messaging Protocol (AEIMP). Applications can use the Event Manager to send high-level events that follow their own protocol.

The Program-to-Program Communications (PPC) Toolbox is a set of low-level routines that allow applications to communicate on the local computer or over a network. Using the PPC Toolbox, applications can exchange blocks of data with each other by reading and writing low-level message blocks. The PPC Toolbox provides a method of communication between applications that is more useful for applications that are closely integrated, specifically designed to work together, or dependent on each other for information. The PPC Toolbox is typically more useful for code that is not event-based.

Your application can use the PPCBrowser function to allow the user to choose another application to which to send high-level events or low-level message blocks. The PPCBrowser function provides a standard user interface for choosing an application to communicate with, much like the Standard File Package provides a standard user interface for opening a file.

All these forms of interapplication communication are based on the premise that applications cooperate with each other. Both the application sending the high-level event or low-level message block and the application receiving it must agree on the protocol of communication.

Figure 1-3 shows that your application can use the Edition Manager to publish and subscribe data. Your application can use the Apple Event Manager to send and process Apple events and the Event Manager to send and receive high-level events. Your application can use the PPC Toolbox to read and write low-level message blocks. Your application can use any of these methods to communicate with other applications located on the same computer or across a network.

As Figure 1-3 shows, managers in the IAC architecture can use the services of other managers. For example, the Apple Event Manager uses the communication services of the Event Manager. The Event Manager in turn uses the PPC Toolbox on behalf of applications.

Figure 1-4 shows how two different applications can use the Edition Manager to publish and subscribe, and how they can use the routines provided by the Apple Event Manager, the Event Manager, or the PPC Toolbox to communicate with each other.

The next sections describe the three parts of the IAC architecture: the Edition Manager, the Apple Event Manager and Event Manager, and the PPC Toolbox.

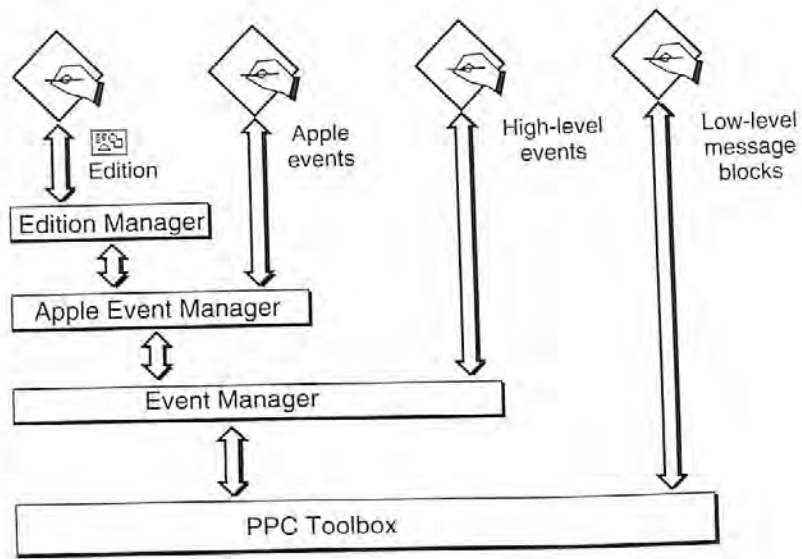


Figure 1-3. Using interapplication communication

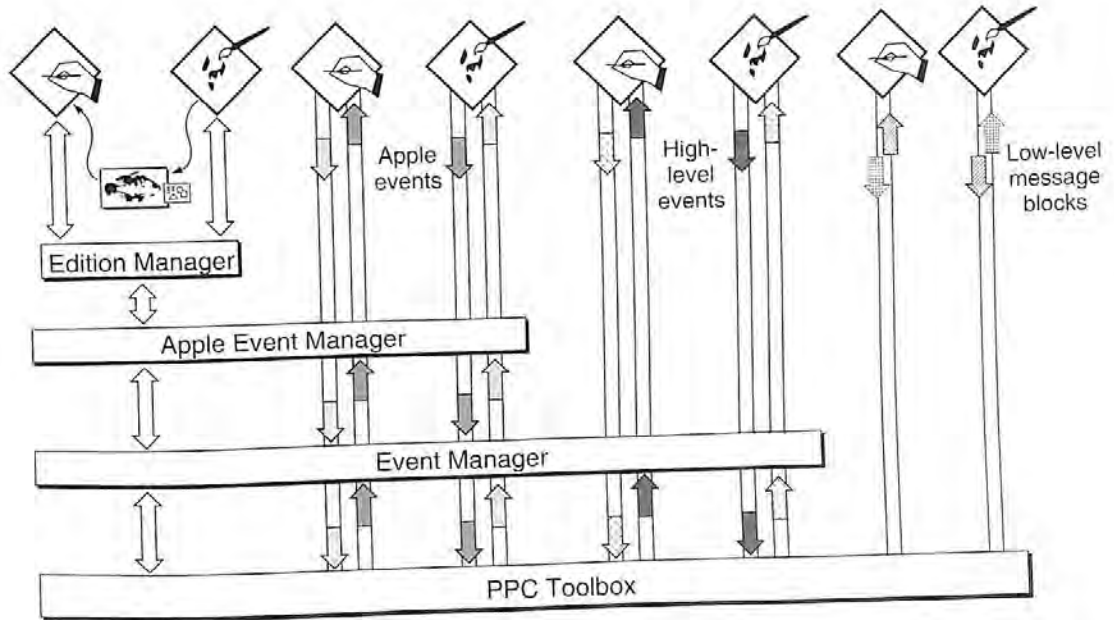


Figure 1-4. Applications using interapplication communication

Sharing Data Among Applications

The Edition Manager lets applications share dynamic data at the user's request. (The Clipboard lets applications share static data.) You build publish and subscribe capabilities into your application in much the same way that you build copy and paste into your application.

Using the Edition Manager, you can let a user publish data by selecting a portion of text, graphics, or other data within a document and choosing Create Publisher from the Edit menu. When the user performs this action, your application saves the selected information in a separate file. The information that is stored in a separate file is referred to as an **edition**. You can also let a user subscribe to data in an edition by choosing Subscribe To from the Edit menu; when the user chooses an edition, your application includes the information from the edition in the current document. The information in an edition can be shared by many documents.

A **publisher** is a portion of a document that is made available to other documents through an edition. A **subscriber** is a portion of a document that receives the information from an edition.

Figure 1-5 shows a document containing a publisher, a file containing an edition, and a document containing a subscriber. The bottom fish in the Fishes of the World document is a publisher. The information from this publisher is made available to other documents through the Illustration edition. The Aquarium poster document contains a subscriber that gets its information from the Illustration edition. Note that when a user selects a publisher or subscriber within a document, your application should display a border surrounding the publisher or subscriber.

In general, when a user modifies the contents of a publisher and saves the document, your application should write the new data to the edition. The Edition Manager then informs all open applications with documents that subscribe to the edition that the edition contains updated information. These applications can then automatically update the subscribers in

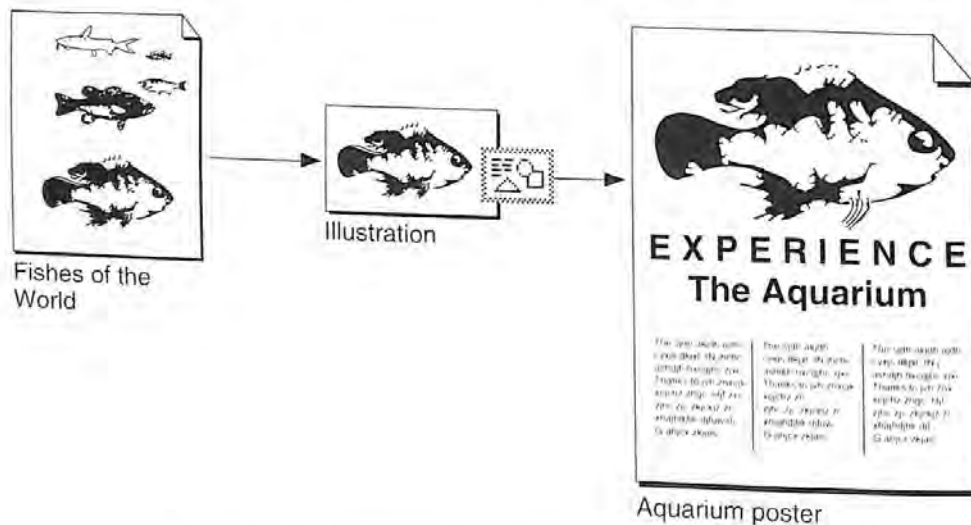


Figure 1-5. A publisher, an edition, and a subscriber

the documents. For example, in Figure 1-5, if the user changes the color of the fish in the Fishes of the World document and then saves the document, the change can be automatically made in the Illustration edition and the Aquarium poster document.

Figure 1-6 shows how a user might create a poster by using information from other documents. For example, the user could subscribe to separate editions containing an illustration created by a graphics designer, text created by a writer, and a headline created by an editor.

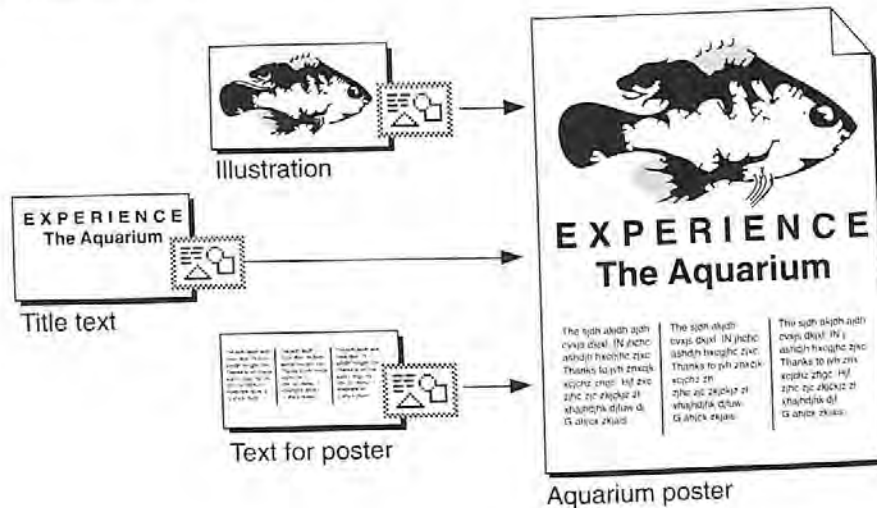


Figure 1-6. Sharing dynamic data with other applications

Your application should save the new information in the edition whenever the user edits the publisher and saves the document that contains the publisher—unless the user has indicated that the information should be saved in the edition on request only. Saving new information in an edition replaces the previous contents of the edition.

When the information in an edition changes, the Edition Manager informs your application. Your application should then update any subscribers with the new information from the edition (unless the user has indicated that updates should be incorporated on request only).

For example, a user might open a word-processing document called My Stocks that accesses information from an edition called Stock Report. The Stock Report edition might be updated twice a day by an on-line database. As the information in the edition changes, the My Stocks document can receive automatic updates with the latest information.

You can implement publish and subscribe capabilities in your application by using the routines provided by the Edition Manager and supporting the required set of Apple events. See the Edition Manager chapter for sample code that shows how to add these features to your application.

Sending Events Between Applications

The Macintosh Operating System provides routines that allow your application to send and receive events using the Apple Event Manager and Event Manager. The Event Manager provides a general method for communication between applications. The Apple Event Manager provides a standard method of communication between applications using the Apple Event Interprocess Messaging Protocol. (The PPC Toolbox can be used to read and write low-level message blocks and is more useful for applications that are closely integrated or perform coordinated tasks.)

Using the Apple Event Manager or Event Manager, applications can send events to other applications to request services or information. You can send these events between applications on the same computer or between applications located on different computers on a network. The Apple Event Manager uses the services of the Event Manager to send and receive Apple events. The Event Manager uses the communication services of the PPC Toolbox on behalf of your application to send and receive events.

For high-level events and Apple events, the applications involved must agree on what they can ask each other and on the action that should be taken in each situation. Both the application sending the event and the application receiving the event must agree on the protocol of communication.

Your application should support at least the required set of Apple events sent by the Operating System. If you plan to implement publish and subscribe capabilities, your application should also support the Apple events sent by the Edition Manager. You can also implement other common Apple events or design your own customized Apple events. In addition, sets of Apple events exist for many specific categories of applications (for example, word processors or spreadsheets).

If your application acts on an Apple event, it should perform the standard action requested by that event. This helps ensure that other applications (and eventually users) can send an event to a particular type of application and expect the other application to understand and act on the event in a standard way.

In most cases, you should use Apple events to communicate with other applications. However, if necessary, you can implement your own protocol for high-level events. Figure 1-7 shows how two applications might use high-level events. For example, a user might need to update the telephone numbers of everyone in the marketing department. To accomplish this, the user might use a word-processing application to send a high-level event with the new telephone numbers across a network to a directory application running on a Macintosh computer at the company's headquarters. When the telephone directory application receives the high-level event, it updates its directory with the new telephone numbers.

See the Event Manager chapter in this volume for information on how to send and receive high-level events. See the Apple Event Manager chapter for information on the Apple Event Interprocess Messaging Protocol.

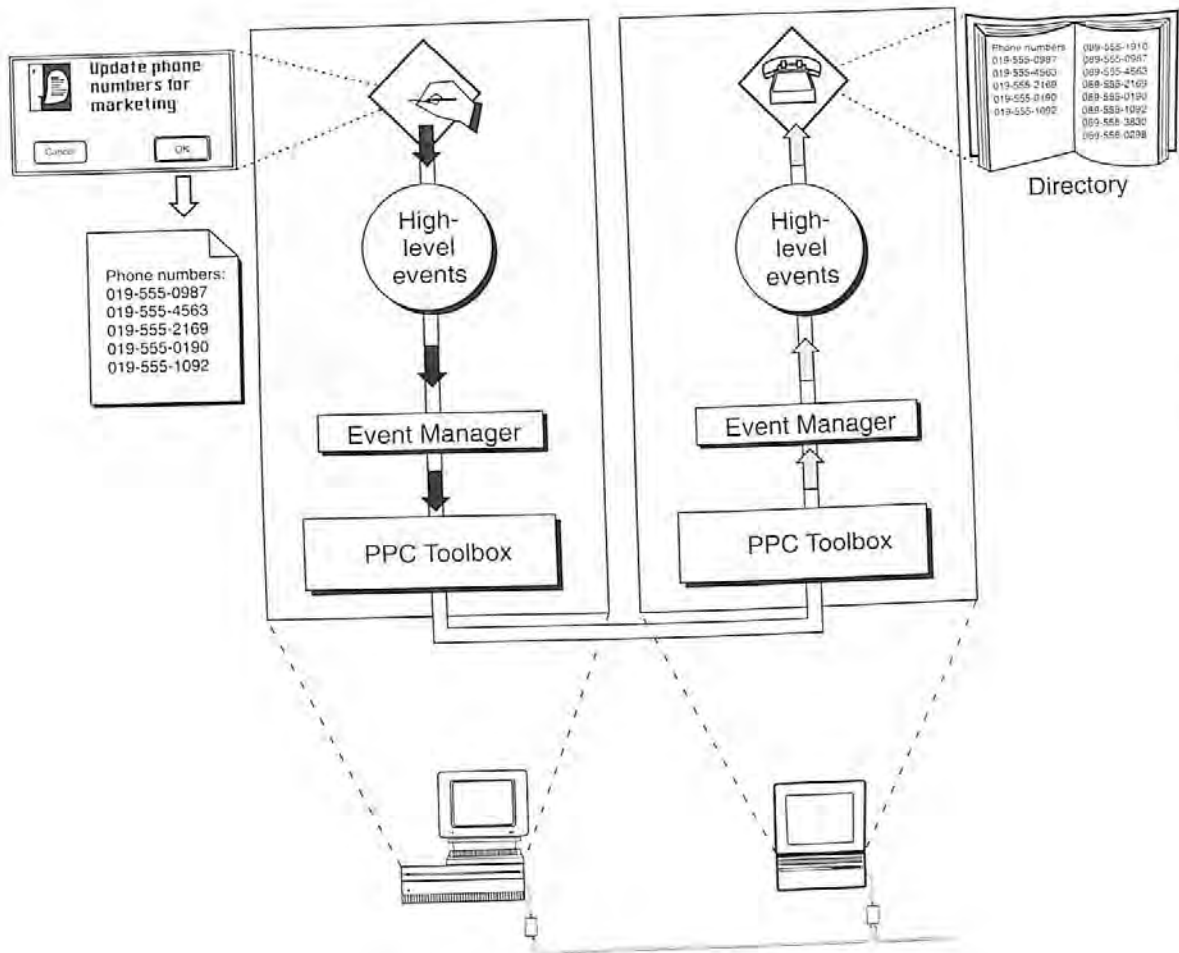


Figure 1-7. Sending events to other applications

Exchanging Message Blocks Between Programs

Using the Event Manager or Apple Event Manager to send events should meet the needs of most applications for program-to-program communication. However, for low-level control or to get services not provided by the Event Manager or Apple Event Manager, you can use the PPC Toolbox. The PPC Toolbox lets you send larger amounts of data to other applications located on the same computer or across a network. The PPC Toolbox can also be used by pieces of code that are not event-driven. The PPC Toolbox is usually called by the Operating System; device drivers, desk accessories, or other code modules can also use it.

Using the PPC Toolbox to send data between programs requires that both your program and the program you're communicating with are open at the same time. To initiate communication, one program opens a port and requests a session with another program. The target program must also open a port and accept the request. Once a session is established, the two programs can read and write low-level message blocks.

See the Program-to-Program Communications Toolbox chapter in this volume for information on reading and writing low-level message blocks between programs.

Remote Data Access

Using the Data Access Manager, your application can communicate with databases or other data sources running on a Macintosh computer or on a remote host computer. For example, your application can use high-level routines to open a document containing commands to be sent to a remote data server; initiate communication with the remote data server; send the commands to the server; and (after the server executes the commands) retrieve the requested data from the server. You can also use the Data Access Manager to send data to a remote database or other data source.

If your application knows how to create commands for a remote data server, then your application can use low-level routines to send these commands and data directly to the data server.

Figure 1-8 shows how a user in San Francisco might use a spreadsheet application to request data from a company database in New York. The spreadsheet application can use the Data Access Manager to request the data from the database. The database application in New York sends back the requested data, and the spreadsheet application can use this data to generate a graph of the information.

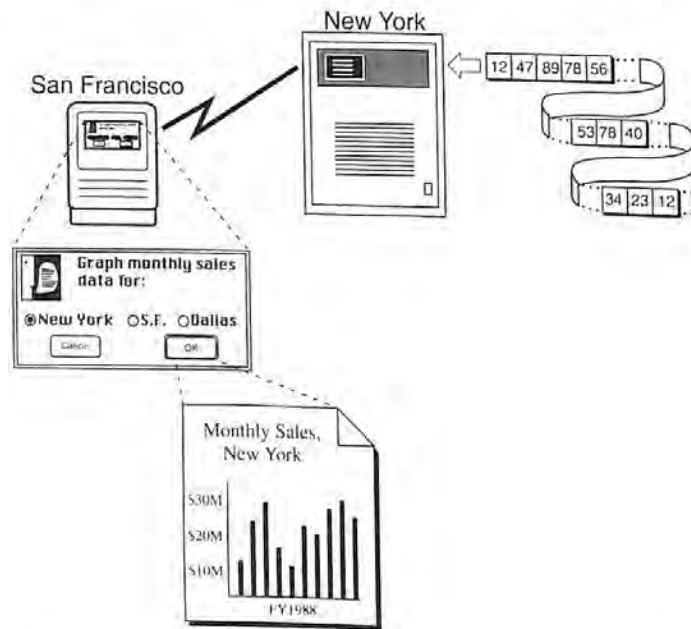


Figure 1-8. Requesting data from a remote database

See the Data Access Manager chapter for information on sending and retrieving information from a remote database or other data source.

Enhanced User Interface

The user interface for system software version 7.0 contains noticeable improvements, such as support for movable modal dialog boxes, and several new features. The Apple menu can now contain applications, documents, folders, or other Finder objects. You can supply small icons that the Finder displays in the Apple menu for your application and documents created by your application. Names of open applications now appear in the Application menu, a new menu to the right of all other menus. The Finder displays the small icon for your application in the right side of the menu bar whenever your application is active.

The structure of the System Folder has changed, including the addition of new folders that reside inside the System Folder. You can now store preference files in the Preferences folder and temporary files in the Temporary Items folder.

The Control Panels folder, which is inside the System Folder, replaces the Control Panel desk accessory. Control panels now appear as individual documents in the Control Panels folder. The user can open the Control Panels folder from the Finder or the Apple menu. In addition, if you develop video cards, you can create an Options dialog box that is used with the Monitors control panel.

In version 7.0, fonts, desk accessories, keyboards, international resource collections, and sounds are represented as icons on the desktop. The user installs fonts and sounds by dragging their icons to the System Folder icon. The user can store desk accessories in the Apple Menu Items folder within the System Folder or anywhere in the volume. You can now distribute fonts and desk accessories as movable resource files with separate icons.

The Finder now lets you create one or more icons for a single document or other desktop object; one of the icons represents the real object, and the others are aliases that point to the object. Aliases can give convenient access to documents that are nested within many folders or that reside on a file server.

The Finder can display help balloons with descriptive text when the user moves the cursor to certain elements of the Finder user interface while help is activated. In addition, if you use standard windows in your application, the Help Manager automatically displays help balloons for standard elements of the window, like the title bar and close box. You can use the features of the Help Manager to display help balloons for other elements of the user interface of your application. For example, you can create help balloons for menus, dialog boxes, and controls used by your application.

See the Control Panels, Finder Interface, Help Manager, and User Interface Guidelines chapters in this volume for information on these user interface features.

Sound

Your application can create and play sounds, mix and synchronize multiple channels of sound, expand and compress sound data, record sound, and play sounds continuously from disk using the Sound Manager.

The Sound Manager provides a rich set of routines for producing sounds, from playing a single sound to playing a set of digitally recorded sounds. You can also compress sound data for efficient storage of sound data on disk, and expand compressed sound data in real time.

See the Sound Manager chapter in this volume for complete information on using sound in your application.

TrueType Fonts

System software version 7.0 provides support for TrueType fonts. The Font Manager uses equations (instead of bitmaps) to define the appearance of glyphs in TrueType fonts. After using the equation to define a specific glyph in a particular font, the Font Manager translates the outline to a bitmap for display on the screen.

The advantage of TrueType fonts is that a single TrueType font can be used to generate glyphs at any size. The TrueType font includes instructions that fine-tune the image of the font at different sizes. TrueType fonts are also resolution independent; the same TrueType font can generate glyphs on a 72 dpi device or a 300 dpi device.

Your application can immediately take advantage of TrueType fonts if they are supported by the user's system software. However, the Font Manager still supports bitmapped fonts, and gives preference to bitmapped fonts over TrueType fonts if both are available for a specific typeface at a particular size.

To offer full support for TrueType fonts, your application can provide a menu command (such as Size or Other) to let the user choose any size of a TrueType font. Your application can also request that the Font Manager always choose TrueType fonts over bitmapped fonts.

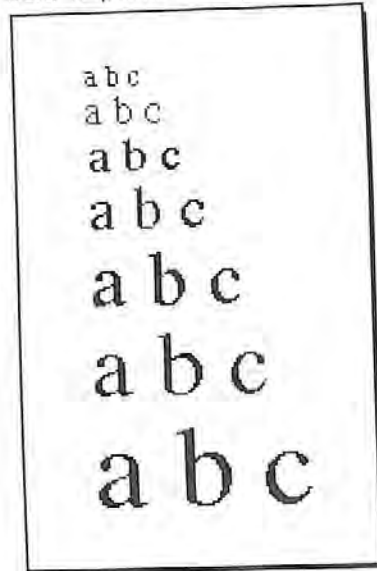
Figure 1-9 shows an example of on-screen glyphs generated using a TrueType font and a bitmapped font. The left side of the figure shows glyphs in a TrueType font that is rendered at 12, 16, 19, 24, 31, 37, and 45 points. The right side of the figure shows glyphs in a bitmapped font scaled at the same sizes.

See the Font Manager chapter for an introduction to TrueType fonts and for information on using TrueType fonts in your application.

Graphics

The Macintosh User Interface Toolbox provides a rich set of routines that support graphics. Using the Toolbox routines, your application can provide fast and high-quality graphics and visual display to the user.

TrueType font scaled on screen
from 12 points to 45 points



Bitmapped font scaled on screen
from 12 points to 45 points

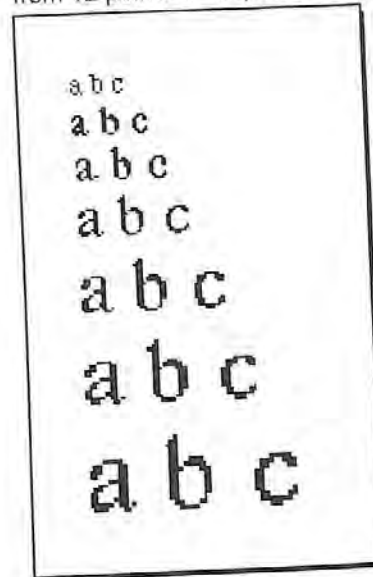


Figure 1-9. Comparison of TrueType and bitmapped fonts

You can use the routines provided by QuickDraw to draw text, straight lines, ovals, rectangles, or any variety of shapes. QuickDraw lets you define multiple drawing environments (ports)—each with its own coordinate system, location on the screen, and other characteristics. QuickDraw also performs automatic clipping of drawing environments—preventing another application from drawing in the drawing environment used by your application. QuickDraw manages all drawing to the screen and provides a flexible set of routines your application can use to perform most graphics operations.

Color QuickDraw provides support for gray-scale and color devices. In addition, users can connect multiple monitors of different sizes, depths, and color capabilities. Color QuickDraw automatically draws to the appropriate screen and takes advantage of the special characteristics of that device.

Color QuickDraw in version 7.0 supports both indexed and direct devices. Indexed devices typically have a color look-up table with 256 entries, meaning that up to 256 different colors can be displayed at once on the screen. The user's video card and monitor determine the number of bits per pixel and the number of colors that can be displayed on the screen. For indexed devices, Color QuickDraw supports 1, 2, 4, or 8 bits of information per pixel.

Direct devices do not use a color look-up table; instead, the video card contains enough RAM to directly store color information for each pixel. This allows direct devices to display up to 16 million colors. For direct devices, Color QuickDraw supports 32 bits of information per pixel (although only 24 are actually used). See the Graphics Overview chapter in this volume for a comparison of indexed and direct devices.

Using the Palette Manager, you can create palettes for your application. A palette is a convenient way to group collections of colors. You can also use palettes if your application makes special uses of color—for example, if your application needs color table animation. See the Palette Manager chapter in this volume for information on the default color tables supplied with version 7.0 and for information on how to set up and maintain palettes.

You can use the Color Picker Package to offer users a standard dialog box for choosing a color. The user can choose any color from the entire range the available device can display. See the Color Picker Package chapter in this volume for information on how to display the Color Picker dialog box and for a description of the various color models used by the Color Picker Package.

You can examine the contents of pictures and pixel maps using the Picture Utilities Package. See the Picture Utilities Package chapter in this volume for more information.

You can use offscreen graphics to prepare images in a graphics environment you create and then move the images quickly into view. The Graphics Devices Manager lets your application get information about particular graphics devices and provides routines your application can use if it needs exacting control of the graphics environment.

For an introduction to graphics on the Macintosh computer, see the Graphics Overview chapter in this volume. If you're developing a graphics-intensive application, see the Color QuickDraw, Palette Manager, and Graphics Devices Manager chapters in this volume for information on routines that provide advanced graphics features.

System software version 7.0 also provides support for color icons. See the Finder Interface chapter in this volume for information on how you can create color icons for your application and the documents it creates.

File Management

Your application can easily locate the files it needs by using alias records. An alias record is a data structure that identifies a file, folder, or volume. Whenever your application needs to store the location of a file or directory that it might need later, you can record the location and other identifying information in an alias record. The next time your application needs the file or directory, you can use the alias record to locate it, even if the user has renamed it, copied it, restored it from backup, or moved it. You can also use alias records to identify objects on other volumes, including AppleShare® volumes. The Alias Manager provides routines for managing the information in alias records.

Note that the Finder creates alias objects that are visible to the user, while your application usually creates alias records when it needs to store identifying information about a file or directory that it uses internally.

You can also quickly search a disk for particular files using File Manager routines. You can search for one or more files that match certain criteria that your application specifies. For example, your application can search for all files that have a modification date later than June 15, 1991, and the File Manager returns to your application a list of all files that match this specification.

In version 7.0, individuals can share files with other users. A user can make all files within one or more of the folders on a local disk available over a network. This increases the chance that documents created by your application are used in a shared environment.

The File Manager provides a new standard format for identifying files. You can use this standard format in File Manager routines, and other managers also accept files specified in the new format.

The user interface for opening and saving a file is enhanced in version 7.0. The Standard File Package provides two new procedures, StandardGetFile and StandardPutFile, that your application can use to display the standard user interface for choosing a file. To customize the user interface for choosing a file, you can use the new CustomGetFile and CustomPutFile procedures.

See the File Manager chapter in this volume for information on identifying and locating files on a volume, see the Standard File Package chapter for information on letting the user choose a file, and see the Alias Manager chapter for information on using alias records.

Memory Management

The Macintosh Operating System manages the loading of applications, desk accessories, and other code into and out of memory. Applications must share the amount of memory available. Without virtual memory, if an application needs a greater amount of memory than is currently free for application use in the user's system, the user must free up some memory. With virtual memory, the Operating System can store elsewhere the contents of memory in use by other applications in order to make room for the active application.

Virtual memory extends the available memory beyond the limits of physical RAM by using part of the available secondary storage (such as a hard disk) to hold portions of programs and data not currently in use. When an application needs portions of memory stored on disk, the Operating System brings those portions back into physical memory by swapping them with other unused portions of memory.

The operation of virtual memory is mostly transparent to your application. The user sets options in the Memory control panel to control various features of virtual memory. The user chooses whether virtual memory is turned on and, if so, how much virtual memory is available. The main benefit of virtual memory is that it allows users to run more applications at once and work with larger amounts of data.

See the Memory Management chapter in this volume for further information on using virtual memory.

Temporary Memory

Your application can allocate temporary memory if it needs additional memory for short-term purposes. Your application is not always guaranteed the desired amount of memory, so it should work correctly even if it does not get the requested memory. For example, you might allocate a small buffer in your application heap to copy data, and request additional temporary memory. If the temporary memory is available, your application can use it to copy large amounts of data more quickly. If the temporary memory is not available, your application should still be able to perform the copy, although it might take a little longer. As soon as your application finishes using the temporary memory, you should release it so that the memory can be made available to other applications.

See the Memory Management chapter for further information on using temporary memory.

24-Bit and 32-Bit Addressing

For Macintosh computers that support 32-bit addressing, the Memory Manager in version 7.0 uses all 32 bits of a memory address when the 32-bit addressing setting in the Memory control panel is on. Earlier versions of system software use 24-bit addressing, in which only the first 24 bits of a memory address are significant, and the upper 8 bits are ignored. For compatibility, all machines that support 32-bit addressing also support 24-bit addressing.

Macintosh computers that support 32-bit addressing can run with either 32-bit addressing or 24-bit addressing, but not both at the same time. The user chooses 32-bit addressing or 24-bit addressing by changing the setting in the Memory control panel and restarting the computer.

Applications that use the upper 8 bits of a memory address do not work correctly in 32-bit addressing mode. Applications that strip the upper 8 bits of a memory address or rely on the structure of the Memory Manager heap also do not work correctly in 32-bit addressing mode. Therefore, your application should not directly manipulate the bits in a memory address. If your application can operate correctly in 32-bit addressing mode, you can indicate this to the Operating System by setting a flag in your application's 'SIZE' resource. See the Event Manager chapter for a discussion of the 'SIZE' resource.

If you use your own customized window definition functions or customized control definition functions, see the Memory Management chapter for guidelines on avoiding memory address violations. The Memory Management chapter also provides further guidelines on how to write an application that works with 32-bit addressing.

Process Management

System software version 7.0 provides support for process management. Your application can get information about any currently running process, including your own. For example, for a specified process, you can find the application's name, type and signature; the number of bytes in the process partition, the number of free bytes in the application heap, the application that launched the process, and other information. Your application can also launch other applications and desk accessories.

When a user opens a desk accessory in version 7.0, the Operating System launches the desk accessory in its own partition. When a desk accessory is open, the Finder puts the name of the desk accessory in the list of open applications in the Application menu, and also gives the active desk accessory its own About command in the Apple menu that includes the name of the desk accessory. This makes the user interface for desk accessories more consistent with the user interface of small applications.

You can achieve greater control over other applications using the Process Manager routines. You can bring an application to the front, get information about other applications, and launch other applications without terminating your own application. Your application can also receive notification if any application that it has launched terminates.

System software version 7.0 provides greater support for launching applications and documents at startup. All desktop objects in the Startup Items folder are automatically opened at startup. All background applications in the Extensions folder are launched early in the startup sequence before the Finder is started. Background applications generally perform a specific task and are invisible to the user. The Startup Items folder and Extensions folder are located inside the System Folder.

See the Process Management chapter in this volume for information on launching other applications and getting information on currently running processes.

Timing Services

You can schedule routines to execute at a later time using the Time Manager. The Time Manager provides a hardware-independent method of performing time-related tasks.

You can schedule routines to run periodically or after a specified delay. Time delays can be specified in milliseconds or microseconds in version 7.0. You can achieve a maximum resolution of 20 microseconds. This gives you greater accuracy in coordinating sound, multimedia, and other events that require precise timing.

See the Time Manager chapter in this volume for information on how to schedule a routine for later execution and how to compute elapsed time.

Compatibility

You can determine what features are available on a Macintosh computer using the Gestalt function. The Gestalt function provides information about various attributes, versions, and features of particular software and hardware available on the currently running system.

The Compatibility Guidelines chapter in this volume discusses guidelines you should follow to ensure that your application is compatible with previous versions of Macintosh system software as well as with new releases of Macintosh system software.

These guidelines can help you develop your application so that it is compatible across the Macintosh family of computers. The guidelines also provide information on how to make your application compatible with A/UX[®] and how to design your application so that it can be easily localized for use in other regions.

Worldwide Development

As you develop applications for worldwide markets, you need to consider differences in scripts, languages, and regions. The Macintosh system software presents one of the most flexible architectures for developing applications that can support more than one script.

A script, such as Roman, Kanji, or Arabic, is a writing system for a human language such as English, Japanese or Arabic. Scripts have different characteristics; for example, they can differ in the direction in which their characters and lines run and in the number of characters in their character sets. The way in which you need to input, display, render, and edit text may change depending on the script in use.

A script system is a collection of software facilities that provides for basic differences between writing systems. Script systems include character sets, fonts, keyboards, and routines for text collation and word breaks. Examples of script systems are Roman, Japanese, Arabic, Hebrew, Thai, Devanagari, and Korean. A script system can also be localized for a particular language, region, or country. For example, the Roman script system has been localized for French, British, Italian, and U.S. users (among others). The system software of all Macintosh

computers includes the Roman script system. If another script system is required, it is also customized for the particular language or region. You can use the Script Management System to help you display text in the correct format for various scripts.

Worldwide system software consists of the Macintosh Script Management System (that is, the Script Manager and one or more Macintosh script systems) and related components (including the International Utilities Package, the international resources, and keyboard resources).

Measurement systems often differ from country to country, as do currency, sorting order, word boundaries, and the formatting of dates and times. The International Utilities Package handles formats for the presentation of numbers, currency, time, and dates in countries around the world. The international resources and several of the keyboard resources also contain region-specific or language-specific information, such as date and time formats.

TextEdit also provides support for working with different script systems. You can use TextEdit to let the user edit and display text in multiple scripts and styles when a non-Roman script system is in use. TextEdit automatically handles text with more than one script, style, and direction. For example, TextEdit supports mixing English text (a left-to-right directional script) with Arabic text (a right-to-left directional script) in the same line.

You should use resources to store text for menus, dialog boxes, and other parts of the user interface of your application. This lets a translator localize your application for a particular language, region, or country without requiring modification of your code. In addition, by using routines provided by the Macintosh Script Management System, you can write your application so that it works independently of the particular script in use.

Figure 1-10 shows a document created by an application that uses the Macintosh Script Management System to support more than one script system.

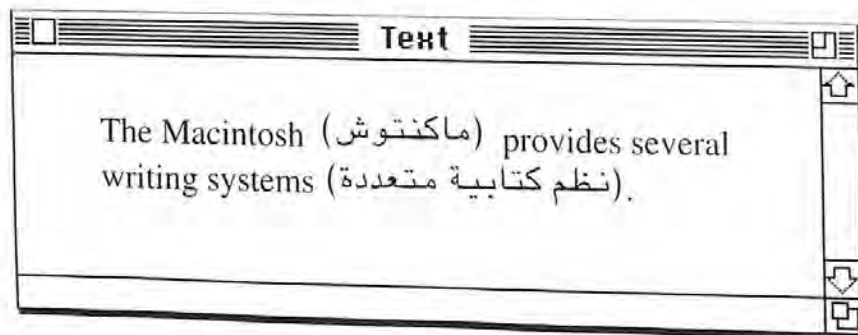


Figure 1-10. Using multiple scripts in a single document

See the Worldwide Software Overview chapter for an introduction to designing your application for worldwide markets, and see the User Interface Guidelines chapter for guidelines related to developing your application for use around the world. See the TextEdit chapter for information on using TextEdit when a non-Roman script system is in use. *Macintosh Worldwide Development: Guide to System Software* (available from APDA®) provides a complete description of all components of the worldwide system software, including routines in the Script Manager.

Communication Over a Network

The Macintosh Operating System provides many routines to support applications communicating and sharing data across a network. You can send events between applications located on different computers using the Event Manager or Apple Event Manager, and read and write low-level message blocks using the PPC Toolbox. You can send and retrieve information from a remote database or other data sources using the Data Access Manager. You can share data and files between applications on different computers using file sharing, the Edition Manager, and the Alias Manager.

In addition, you can use the network and communication services provided by the AppleTalk® Manager or Communications Toolbox. The AppleTalk Manager provides routines your application can use to send and receive information over an AppleTalk network.

The AppleTalk Manager in version 7.0 supports various link access protocols (for example, the LocalTalk® Link Access Protocol and the EtherTalk® Link Access Protocol) that can be used for AppleTalk communication. Your application can also use a new protocol, the AppleTalk Data Stream Protocol (ADSP), to exchange information between two equal entities. Either end of an ADSP connection can send data at any time. You can use ADSP to establish two-way communication between computers—for example, for use in office conferencing. See the AppleTalk Manager chapter for information on the device drivers and protocols associated with AppleTalk.

The Communications Toolbox provides your application with a standard interface for various communication services (such as data connections, file transfer, and terminal emulation) that are often used with a modem, other serial connections, or over an AppleTalk network. See *Macintosh Communications Toolbox Reference Guide* (available from APDA) for additional information on the routines provided by the Communications Toolbox.

Hardware Interfaces

The Macintosh family of computers supports many different types of hardware, including mouse devices, keyboards, display devices, hard disks, floppy disks, CD-ROM discs, and other devices. These devices are supported through various hardware interfaces, including SCSI (Small Computer System Interface), ADB (Apple Desktop Bus™), and SCC (Serial Communications Chip). In addition, a number of different devices can be supported through the expansion interfaces (the NuBus™ and processor-direct slots).

You can design expansion cards and drivers for the NuBus and processor-direct slots. For specific hardware information for the Macintosh family, see the *Guide to the Macintosh Family Hardware*, second edition. For information on writing a driver for the Macintosh family, see *Designing Cards and Drivers for the Macintosh Family*, second edition, and for system software information, see *Inside Macintosh*, Volumes I–VI.

Volume VI (this volume) contains information on the new Power Manager and additional information on the Slot Manager. The Power Manager is a new manager used with the Macintosh Portable. The Slot Manager in version 7.0 supports 32-bit addressing of NuBus cards. See the Power Manager chapter and Slot Manager chapter in this volume for specific information on these managers.

OVERVIEW OF CHAPTERS IN THIS VOLUME

The User Interface Guidelines chapter and Compatibility Guidelines chapter provide important information about designing your application to take advantage of the Macintosh user interface and to ensure compatibility across the Macintosh family of computers.

The rest of the chapters in this volume show how to use the new features of version 7.0 in your application. Each chapter gives detailed descriptions of each manager, including routines, parameters, and data structures.

Figure 1-11 shows where you can find a detailed description of how to use each feature of the system software version 7.0 environment.

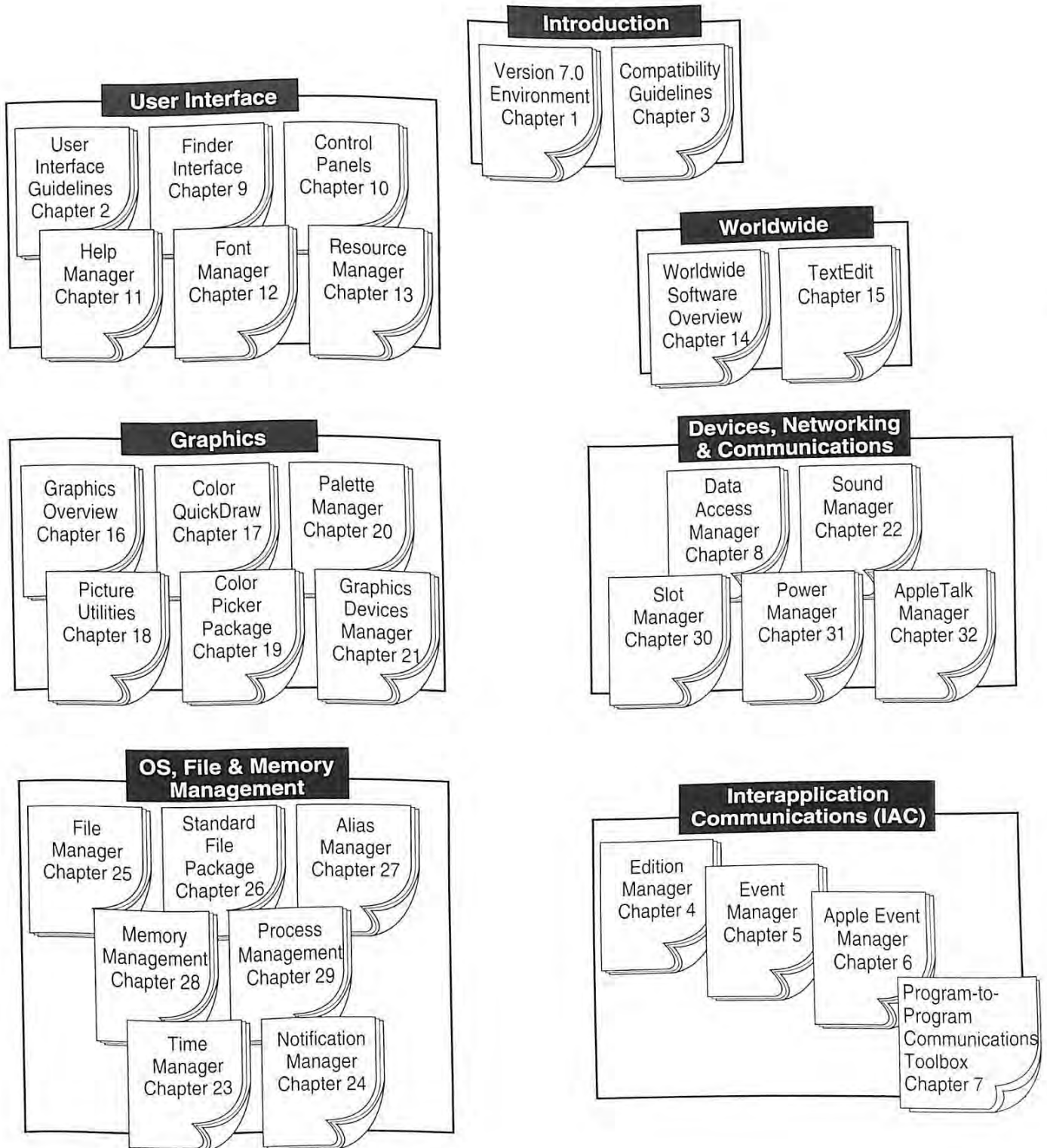
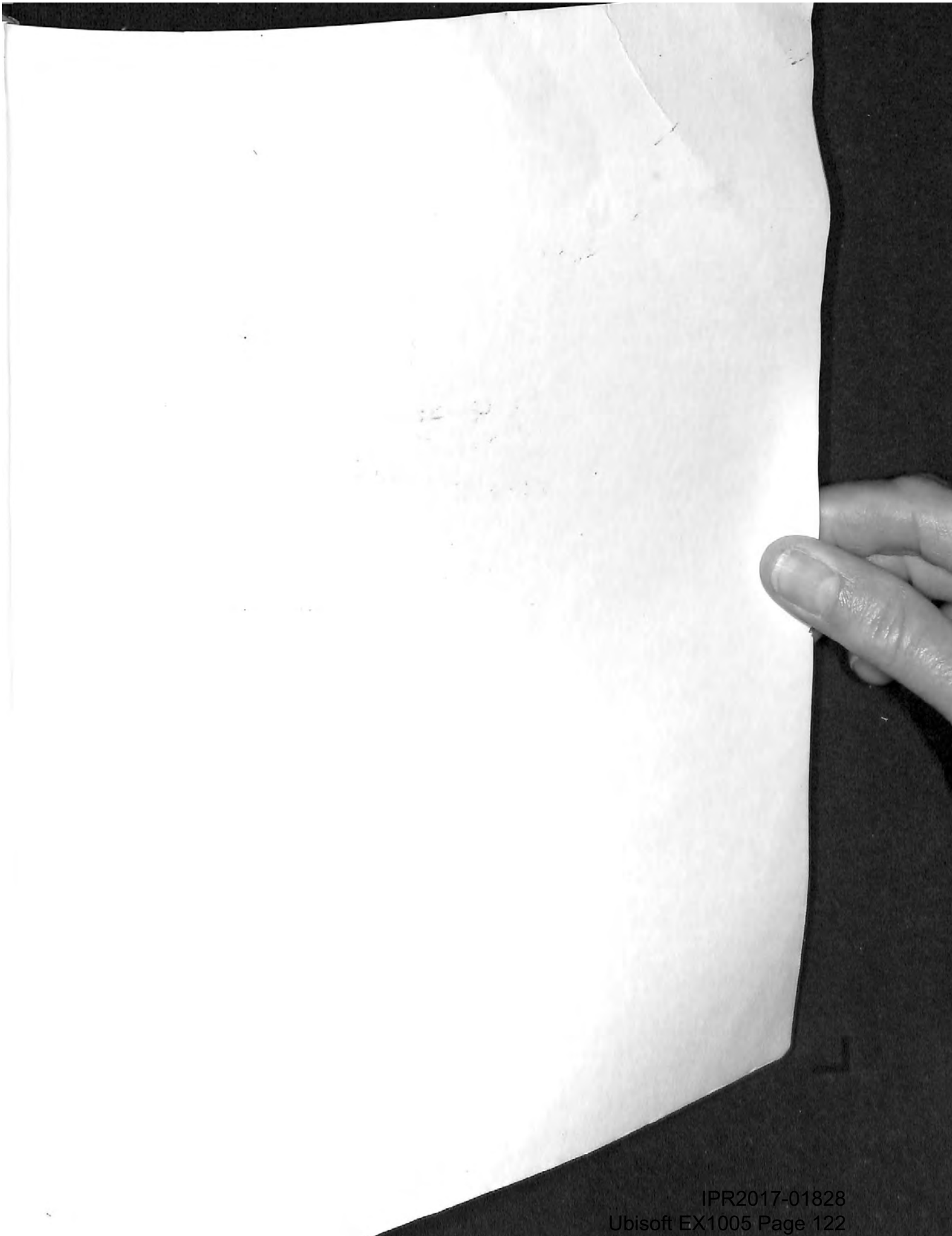


Figure 1-11. Overview of chapters in Volume VI

DATE DUE

NOV 5 1991	APR 21 1994
NOV 15 1991	UIC-REC'D MAY 07 '94
July 17, 1992	DEC 19 1995
OCT 22, 1992	
UIC-REC'D FEB 23 '93	
FEB 24 1993	JAN 18 1996
	JUN 19 1998
MAR 29 1993	
MAR 30 1993	JUN 10 1998
APR 30 1993	
UIC-REC'D MAR 18 '94	
MAR 24 1994	
MAY 15 1994	
CAYLORD	PRINTED IN U.S.A.





Keyword Local Catalog Only Find

Login
Ask A Librarian

Advanced Search | Classic Search | Course Reserves | Search History

Back to Search Results Cite this Email this Add to favorites Staff view More Information
Google

Inside Macintosh /

[written by Caroline Rose with Bradley Hacker ... et al.]

Main Author: Rose, Caroline
Other Names: Hacker, Bradley.
Published: Reading, Mass. : Addison-Wesley Pub. Co., c1985-1991.
Topics: Macintosh (Computer)
Tags: No Tags, Be the first to tag this record! Add

- More Details
- Location & Availability
- Table of Contents
- User Reviews
- Published Reviews
- Request Item

Northern Illinois University

Location: NIU--Main Collection--Founders Library
Call Number: QA76.M3 R674 1985
 Text me this call number

Copy: 1

Library Has (Volumes): v.1 c.1
 v.2 c.1
 v.3 c.1
 v.4 c.1
 v.5 c.1
 v.6 c.1

Status: v.6 c.1 - Checked out (Due: August 5, 2017)

Keyword Local Catalog Only Find

Advanced Search | Classic Search | Course Reserves | Search History

Libraries that Own Item

This screen shows libraries that own the item you selected.

Ask A Librarian

Location	Library	Local Holdings	Code
US,IL	CHICAGO PUB LIBR		CGP
US,IL	HAROLD WASHINGTON COL		IBJ
US,CA	ALIBRIS		ALBRS
US,CT	MARAKON ASSOC		M3R
US,CT	SOUTHERN CONNECTICUT STATE UNIV		CTN
US,DC	NAVAL RES LAB		NRL
US,DC	US NAVAL OBSERVATORY		DNO
US,DE	E J DU PONT DE NEMOURS		DLI
US,DE	WILMINGTON PUB LIBR		DWW
US,ID	IDAHO NAT LAB		ANW
US,IN	ALLEN CNTY PUB LIBR		IMF
US,IN	LAKE CNTY PUB LIBR		ILC
US,KY	FORT CAMPBELL, RF SINK LIBR		APK
US,MA	HEWLETT-PACKARD CO, APOLLO SYST DIV		APQ
US,MA	MARINE BIOLOGICAL LAB/WOODS HOLE OCEANOGR		MBW
US,MA	MASSACHUSETTS INST OF TECH, LINCOLN LAB		LIN
US,MA	MOTOROLA INFO SYST GROUP		OCX
US,MA	NOBLE, INC		NOG
US,MD	DIVISION OF CMP RES & TECH		DCR
US,MD	US ARMY, RES LAB		ADB
US,MI	DOW CHEMICAL CO, THE		MDC
US,NC	BAKER & TAYLOR INC TECH SERV & PROD DEV		BTCTA
US,NC	DURHAM CNTY LIBR		NZD
US,NC	SALEM COL	v.6	NSC
US,NC	UNIV OF N CAROLINA, ASHEVILLE		NIM
US,NH	YBP LIBRARY SERVICES		YDX
US,NJ	DEVRY UNIV LIBR		DYY
US,NJ	WOODBRIIDGE PUB LIBR		WRR
US,NY	PATCHOGUE-MEDFORD LIBR		SGK
US,NY	SUNY COL AT ONEDNTA		ZBM
US,OH	BAKER & TAYLOR		BAKER
US,OH	FINDLAY HANCOCK PUB LIBR		FPL
US,PA	TEMPLE UNIV HEALTH SCI CTR LIBR		TMH
US,TX	HOUSTON COMMUN COL		THC
US,VA	NATIONAL RADIO ASTRONOMY OBSERVATORY		RAO
Australia	SHORE SCH BOER WAR CONTINGENT MEM LIBR	Local Holdings Availa...	ATSHR
France	BIBLIOTHEQUE DE L'INPL NANCY 3		FNK
France	BIBLIOTHEQUE DE L'UNIV DE RENNES 1		ZUJ
France	BIBLIOTHEQUE UNIV PIERRE ET MARIE CURIE		FUJ
France	BIBLIOTHEQUE UNIVERSITAIRE DE METZ		ERM
France	BU DE LA MEDITERRANEE		FUM
France	UNIVERSITE DE LORRAINE, NANCY 1		BNB
Germany	D L R		EIJ
Japan	NIIGATA UNIV OF INT & INFO STUDIES		NWL
Korea	KOREA ADV INST OF SCI & TECH		KSZ
Malaysia	UNIVERSITI PUTRA MALAYSIA		MYPUT
New Zealand	WELLINGTON CITY LIBR		NZWPM
Singapore	NATIONAL LIBR BRD, SINGAPORE		SINLB
Singapore	TEMASEK POLYTECHNIC LIBR		SITPL
Taiwan	NATIONAL CHENG KUNG UNIV		CKT
United Kingdom	CAMBRIDGE UNIV		CUD
United Kingdom	SWANSEA UNIV		SNU

Record for Item: "Inside Macintosh. / Volum..." (Libraries with Item)

Get This Item

Availability: Check the catalogs in your library.

- Libraries worldwide that own item: 52
- Search the catalog at the Library of University of Illinois at Urbana-Champaign

External Resources:

- Discover UJUC Full Text
- Interlibrary Loan Request
- Cite This Item

Find Related

More Like This: [Advanced options](#)

Find Items About: [Inside Macintosh \(2\)](#) [Apple Computer, Inc. \(2,137\)](#)

Title: **Inside Macintosh, Volume VI.**

Corp Author(s): [Apple Computer, Inc.](#)

Publication: Reading, Mass. : Addison-Wesley Pub. Co., Year: 1991

Description: 1 volume (various pagings) : illustrations (some color) ; 28 cm

Language: English

Standard No: ISBN: 0201577550 ((pbk.)); 9780201577556 ((pbk.))

SUBJECT(S)

Descriptor: [Macintosh \(Computer\)](#)
[Macintosh \(Computer\)](#)

Note(s): By Apple Computer, Inc.--Cover / "Supplements Inside Macintosh, Volumes I through V, with information about system software version 7.0"--Cover

Class Descriptors: LC: [QA76.8 M3](#); Dewey: [005.255](#)

Vendor Info: Baker & Taylor Baker and Taylor YBP Library Services (BKTY BTCP YANK) 44.95 Status: active

Document Type: Book

Entry: 19911105

Update: 20150605

Accession Number: 24681256



LEADER 00000nam 2200001a 4500
 001 24681256
 005 19920422141835.0
 008 91118551991 maua 001 0 eng d
 020 0201577550 (pbk.) :
 040 IHF [CIMF]dW1H
 049 IZHW
 090 QA76.8.M3 [b157 1991
 092 005.265]bW31
 245 00 Inside Macintosh. |nVolume VI
 260 Reading, Mass. : |bAddison-Wesley Pub. Co., |cc1991
 300 1 v. (various pagings) : |b1 ill. (some col.) ; |c29 cm
 500 "By Apple Computer, Inc."--Cover
 500 "Supplements Inside Macintosh, Volumes I through V, with
 information about system software version 7.0"--Cover
 650 0 Macintosh (computer)
 710 2 Apple Computer, Inc.
 991 920586
 992 lc920506
 993 BOOK

LOCATION
UNCA.GENERAL

CALL #
QA76.8.M3 I57 1991

STATUS
NOT CHK'D OUT

Google



Scholar 3 results (0.05 sec)

My Citations

All citations Volume VI, Apple Computer, Inc., 1991

Articles Search within citing articles

Case law

My library [CITATION] Learn C on the Macintosh

D Mark - 1995 - dl.acm.org

Any time Cited by 1 Related articles All 2 versions Cite Save More

Since 2017

Since 2016 [CITATION] Learn C on the Macintosh: Includes Special Version of Symantec's Think/Disk

Since 2013 D Mark - 1991 - dl.acm.org

Custom range... Related articles Cite Save More

Sort by relevance: [PDF] Learn

[PDF] ebooksbucket.com

Sort by date D Mark - 1993 - ebooksbucket.com

Some time ago, back in 1988 or 1989, I remember reading an article written by someone at Apple Computer. The article was addressed to all Macintosh programmers, and spoke to the future of Macintosh software development. Back then, the vast majority of Macintosh

include citations Related articles All 17 versions Cite Save More

Create alert Create alert



Learn
on the **C++**
Macintosh[®]

INCLUDES
SPECIAL VERSION
OF SYMANTEC C++
FOR MACINTOSH

Dave Mark

Addison-Wesley Publishing Company
Reading, Massachusetts • Menlo Park, California • New York
Don Mills, Ontario • Wokingham, England • Amsterdam
Bonn • Sydney • Singapore • Tokyo • Madrid • San Juan
Paris • Seoul • Milan • Mexico City • Taipei

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Addison-Wesley was aware of a trademark claim, the designations have been printed in initial capital letters.

The authors and publishers have taken care in preparation of this book and software package, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

Library of Congress Cataloging-in-Publication Data

Mark, Dave.

Learn C++ on the Macintosh : includes a special version of
Symantec's C++ compiler / Dave Mark.

p. cm.

Includes bibliographical references and index.

ISBN 0-201-62204-1

1. Macintosh (Computer)--Programming. 2. C++ (Computer program
language) I. Title.

QA76.8.M3M36772 1993

005.26'2--dc20

93-61

CIP

Copyright © 1993 by Dave Mark

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America. Published simultaneously in Canada.

Sponsoring Editor: Keith Wollman

Project Editor: Elizabeth G. Rogalin

Production Coordinator: Gail McDonald Jordan

Cover design: Jean Seal

Icons in book designed by Crystal Sarno, Graphic Perspectives, Inc.

Set in 11 point Palatino by Rob Mauhar and Lenity Himburg, CIP

2 3 4 5 6 7 8 9 -MA- 9796959493

Second printing, January 1994

To the rest of the quadrangle:

Daniel, Deneen, and Stu.

See you all New Year's Eve!

Contents



Preface *ix*
Acknowledgments *xi*

Chapter 1 Welcome Aboard 1

What's in the Package? 3
Why Learn C++? 4
What Should I Know to Get Started? 4
What Equipment Do I Need? 5
The Lay of the Land 5
Strap Yourself In... 8

Chapter 2 Installing THIN C++ 9

Installing THIN C++ 12
Testing THIN C++ 13
Features of THIN C++ 16
Summary 20

Chapter 3 Remembering C 21

Literal Constants 24
Variables 28
Operators 36

v

Statements 41
Functions 42
Preprocessor Directives 43
Comments 43
Summary 44

Chapter 4 Introducing C++ 45

Getting C Code to Run under C++ 48
New Features of C++ 52
The // Comment Marker 52
Handling Input and Output 53
Default Argument Initializers 63
Reference Variables 66
Function Name Overloading 73
The new and delete Operators 77
The Scope Resolution Operator 84
inline Functions 87
Summary 90

Chapter 5 Object Programming Basics 91

The Organizational Power of the Struct 94
Encapsulating Data and Functions 95
Creating an Object 96
Accessing an Object's Members 97
Deleting an Object 100
Writing Member Functions 101
Access Priveleges 106
Bringing It All Together 109
Friends 115
Summary 123

Chapter 6 Derived Classes 125

- Inheritance** 128
- Access and Inheritance** 130
- Derivation, Constructors, and Destructors** 136
- Base Classes and Constructors with Parameters** 143
- Overriding Member Functions** 149
- Summary** 154

Chapter 7 Operator Overloading 155

- The Operator Overloading Alternative** 159
- A Few Restrictions** 168
- Covering All the Bases** 170
- Special Cases** 179
- Summary** 200

Chapter 8 Inside iostream 201

- The Character-Based Interface** 204
- The iostream Classes** 204
- Working with Files** 212
- The iostream State Bits** 216
- More File Info** 222
- Customizing iostream** 224
- Formatting Your Output** 231
- Using Manipulators** 236
- istream and ostream** 238
- Summary** 240

Chapter 9 C++ Potpourri 241

- Templates 244
- The Templates Approach 245
- Multiple Inheritance 256
- Resolving Ambiguities 264
- Initialization Versus Assignment 274
- Memberwise Initialization 278
- Summary 286

Chapter 10 Moving On 287

- The User Interface 289
- The Macintosh Toolbox 291
- Class Libraries 293
- Books and Other Reference Material 296
- Go Get 'Em... 299

Appendices

- Appendix A Glossary 301
- Appendix B Source Code Listings 307
- Appendix C fstream.h 345
- Appendix D iomanip.h 353
- Appendix E iostream.h 355
- Appendix F strtrea.h 379
- Appendix G Unofficial C++ Style Guide 385
- Appendix H The Complete Symantec C++
Development Environment 421
- Appendix I Bibliography 431

Bibliography

1. "Unofficial C++ Style Guide," David Goldsmith and Jack Palevich, in *develop*, *The Apple Technical Journal*, Issue 2, April 1990, Apple Computer, Inc., Cupertino, CA.
2. *The C++ Programming Language*, 2nd ed., Bjarne Stroustrup, 1991, Addison-Wesley Publishing Company, Reading MA.
3. *C++ Primer*, 2nd ed., Stanley B. Lippman, 1991, Addison-Wesley Publishing Company, Reading MA.
4. *Advanced C++: Programming Styles and Idioms*, James Coplien, 1992, Addison-Wesley Publishing Company, Reading MA.
5. *Elements of C++ Macintosh Programming*, Dan Weston, 1990, Addison-Wesley Publishing Company, Reading MA.
6. *C++ Programming with MacApp*, David A. Wilson, Larry S. Rosenstein, and Dan Shafer, 1990, Addison-Wesley Publishing Company, Reading MA.
7. *Inside Macintosh*, Volumes I–V, Apple Computer, Inc., 1991, Addison-Wesley Publishing Company, Reading MA.
8. *Inside Macintosh*, Volume VI, Apple Computer, Inc., 1991, Addison-Wesley Publishing Company, Reading MA.
9. *New Inside Macintosh*, Apple Computer, Inc., 1992–1993, Addison-Wesley Publishing Company, Reading MA.

10. *Macintosh Human Interface Guidelines*, 1992, Addison-Wesley Publishing Company, Reading MA.
11. *Macintosh C Programming Primer*, Volume I, 2nd ed., Dave Mark and Cartwright Reed, 1992, Addison-Wesley Publishing Company, Reading MA.
12. *Macintosh C Programming Primer*, Volume II, Dave Mark, 1990, Addison-Wesley Publishing Company, Reading MA.
13. *Macintosh Pascal Programming Primer*, Volume I, Dave Mark and Cartwright Reed, 1991, Addison-Wesley Publishing Company, Reading MA.
14. *Learn C on the Macintosh*, Dave Mark, 1991, Addison-Wesley Publishing Company, Reading MA.
15. *The C Programming Language*, 2nd ed., Brian Kernighan and Dennis M. Ritchie, 1988, Prentice-Hall, Englewood Cliffs, NJ.
16. *Macintosh Programming Secrets*, 2nd ed., Scott Knaster, 1992 Addison-Wesley Publishing Company, Reading MA.
17. *How to Write Macintosh Software*, 3rd ed., Scott Knaster, 1992, Addison-Wesley Publishing Company, Reading MA.
18. *The Annotated C++ Reference Manual*, Margaret Ellis and Bjarne Stroustrup, 1990, Addison-Wesley Publishing Company, Reading MA.