UNITED STATES PATENT AND TRADEMARK OFFICE
_____

BEFORE THE PATENT TRIAL AND APPEAL BOARD
_____

UBISOFT, INC. AND SQUARE ENIX, INC.,
Petitioners,

v.

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

U.S. Patent No. 6,510,466
_____

DECLARATION OF SCOTT BENNETT, Ph.D.
20 April 2017

TABLE OF CONTENTS

Page

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, P.A. to authenticate and establish the dates of public accessibility of certain documents in an *inter partes* review proceedings for U.S. Patent No. 6,510,466. For this service, I am being paid my usual hourly fee of $88/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;

1

- 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 cataloging 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. Members of my family are university researchers. Over the years, I have read some of the voluminous professional

2

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 14 December 1997.

9. *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

3

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.

10. *Persons of ordinary skill in the art*. I am told by counsel that the subject matter of this proceeding relates generally to managing application programs over a network.

11. I have been informed by counsel that a "person of ordinary skill in the art at the time of the invention" 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.

12. 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; and 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.

13. It is my opinion that such a person would have been engaged in advanced research, learning through study and practice in the field and possibly through formal instruction about the bibliographic resources relevant to his or her

4

research. In the 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.

14.   *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.

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

16.   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.

17.   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

5

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.

18.   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.

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

20.   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.

6

21.   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.

22.   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.

23.   *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.

24.   Indexing services use a wide variety of controlled vocabularies to provide subject access and other means of discovering the content of documents.

7

The formats in which these access terms are presented vary from service to service.

25. 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.

26. Prominent indexing services include:

27. <u>ACM Digital Library.</u> This index is produced by the Association for Computing Machinery, the world's largest scientific and educational computing society. AMC Digital Library contains the full text of all AMC publications, hosted full-text publications from selected publishers, and the ACM Guide to Computing Literature—a comprehensive bibliography of computing literature beginning in the 1950s with more than a million entries. All metadata in the database are freely available on the Web, including abstracts, linked references, citing work, and usage statistics. Full-text articles are available with subscription.

IV. OPINIONS REGARDING INDIVIDUAL DOCUMENTS

Document 1. Jeffrey F. Hughes and Blair W. Thomas. Novell's Guide to NetWare® 4.1 Networks. Jan Jose, CA: Novell Press, 1996.

1. **Authentication**

8

28.   Document 1 is a book by Jeffrey Hughes and Blair Thomas published by Novell Press in 1996. Attachment 1a is a true and accurate copy of the book's cover, title page, title page verso, preliminary matter, table of contents, and Chapters 1-5 and 12-13 from the Southwestern Illinois College. Attachment 1b is a true and accurate copy of that library's catalog record for Document 1, showing the book's location and availability.

29.   Attachment 1a is in a condition that creates no suspicion about its authenticity. Specifically, Chapters 1-5 and 12-13 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.

30.   Attachment 1c is a true and accurate copy of the ACM Digital Library index record for Document 1, showing the International Standard Book Number (ISBN) 156884736X, identical (except for format) to that provided on the verso of the title page in Attachment 1a: 1-56884-736-X.

31.   I conclude, based on finding Document 1 in a library and on finding library catalog records and an online record for Document 1, that Document 1 is an authentic document and that Attachment 1a is an authentic copy of Document 1.

2. **Public accessibility**

32.   Attachment 1d is a true and accurate copy of a Statewide Illinois Library catalog record for Document 1, showing this book held by 71 libraries world-wide. Attachment 1d also indicates that Document 1 was cataloged or indexed in a meaningful way—including being cataloged by subject. The date of entry in Attachment 1d is 5 January 1996, roughly two months before the publication of Document 1 (discussed immediately below). In my opinion, Document 1 was bibliographically identifiable by 5 January 1996. [1]

33.   Attachment 1e is a true and accurate copy of the United States Copyright Office record for Document 1. It shows the book was published on 8 March 1996 and registered for copyright on 3 April 1996. I conclude from this copyright record that Document 1 was publicly available from its publisher on or about 8 March 1996.

34.   Attachment 1f is a true and accurate copy of the Auburn University Libraries catalog record, in MARC format, for Document 1.   In Attachment 1f,

---

[1] Catalog records for books are sometimes publicly available before the book's publication. In this case, the book was published with the Library of Congress Card No. 95-82352 printed on the verso of the title page. This indicates the book's publisher had been in contact with the Library of Congress about the cataloging of the book before the book's printing and publication.

the MARC Field 040, subfield a, indicates that this catalog record for Document 1 was created by the Welch Medical Library at Johns Hopkins University (OCLC code = JHW). The MARC Field 008 indicates this catalog record was created on 24 April 1996, just a few days after the publication of Document 1. 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 May 1996.

3. Conclusion

Based on the evidence presented here—book publication, Copyright Office record, online indexing, and library cataloging—**it is my opinion that Document 1 is an authentic document that was bibliographically identifiable by 5 January 1996, was publicly available from its publisher on or about 8 March 1996, and was publicly available in at least one library by May 1996**.

V. ATTACHMENTS

35. 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.

36.   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.

37.   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 24 March – 14 April 2017 and all URLs referenced in this declaration were available 14 April 2017.

VI.   CONCLUSION

38.   In summary, I have concluded that Document 1, discussed above, is an authentic document that was publicly accessible before 14 December 1997.

39.   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.

40.   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 20th day of January, 2017 in Urbana, Illinois.

_____
Scott Bennett

13

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

14

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 Collaborary 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*

Novell's Guide to

NetWare 4.1 Networks

Jeffrey F. Hughes and Blair W. Thomas
Senior Consultants for Novell, Inc.

The Only Authoritative, Comprehensive, Expert Guide to NetWare 4.1

◆

Master NDS Tree Design, Partitioning, Replication, Navigation, Performance, and Security

◆

The Authors Unequaled Real-World Experience, which has Benefited Top International Corporations– is *Now Yours!*

Free Poster Inside!
Test Your NDS
Design Skills!

"An insightful guide to NDS, the heart of NetWare 4.1's powerful, innovative technology."

David Eckert, Product Manager, NetWare Directory Services, Novell, Inc.

NOVELL PRESS

Smart Books from the Novell Experts

SOUTHWESTERN
ILLINOIS COLLEGE
BELLEVILLE   GRANITE CITY   RED BUD

NOVELL'S®

▶ •

Guide to NetWare® 4.1 Networks

Jeffrey F. Hughes and Blair W. Thomas

NOVELL PRESS®

Novell Press, San Jose

IDG BOOKS WORLDWIDE

John Kilcullen, *President & CEO, IDG Books Worldwide, Inc.*
Brenda McLaughlin, *Senior Vice President & Group Publisher, IDG Books Worldwide, Inc.*
The IDG Books Worldwide logo is a trademark under exclusive license to IDG Books Worldwide, Inc., from International Data Group, Inc.

NOVELL PRESS

Rosalie Kearsley, *Publisher, Novell Press, Inc.*
Novell Press and the Novell Press logo are trademarks of Novell, Inc.

Preface

Since its release in 1993, Novell's NetWare 4 operating system has gained popularity as the network operating system of choice for large and small organizations around the world. Among its many new features, NetWare 4.1 offers NetWare Directory Services (NDS), which provides a powerful name service. NetWare Directory Services is unique in the industry and enables you to organize network resources, provide easy access to those network resources, and centralize the administration of your network.

This book is your comprehensive design and implementation guide to NetWare 4.1. Topics cover the fundamentals of NetWare 4.1 to managing the internal operations of NetWare Directory Services. The information presented in this book will help you design, implement, maintain, and troubleshoot all aspects of NetWare 4.1.

Although NetWare 4.1 offers a new and different approach to networking, its capabilities are enormous. The Novell Directory is still in its infancy as users, administrators, and vendors realize the potential of a name service. More and more, applications are being created for this technology.

What You'll Learn from Reading this Book

This book is written for all LAN administrators, system administrators, consultants, resellers, and any others who design, implement, and support NetWare 4.1 networks.

Using this book you will learn basic to complex concepts and rules on all aspects of NetWare 4.1. Whether your interest lies solely in designing a NetWare 4.1 tree or in understanding the internal functions and components of NetWare Directory Services, you will find this book to be the definitive source.

How this Book is Organized

This book is organized into four conceptual parts: Basic NetWare 4.1 information, NDS design topics, operations and maintenance of NetWare 4.1, and administration of NetWare 4.1.

Part I, The Basics of NetWare 4.1, details the essential features of the operating system and NetWare Directory Services, introduces NetWare 4.1 utilities, and provides readers a comprehensive look at NDS objects and how these objects are used to build the Directory tree. An understanding of NDS objects is a prerequisite to utilizing the full potential of NDS. Naming rules for navigating the tree from top to bottom are covered in the last chapter of Part I.

Part II, Designing NetWare Directory Services, describes in great detail the steps required to design an NDS tree and covers design options for NDS trees, with many examples based on the fictitious ACME company case study. In addition, the design topics include a thorough discussion of partitioning and replication of your NDS tree. The last chapter of Part II provides clear explanations and examples of how to design your NDS tree for time synchronization.

Part III, Operations and Maintenance of NetWare 4.1, provides a much-needed look at the internal operations of NetWare Directory Services. Internal operations of NDS in a single-server environment, as well as the steps that occur during the first NetWare 4.1 server installation, are discussed in detail. In addition, this part explains and thoroughly discusses the background processes of NDS through definitions and examples.

Next, we move on to more complicated multiple server operations that include partitioning operations and managing this complex network environment. The first two chapters in Part III provide a thorough discussion on troubleshooting NetWare Directory Services internal operations. Effective troubleshooting requires an understanding of NDS internal processes and the tools you can use to make adjustments to NDS. Refer to Chapter 10 for a detailed description on how to use the DSREPAIR and DSTRACE utilities during troubleshooting operations.

We conclude this section by discussing the tuning parameters you can use to maximize stability and performance of your NetWare 4.1 network.

Part IV, Administration of NetWare 4.1, rounds out the many day-to-day aspects of NetWare 4.1 that you will need to know in order to install, manage, and support your network. Topics include designing login scripts and access for NetWare 4.1,

providing a comprehensive security plan for your network, and printing in mixed NetWare 3 and NetWare 4 environments as well as printing in a pure NetWare 4.1 network.

This part also presents valuable information on preparing for a NetWare 4.1 migration, including creating an implementation schedule, gaining lab experience, and readying your NetWare 3 servers. Also covered are the migration options available to you, the advantages and disadvantages of these options, and migration strategies to move you from other operating systems such as Banyan, LAN Server, LAN Manager, and NT.

Client migration options are discussed in terms of the different approaches that are available to move workstations into Novell's Virtual Loadable Module technology.

The concluding chapter of this book outlines the steps involved in using a tape backup for NDS and presents valuable information on how to provide a high degree of fault tolerance for NetWare Directory Services.

The four appendices include valuable information that can be used as a handy reference. Appendix A lists, with descriptions and actions, the NetWare Directory Services error codes. Use Appendix A in conjunction with Part III of this book for more information on troubleshooting NDS.

Appendix B covers the server console commands. Server console commands enable you to change server parameters, monitor the server's utilization, and control the allocation of its resources.

Appendix C provides information on Novell's internal debugger and can be used to diagnose NetWare 4.1 operating system problems.

Appendix D outlines the many client parameters that can be placed in the user's NET.CFG file for greater performance of memory at the workstation.

Special Features in this Book

Throughout this book we use a fictitious company by the name of ACME, which serves as the basis for many of the examples illustrated in the book. We chose as an NDS tree a large, worldwide company with wide area connections. Our intent is not to preclude smaller companies, but to demonstrate as closely as possible the many concepts of NetWare 4.1. Regardless of the size of your NetWare

4.1 installation, the design strategies are the same. You may have fewer wide area connections or no WAN at all. As you read through the chapters (especially the design chapters), you will understand how to design a tree that meets your particular needs.

We have also, where appropriate, included Consulting Experiences based on actual consulting visits with our customers. These Consulting Experiences emphasize particular points or examples and demonstrate concepts and approaches to NetWare Directory Services design.

Finally, this book contains a value-packed CD-ROM containing a two-user version of NetWare 4.1. If you are new to NetWare 4.1, we encourage you to install this CD-ROM to learn and gain experience as you read through the book.

Sources

In the process of researching this book, we gratefully acknowledge the following Novell sources:

"Backing up and Restoring NetWare Directory Services in NetWare 4" (Novell Application Notes, August 1995)

"Tuning the Server Memory Calculation Worksheet" (Novell Application Notes, November 1995)

Contents at a Glance

Table of Contents

The Basics of NetWare 4.1

C H A P T E R   1

Introduction to NetWare 4.1

"The only way round is through." Robert Frost

NetWare is the world's most popular network operating system. With NetWare 4.1, Novell has incorporated all the advantages of previous versions of NetWare and has added new features that build on the foundation to provide a distributed computing infrastructure. NetWare 4.1 increases productivity, reduces costs, and simplifies installing and maintaining your network.

Key to NetWare 4.1 is NetWare Directory Services (NDS), which is a special purpose name service that enables you to find and use network resources and data as a single integrated system. NDS provides this powerful operating system with the same speed and reliability that you've come to expect with NetWare 3.

Many corporations have embraced and implemented NetWare as an integral part of everyday business computing solutions. Today, local and wide area networks are running business-critical applications once considered the sole domain of mainframe computers. More and more businesses are finding that they can operate more efficiently and cost-effectively on NetWare networks and still receive the same security and administrative benefits of larger systems.

With the advent of NetWare 4.1, NetWare has become more powerful, scaleable, and flexible than any other network operating system available today. Before understanding what NetWare 4.1 is, it helps to understand the purpose and function of a network operating system. From this starting point we will begin an extremely comprehensive look at the fastest growing network operating system currently available.

Network Operating Systems

A network consists of resources such as servers, workstations, printers, bridges, routers, gateways, and other peripheral computer equipment (CD-ROMs or jukeboxes, modems, and so on).

The network operating system (NOS) is software that communicates with each of these devices to form an integrated system. Some of the resources tied together by NetWare Directory Services are shown in Figure 1.1.

The primary goal of the network operating system and all the other resources on the network is to build an infrastructure that distributes the processing power among all the network devices. The major network components or devices are the server and client.

The architecture that distributes the processing between these machines is called distributed processing. The following sections describe the server, then the client, and finally explain how processing can be distributed between each.

FIGURE 1.1

NetWare Directory Services ties together the network resources from a central device called the server.
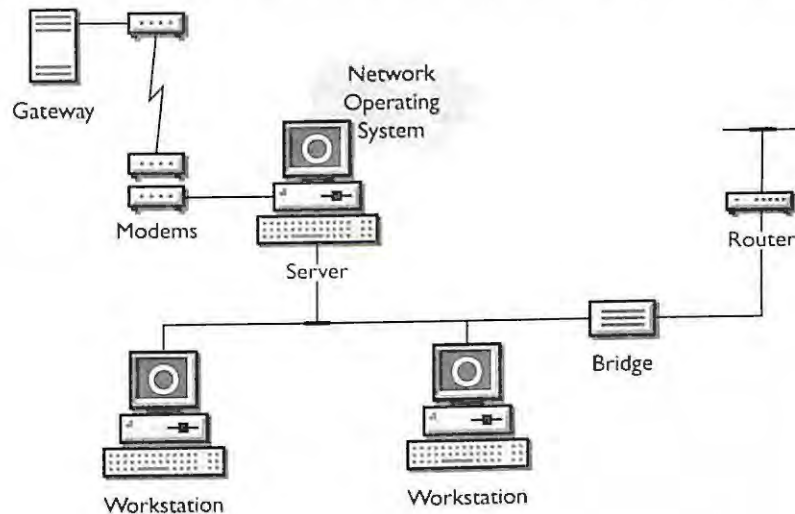
SERVER

The network operating system (NOS) is simply the software program that is loaded and runs on the main computers called servers. The NOS enables the local resources of the server to be shared among all the network users. The NOS controls the following resources:

► Memory

► File System (disk drives)

► CPU scheduling

► Input/output to shared network devices (CD-ROM, modems, and so on)

► Workstation connectivity and access to file system resources

► Loading and distributing application programs

Figure 1.2 shows the local server resources that are managed by the NOS.

FIGURE 1.2

The NOS manages the sharing of the server's local resources.
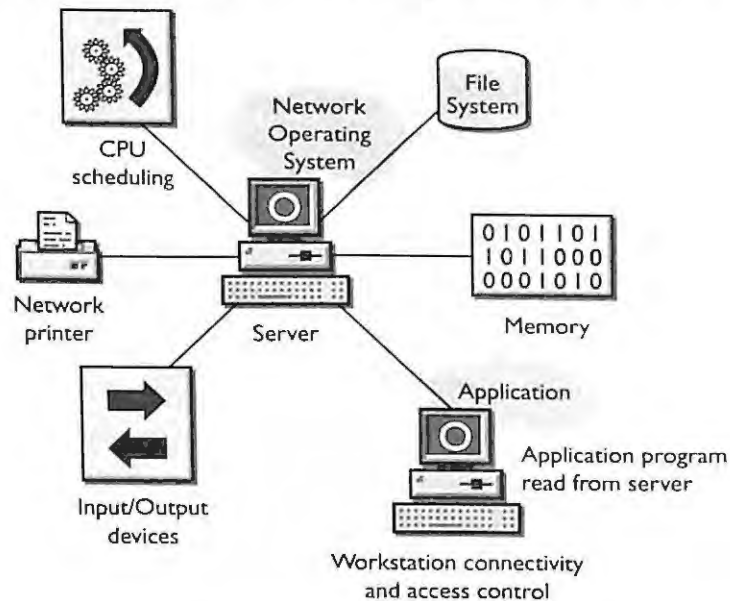
The NOS software runs as either a dedicated machine or can be distributed equally across all nodes on the network. An NOS that is distributed among all network nodes is called a peer-to-peer NOS. NetWare Lite is an example of a peer-to-peer NOS. A NOS whose major portion runs on a hardware platform is a centralized NOS. NetWare 4.1 is an example of a dedicated NOS in which the central node is called the server.

CLIENT

The portion of the NOS that connects the workstations to the server is called the client software. The client software runs on the workstations and is the consumer of services provided by the NetWare server. It also enables the workstation to load applications or share resources from the server. Figure 1.3 illustrates the relationship between the NOS running at the server and the client software running at the workstation.

Workstations that load the client portion of the network operating system are often referred to as clients. In this book, we will use the terms workstation and client interchangeably. Technically, only the workstations that load the client software and make use of the services provided by the servers should be called clients. NDS servers also act as clients of other NDS servers in the tree, but not in the same way as a workstation. For example, an NDS server may communicate with other NDS servers during a login process.

FIGURE 1.3

Client software loads on the workstation and enables it to connect to the server.

Client software connects and logs in to the server.

Network Operating System

Client Software

Server
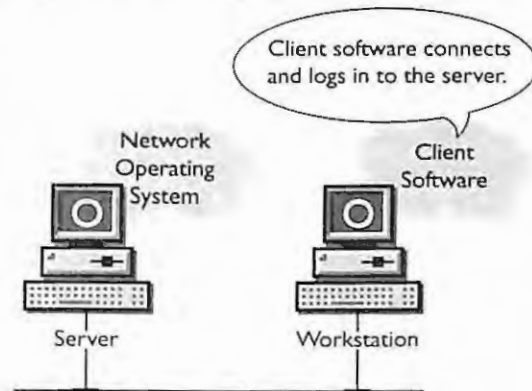
Workstation

Distributed Processing

Distributed processing occurs when the NOS coordinates the processing in a decentralized fashion. Applications running on the network are a type of distributed processing. For example, each workstation requests from the operating system its own copy of an application. The application is then loaded and run entirely on the workstation. The workstation runs the application independently, and the network operating system simply coordinates access to the shared resources. In the network, application processing can be distributed (client-based) or centralized (server-based) or both (client/server-based). Each of these methods offers advantages and disadvantages that make them right for certain applications and wrong for others.

CLIENT-BASED APPLICATIONS

The types of applications that run entirely on the workstation are called client-based applications. For example, all the clients may use the same word-processing application, but separate copies of the application are executed in each workstation. A copy of the application is transmitted from the server to the client but is executed at the client workstation. In general, applications that are keyboard and display intensive with minimal disk I/O are well suited for the client.

Spreadsheets and word-processing software are examples of client-based applications because they are heavy users of the display and keyboard with infrequent disk or file access. There is no benefit in running any part of these applications outside the workstation

once the program has been loaded into workstation memory from the server. Figure 1.4 illustrates client-based application processing in which the application is loaded from the server.

FIGURE 1.4

Client-based application processing. The application is loaded from the server.

SERVER-BASED APPLICATIONS

Applications that do all the processing locally at the server are known as server-based applications. These applications are usually specialized applications that execute only on the server. For example, backup/restore software needs to read the file system and write it to the local tape device (on the server) and may not need to transfer data across the network. Other types of backup software run as clients on a server and transfer files from other servers. Another example is network management software that controls or monitors the operation of the operating system. In addition, NetWare Loadable Modules are primarily server-based applications, although some have a client counterpart that does some of the processing. Figure 1.5 illustrates server-based application processing.

FIGURE 1.5

Server-based application processing
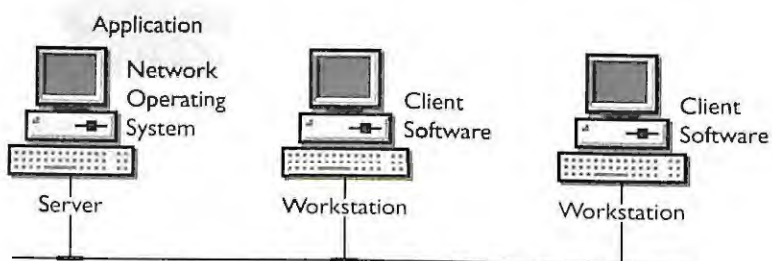
CLIENT/SERVER-BASED APPLICATIONS

Client/server-based applications split the responsibility of the processing between the client and the server. The client and server work together to execute the application even though it is running on different machines. The fact that the parts of the application are running on different machines is entirely transparent to the user. The software that runs on the server is called the back-end and manages the shared information. The client portion of the application is called the front-end and allows communication and access to the server (or back-end). Figure 1.6 illustrates client/server application processing.

FIGURE 1.6

Client/Server application processing
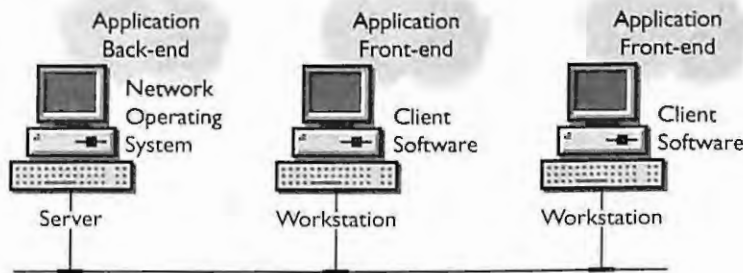
What is NetWare 4.1?

NetWare 4.1 offers the widest range of distributed applications or network services in the industry. In Figure 1.7, you can see that NetWare 4.1 provides file, print, directory service, database, communication (includes host connectivity), messaging, network management, software distribution, imaging, and telephony.

FIGURE 1.7

NetWare 4.1 is defined as a complete range of network services.
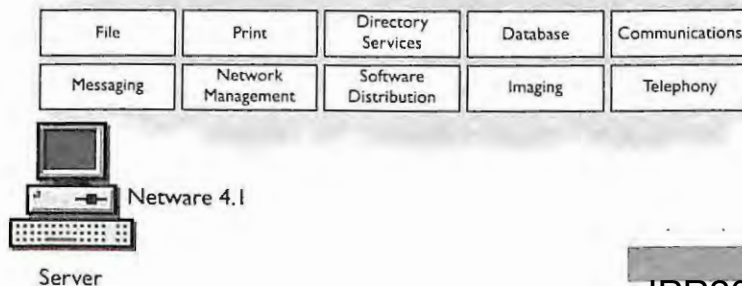
These network services are provided to the users regardless of their type of desktop. The goal of NetWare 4.1 is to tie users running DOS, MS Windows, OS/2, Macintosh, UNIX, or Windows NT into a distributed information system, as shown in Figure 1.8.

NetWare 4.1 allows these diverse workstations to access NetWare services and perform distributed processing in their native environments. Figure 1.9 shows how it is possible for users to choose the desktop system that best fits their needs and still share services and information with other network users using different platforms.

FIGURE 1.8

NetWare 4.1 connects desktops running DOS, Windows, OS/2, Macintosh, UNIX, and Windows NT operating systems.
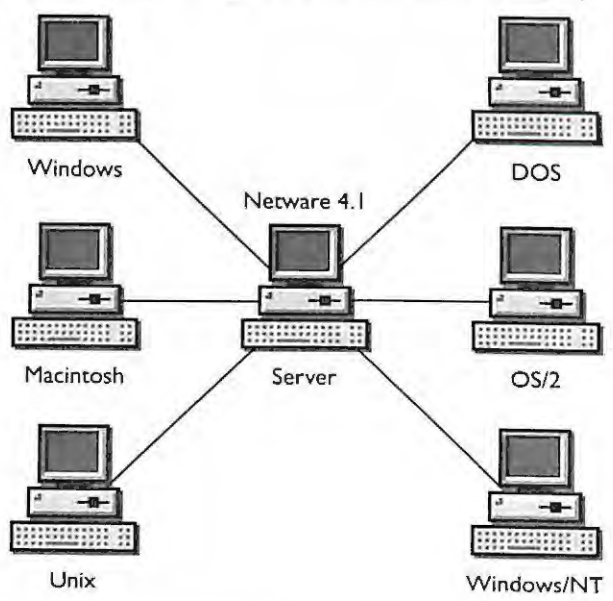
In addition to the integration of desktops, NetWare 4.1 also integrates larger host computers from vendors such as IBM, Digital, and Hewlett-Packard. Regardless of where information is located on the network, users can access host-based resources and information from their desktops.

Applications that can benefit from client/server-based implementation are database, communication, and transactional applications that require frequent access to disk storage. Consider a database that searches, sorts, generates reports, and so forth. The database will perform better if you place the database engine on the server (where the disk I/O is intensive) and process the data entry and user interface at the client. The client simply passes the data request to the database engine, which performs the action and responds accordingly.

NetWare 4.1 offers distributed processing among workstations running DOS, Windows, OS/2, Macintosh, UNIX, and Windows NT operating systems.
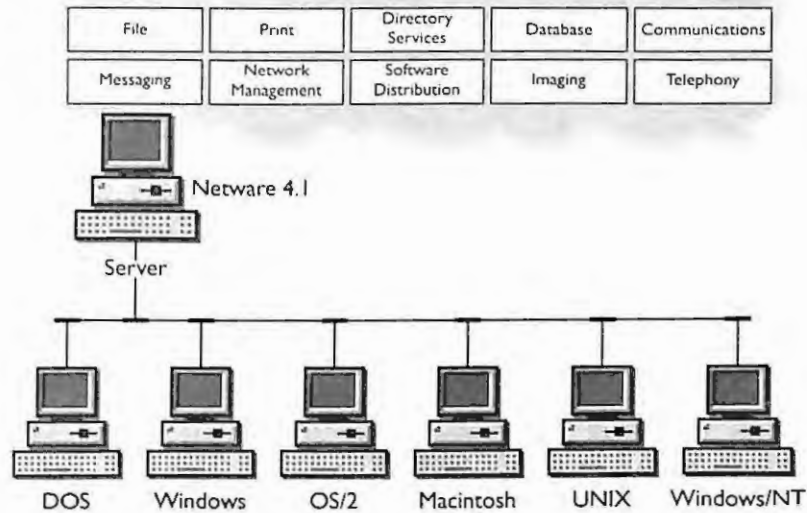
Another example would be an e-mail system in which the user reads and composes mail messages at the client and then passes the responsibility for delivery to a server process. Thus, while the server is delivering the messages to the other mail users, the client (or sender) can continue other activities.

NetWare 4.1 Features

NetWare 4.1 is the next generation of the NetWare operating system and thus inherits all features from previous versions, namely NetWare 3. Some the features shared by both NetWare 3 and NetWare 4.1 are listed below; however, we will not discuss these features in detail in this chapter because they are well known. Here is the list:

- File and Print

- Desktop Heterogeneity of DOS, Windows, OS/2, Macintosh, UNIX (NetWare 4), and NT (NetWare 4)

- Performance with a 32-bit multitasking operating system

- ▸ Modular design using NetWare Loadable Modules (NLMs)

- ▸ Fault Tolerance with NetWare SFT III

- ▸ Transport protocol support of IPX, TCP/IP, Appletalk, OSI, SNA

- ▸ Name space support for DOS, OS/2, Macintosh, Network File System (NFS), OSI's File Transfer Access Management (FTAM)

- ▸ Security

- ▸ Server-based backup architecture on SBACKUP.NLM

- ▸ Common transport interface, such as STREAMS, TLI, and BSD sockets

- ▸ Remote Management Facility (RCONSOLE)

- ▸ Host Connectivity

In addition, NetWare 4.1 introduces many improvements and features not available in previous versions of NetWare. These features include improvements to the existing services such as file systems (suballocation, data compression, read ahead, and data migration), print services, security, auditing, server memory, client software, and others. The most notably new feature is NetWare Directory Services, which provides an easy, intuitive, and integrated view of the network resources to both the user and administrator.

NETWARE DIRECTORY SERVICES

NetWare Directory Services (NDS) is an information name service in NetWare 4.1 that organizes network resources — users, groups, printers, servers, volumes, and other physical network devices — into a hierarchical tree structure. Figure 1.10 illustrates the structure of an NDS tree. We will use many examples of NDS trees in this book.

The NDS tree, also known as the Directory tree, allows resources to be managed and displayed as a single view. By contrast, NetWare 3 provides only a server centric view. You can manage the tree including objects and their various properties by providing varying degrees of security access, giving your network enormous flexibility as it expands and changes.

NDS replaces the bindery found in NetWare 3 networks. A major difference between the two methods is that the Directory is distributed and can be replicated on multiple servers for increased fault tolerance and performance. The bindery in NetWare 3 is a flat structure in which the resources belong to a single server. Compatibility with the bindery is available in NetWare 4.1 for applications requiring bindery services. This feature in NetWare 4.1 is known as Bindery Services and is discussed in great detail in subsequent chapters.

NDS provides capabilities unique to NetWare 4.1. These capabilities are single login, easy administration, and scaleability.

FIGURE 1.10

The structure of an NDS tree
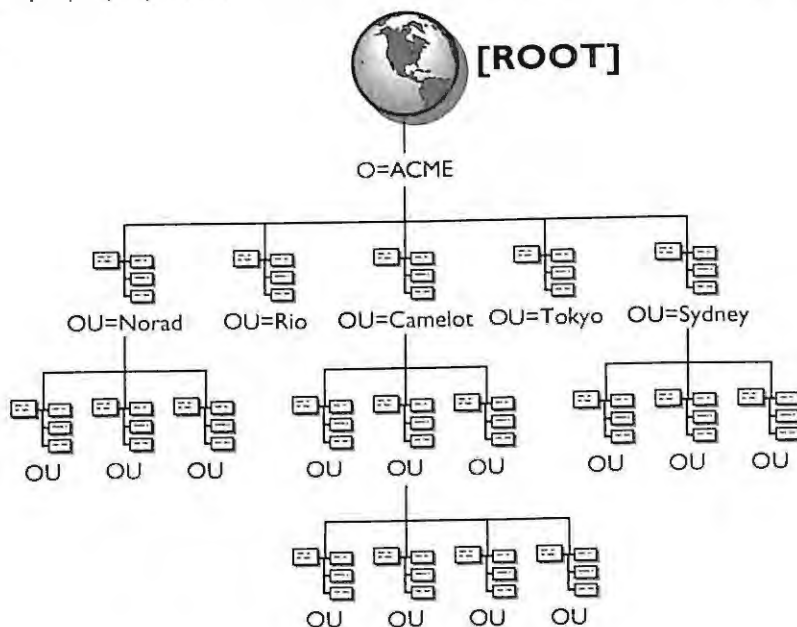
NDS Provides Single Point of Login

The users log in to the network once using one username and password to access all authorized network resources. This means that the users log in to the network, and NDS will process other connections to NetWare 3 and NetWare 4.1 servers if the username and password are the same.

By contrast, NetWare 3 bindery users must log in to each network server individually with a username and password. This could mean that the users would have to provide a different ID and password for each additional server connection. Single user login also

IPR2017-01290
Ubisoft EX1008-1 Page 58

makes your job easier as an administrator. You create each user account only once for all NetWare 4.1 servers in the Directory tree. Multiple user accounts on multiple servers are no longer needed in NetWare 4.1. This feature alone can save you hours of work. As you will see, NDS serves as the central point of management, eliminating tedious duplication and increased administrative costs.

NDS Provides Easy Administration

NetWare 4.1 consolidates most NDS administrative functions into a single, easy-to-use graphics utility that greatly reduces the time you spend on network administration. The Novell utility, called NWADMIN.EXE, is a Windows-based utility that enables you to make changes to the Directory with an easy point-and-click of the mouse. You can get information about a particular object by clicking on its icon. The icon will bring up a dialog box that displays object details that can be modified. This utility is also available in a DOS version known as NetADMIN.

This object-oriented view of the NDS tree is what lets you perform many routine administrative functions easily. NDS objects, files, directories, and server functions can be controlled through the NWADMIN utility. When users change departments, for example, you simply drag and drop the users to a new location to give them appropriate directory rights, or move users with the move command. Adding file system directory rights for a user in previous versions of NetWare requires a multilayered menu and a somewhat tedious process of adding each new user. With NetWare 4.1 all that is required to add directory rights is to drag the user's icon to the specific directory or object, and the NWADMIN utility will ask you for a confirmation prior to completing the request.

NDS is Scaleable

NDS is an object hierarchy that can be divided into smaller sections that can be distributed to any number of network servers. We say it is scaleable because one server does not need to contain all Directory information. With NDS, the information can be distributed and also replicated on multiple servers to provide increased accessibility and reliability. Figure 1.11 shows how NDS can be distributed across the servers in the network.

The feature that divides the hierarchy into smaller pieces is called partitioning. Replication is the mechanism that makes the partition redundant. The partitions and replicas are completely transparent to the users and can be scattered across multiple NetWare 4.1 servers. These features make NDS a powerful facility for storing, accessing, managing, and using information about the network resources regardless of where they

are physically located. This means that your NDS tree can easily grow to meet the demands of your environment. Another benefit is that your NDS tree design can be easily modified to reflect both the organizational and functional changes in your company.

New NDS Features

NetWare 4.1 has added new utilities and features that help simplify your administration of the network. These new features are unique to NetWare 4.1; previous versions of NetWare 4 (v4.01 and v4.02) did not have them. These features and utilities include: merge tree, rename tree, move subtree, rename container, NETSYNC, DSSTAT, DSMAINT, and DUPGRADE. For more information regarding NetWare 4.1 utilities refer to Chapter 2.

FIGURE 1.11

NDS provides distribution of the information across the network servers.
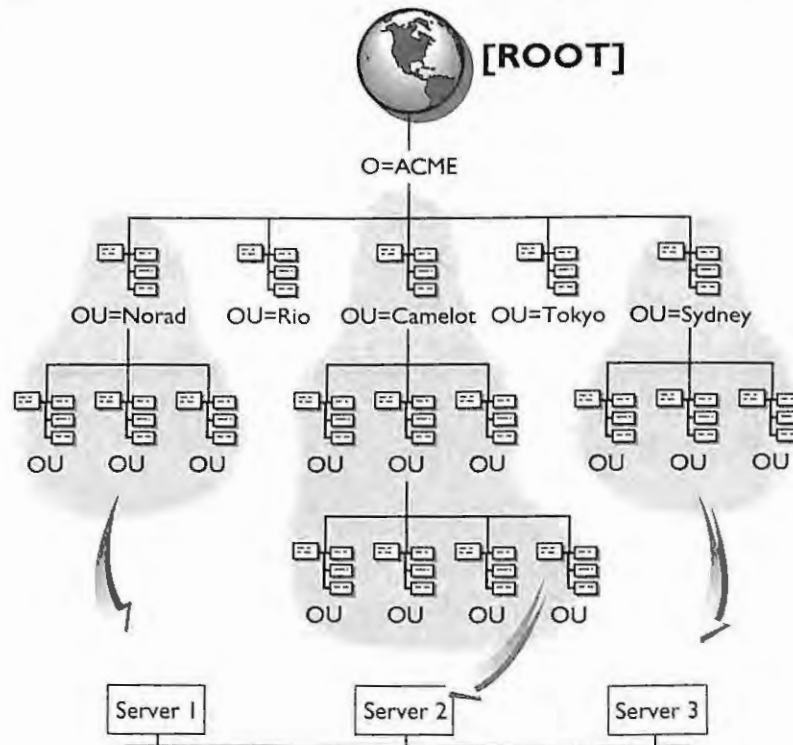
Merge Tree   The DSMERGE utility gives you the ability to merge two separate NetWare 4.1 Directory trees. This feature also gives you greater flexibility during your implementation of NetWare 4.1 and eliminates the task of having to always install every NetWare 4.1 server to the same NDS tree. If a department with an

existing NetWare 3 server wants to upgrade their server (for example, from a NetWare 3.12 250-user to a NetWare 4.1 1,000-user), they can upgrade the server and create their own departmental NDS tree that can be merged into the main corporate tree when appropriate.

Rename Tree One feature of the DSMERGE utility enables you to rename a Directory tree. This is helpful if you are merging together two trees with the same name. The DSMERGE utility requires that each tree have a separate name.

In addition, if two trees from different networks are joined to the same network, one of the trees will have to change its name by using the DSMERGE utility.

Move Subtree Using the PARTMGR utility, you can move an entire subtree (or partition) to another place in the tree. This feature gives you the flexibility to reorganize your tree to match the dynamic organization of your company.

Rename Container Using the NWADMIN utility, you can rename any container object in the tree (excluding the [ROOT] object). This feature was one of the most requested items prior to the release of NetWare 4.1 because it greatly simplifies this administrative task for managing containers.

CONSULTING EXPERIENCE — RENAMING CONTAINER OBJECTS THAT ARE HIGH IN THE TREE

When you rename a container object that is high in the tree (such as the O=Organization object), you are in reality renaming every object below the O=Organization. The renaming process has to change every object's *distinguished name* below the current container. (See Chapter 4 for more information about distinguished names.) Your tree size will determine how long this operation will take.

NETSYNC Utility You can make changes to a NetWare 4.1 server's users or groups and have those changes automatically reflected on the NetWare 3 servers by using the NetWare 4.1 utility called NETSYNC. NETSYNC automatically synchronizes updates to user and group objects made on the NetWare 4.1 server to selected NetWare 3 servers.

For example, you can add a user to the NetWare 4.1 server and that user will be automatically added to the appropriate NetWare 3 server's bindery.

NETSYNC is a set of NLMs that runs on both NetWare 4.1 and NetWare 3 servers. With NETSYNC installed, up to 12 NetWare 3 servers can be administered from one NetWare 4.1 server. All user, disk, and printer information normally controlled by SYSCON utility under NetWare 3 can be created or modified in NWADMIN on the NetWare 4.1 server. You can manage both versions of NetWare with one utility.

NETSYNC allows you to gradually migrate from NetWare 3 to NetWare 4.1 but still centrally manage mixed network environments.

DPSTAT Utility   This utility checks the partition synchronization status across all of a partition's replicas. You can select a specific level in the Directory tree and check the partition there. You can also choose to have the utility check all the branches of the tree below the select point. The synchronization status of all the replicas is then displayed. This partition status is available in DS versions 489 and up and is also provided in the DSREPAIR utility.

DSMAINT Utility   This utility enables you to change the server hardware without completely reinstalling NDS. For example, if you need to upgrade or rebuild the SYS volume on your NetWare 4.1 server, you can use this utility to copy NDS to another device, replace the SYS volume hardware, reinstall the operating system, and then copy the NDS files back on to the newly created SYS volume. All the current object IDs are maintained.

DUPGRADE Utility   NDS has been separated from the operating system (SERVER.EXE) so that new versions of NDS can be easily implemented on your NetWare 4.1 servers. This utility helps you distribute a new version of NDS (DS.NLM) to all servers in a tree. DSUPGRADE copies a specified DS.NLM to the SYS:SYSTEM on all the servers located in the selected portion of the tree.

FILE SYSTEM IMPROVEMENTS

NetWare 4.1 optimizes the use of the server file system with several new improvements over previous versions of NetWare. These improvements are: suballocation, data compression, read ahead, and data migration. These improvements enable NetWare to take full advantage of the available disk space and offer speed improvements as well as savings in server memory.

Suballocation

One of the cost benefits to migrating to NetWare 4.1 is the use of volume block suballocation. Suballocation in NetWare 4.1 subdivides the volume blocks (regardless of size) into smaller suballocation units of 512 bytes. This process ensures that you will never waste more than 511 bytes when saving any file to disk. You now have the flexibility to set your NetWare 4.1 volume block size to the recommended maximum of 64K without sacrificing valuable hard disk space, thus ensuring that disk space is more efficiently used.

Previous versions of NetWare worked with a fixed volume block size determined at the time of installation. The block size became the smallest unit of storage on the NetWare volume and server's hard drive. Generally, the larger the block size, the better the throughput between the file server's hard drive and the file server's RAM or cache. But the large block sizes also had the greatest potential of wasting disk space or creating disk slack.

For example, if your block size is 16K (without suballocation) and you save a small file of 1K, the remaining 15K in the disk block is not used (slack). Figure 1.12 shows that large block sizes without suballocation create slack. Figure 1.13 shows the same example but with suballocation, which reduces slack.

FIGURE 1.12

Large block sizes without suballocation create slack.

16K VOLUME BLOCKS (WITHOUT SUBALLOCATION)

1k | 15k SLACK | 3k | 13k SLACK

FILES TO BE STORED

FILE1.DAT (1K)

FILE2.DAT (3K)
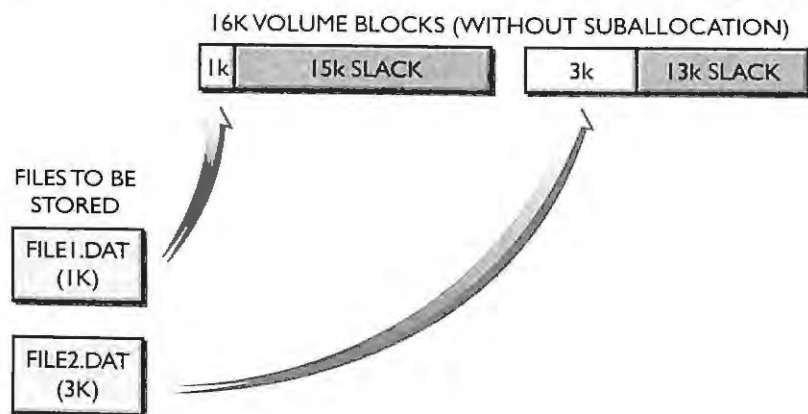
FIGURE 1.13

*Large block size with
suballocation significantly
reduces slack.*

16K VOLUME BLOCKS (SUBALLOCATED INTO 512-byte UNITS)
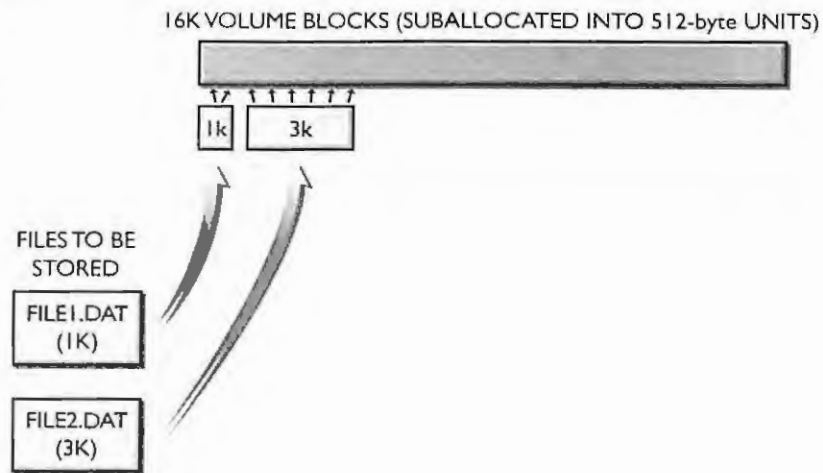
You will notice that in Figure 1.13 the volume block size is still 16K, but the block has been suballocated by the NetWare 4.1 operating system into 512 byte blocks. FILE1.DAT, a 1K file that would ordinarily allocate a full 16K, only takes the required 1K using two suballocated blocks. The other 15K is not wasted but can be used by other files written to the disk. Another file, FILE2.DAT (3K file), is written to disk and starts in the middle of the volume block using six more suballocated blocks. There is no slack in this example.

New files can start in the middle of a volume block that has been suballocated. They do not have to start on the volume block boundary. This means that if you select the recommended 64K volume block size, then several files can be started and stored in the middle of the 64K block.

CONSULTING EXPERIENCE

Some customers will see a return as high as 50 percent of a server's hard disk space by simply migrating to NetWare 4.1. This savings does not include file compression; server data is simply being migrated to NetWare 4.1.

This implies that NetWare 3 servers may have up to 50 percent slack. Slack space is automatically reduced by moving to NetWare 4.1.

The slack can be estimated to equal the total number of files * (volume block size /2). The total savings in disk space from doing the migration from NetWare 3 to NetWare 4.1 can be estimated using the following equation:

```
Savings = total number of files * ((old block size / 2) -
( new block size / 2)).
```

Some example numbers follow:

1 • Slack in NetWare 3: 50,000 files * (4096 bytes / 2) = 102.4 Mb

2 • Slack in NetWare 4: 50,000 files * (512 bytes / 2) = 12.8 Mb

3 • Savings for migrating to NetWare 4.1 from NetWare 3 is:

```
50,000 * ((4096 / 2) - (512 / 2)) = 89.6 Mb
```

File Compression

NetWare 4.1 provides the capability to automatically compress any files on the server that have not been used for a period of time. This compression can result in space savings of up to 50 to 60 percent of a volume in some cases. The File Allocation Table (FAT) continues to show the files as if no change has occurred. When the user wants to retrieve the file, the operating system automatically decompresses the file. This file compression and decompression process is completely transparent to the user.

The file compression feature in NetWare 4.1 is optional, and you can determine which servers can benefit from its use. You can choose which files, directories, or volumes to compress. The compression process runs as a background task, which means that it will never affect the performance of the file server. To enable file compression, use the following SET parameter (all the following SET parameters can be set in the SERVMAN console utility):

```
SET ENABLE FILE COMPRESSION = ON
```

You can decide the hours during the day when you want the file compression process to start and stop scanning for files that meet the compression criteria. To set the hour in the day to start scanning type:

```
SET COMPRESSION DAILY CHECK STARTING HOUR = 0
```

The default start time is 0, which represents 12:00 PM The possible range is 0 through 23, which represents each hour of the day. To stop scanning type (default is 6 or 6:00 AM):

```
SET COMPRESSION DAILY CHECK STOP HOUR = 6
```

If the "check starting hour" is the same as the "check stop hour," then it starts each day at that time and runs as long as necessary to finish all files that meet the criteria.

You can also set a parameter at the server that specifies the period of time the file has not been accessed before the system compresses it by typing:

```
SET DAYS UNTOUCHED BEFORE COMPRESSION = 7
```

The default setting is seven days.

You can control the minimum space savings that must be accomplished before the compressed file is saved. If the operating system cannot gain a given amount of space from the compression of a file, it does not compress the file. The default is 2 percent.

```
SET MINIMUM COMPRESSION PERCENTAGE GAIN = 2
```

Decompression is automatic when the user requests a compressed file. The algorithms provided are extremely fast and do not affect either the client's or the server's performance. By default, compression runs as a background process. You can also control the decompression options when the compressed file is accessed by the users. The default setting is 1 but the range is (0 to 2).

```
SET CONVERT COMPRESSION TO UNCOMPRESSED OPTION = 1
```

0 means that the operating system will always leave compressed.

1 means leave compressed if it has been read only once in the last seven days (or DAYS UNTOUCHED BEFORE COMPRESSION; see parameter above).

2 means always decompress.

NetWare 4.1 also enables you to control the percentage of disk space that must be present before the system attempts to decompress a file. The default is 10 percent.

```
SET DECOMPRESS PERCENT DISK SPACE FREE TO ALLOW COMMIT = 10
```

NetWare 4.1 provides you with new file attributes that define files and when they should be compressed. Chapter 13 shows a complete listing of all the file attributes and their meanings.

Read Ahead

The read ahead feature in NetWare 4.1 provides the requesting workstation faster response when it is reading files from the server. The operating system anticipates the read requests from the workstation and caches the next disk block of a file, providing a client with faster access.

For example, when the workstation makes a request for a file open, the operating system opens the file and assumes that the next request from the workstation is going be a read request. In anticipation of the read request, the read ahead feature caches (places into file cache) the first block of the file. The result is that when the request from the workstation comes in for the first block, the server already has the block available in cache. The workstation request is serviced immediately instead of waiting for hard disk access. As the workstation reads the information out of the block in cache, the operating system will fetch the next block of the file and place it in cache anticipating more read requests. When the workstation has read one-half of the current block in memory (or cache), the operating system determines that the next block is needed. Using this algorithm, the operating system calculates when the next blocks are needed and reads them from disk before the request is made.

This read ahead feature is a low-priority task, which will not be executed when the server is busy handling other processes. The greatest benefit of read ahead is when you are accessing optical media. A CD-ROM drive, for example, is not as fast as a hard drive. Read ahead pulls the next block into cache so that it is ready when the workstation needs it.

Data Migration

Data migration automatically transfers infrequently accessed files off the file server and places them onto another media such as read/write optical disk or tape. The migrated files are removed from the hard drive of the server but can be viewed in the directory listing. The benefit of data migration is that it saves more expensive online hard drive storage and moves data to less expensive, near-line optical or off-line tape storage.

Even though the file has been migrated, it is automatically brought back to the primary storage when a user requests the file. If the near-line media is not mounted, then the server displays a message indicating which media volume needs to be mounted. After the media is available, the migrated file is retrieved and copied back to the server's hard drive. This operation is transparent to the user. However, if the file is on a media that takes time to mount, there will be a delay noticed by the user. You can set a capacity threshold for the volume so that files are migrated (starting with the oldest unused files) until the threshold is met.

PRINT SERVICES IMPROVEMENTS

NetWare 4.1 has the capability of attaching and sharing printer devices that are attached directly to the server, workstation, or the network. Although these features are supported by NetWare 3, major changes have been made to the print services in NetWare 4.1. These improvements include:

- PSERVER supports 256 printers.

- RPRINTER has been replaced with NPRINTER. If the printer is attached to the server, then load NPRINTER.NLM. If the printer is attached to the workstation run NPRINTER.EXE. NPRINTER.EXE can be loaded multiple times on a single workstation to support multiple attached printers.

- NPRINTER.EXE runs on DOS, Windows, and OS/2.

- PSERVER runs only as an NLM. The PSERVER.EXE is no longer needed or supported because the PSERVER supports more printers and NPRINTER.EXE can be loaded multiple times to service multiple printers attached to a single workstation.

- PSERVER performance has been enhanced.

- Quick Setup option in PCONSOLE enables you to create new print objects, such as a print server, printers, and print queues, from a single screen. It then establishes the printing relationship between these new print objects.

> Printer is now an object in NDS and no longer an attribute of a print server. The user can now send the print job directly to the printer without having to know anything about print queues.

> Network users no longer need to be familiar with print servers or print queues; they can send their print jobs directly to the printer by specifying the printer name.

> You can configure the print queue polling time.

> There is now support for Macintosh and NFS clients.

> You can have an unlimited number of print job configurations.

> NetWare 4.1 print services now has Autoconnect for lost printer connections.

The improved functionality of the print services in NetWare 4.1 increases ease of use for each of your network users.

WORKSTATION/CLIENT IMPROVEMENTS

With the advent of NetWare 4.1, Novell improved its DOS workstation software by introducing the Virtual Loadable Module (VLM) architecture. The VLM architecture is based on a set of VLMs (or modules) that is loaded and managed by the VLM.EXE. The VLMs work as a DOS requester to share the DOS environment. The VLM.EXE is a terminate and stay resident (TSR), which manages the loading of the individual VLMs by placing them in either conventional memory, upper memory, or extended memory.

The VLM.EXE automatically detects the presence of upper and extended memory managers and tries to load most of the VLM modules into those memory regions. Although VLM.EXE tries to load its modules high, you can configure which VLM module gets loaded and in which memory space. Typically, you will want the VLMs loaded high because you can save conventional memory for your workstation applications.

The following additional features are now included in the VLM architecture:

> The system supports up to 50 connections to different file servers.

> An auto reconnect feature automatically reestablishes a lost server connection.

- ► Users can now log in and log out while running Microsoft Windows.

- ► Login scripts are supported when users log in while running Microsoft Windows.

- ► Packet Burst provides quicker server responses especially across a wide area network (WAN) link.

- ► LIP (Large Internet Packets) is automatically supported.

- ► Language support can be customized for the workstation.

- ► Packet Signatures are provided for increased security of client to server communication.

The VLM client architecture provides you with more flexibility in terms of adjusting the software parameters for performance and workstation memory. For more information on the VLM architecture, see Chapter 18.

For the first time, NetWare has completely folded the Macintosh into NetWare as a client. With a new product called MacNDS, the Macintosh receives a complete Novell IPX protocol stack. With this support the Macintosh can participate as a full NDS member.

OS/2 is also supported as a client, with the Requester software for OS/2 included with all the other client programs. In addition, the standard version of NetWare 4.1 allows an OS/2 machine to be a nondedicated NetWare server. This might appeal to customers with small branch networks that can't afford a full, dedicated NetWare server.

UnixWare has default NetWare client software built into the UnixWare operating system. What's new with NetWare 4.1 is the UnixWare-formatted electronic documentation.

SECURITY AND AUDITING IMPROVEMENTS

NetWare 4.1 security provides you with the solutions to build a trusted network using components such as client, server, and medium. The NetWare 4.1 network will be the first network operating system to receive the National Computer Security Council's C2 rating. It is currently being evaluated as an entire network, which will result in Red Book C2/E2 status rather than the less stringent Orange Book status of other systems.

NetWare 4.1 accomplishes this level of security by providing client authentication for the login process, NCP packet signature for communications. NDS objects and property rights, and auditing of server events.

Client Authentication

NetWare 4.1 and NetWare Directory Services use the RSA public and private key encryption technology to authenticate the network users. The workstation proves its identity during login and establishes the necessary authentication pieces necessary for background authentication. Once the users are logged in, NDS's background authentication is used to transparently provide access to other privileged network resources. This combination of NDS and authentication services means the user only has to provide a username and password once for servers in a single tree. The rights authorization process to the individual resources such as servers and printers occurs after background authentication has been completed. For more information relating to client authentication, review Chapter 10.

The net effect of a login and background authentication is that neither the passwords nor unencrypted keys are sent across the network. The connection established by the client is unique only for the current session and cannot be forged.

NCP Packet Signatures

NCP (NetWare Core Protocol) Packet Signature is a security feature that protects servers and clients when they communicate via NCP. NCP packet signatures secure the communication between the client and server by forcing each device to stamp each packet with an electronic signature. This process prevents unauthorized users from capturing packets on the wire and regenerating them to seize a user's session. Capturing and regenerating packets that are not from the original sender is known as packet forgery. NCP packet signatures secure the network against intruders.

In order to enable NCP packets at the server console type:

```
SET NCP PACKET SIGNATURE OPTION = 1
```

The default is 1.
The options are:

► 0 — turn packet signature off

► 1 — do packet signature only if client requires it

➤   2 — do packet signature if client can support it

➤   3 — packet signature is required

NDS Object and Property Rights

NetWare Directory Services provides access control to NDS objects and properties. The file system provides its own access; however, both NDS and file system access use the identity that was established during authentication. NetWare 4.1 provides the flexibility for users to manage other objects in the NDS tree. By contrast, NetWare 3 had only the Supervisor (or an operator) who could manage other objects. This access control feature reveals additional functionality for administering security and defining access to your users. For detailed information on security please refer to Chapter 13.

Auditing

Novell provides an auditing capability that allows an auditor to record file system or NDS events, which are stored in an audit log file. Auditing is accomplished through the AUDITCON utility. Some auditable events include:

➤   Changing a user's password

➤   Adding a user

➤   Deleting a user

➤   Creating files

➤   Deleting files

➤   Granting access rights to a user or other object

In order to perform auditing, you must first enable it on a NetWare 4.1 server or on a container independent of a specific server. After auditing is enabled, it can be delegated to another user. This user can then change the password and become an independent auditor. Auditing may best be used for accounts in which such audit trails are required by law or by some other governing body. For more details on auditing, refer to Chapter 2.

MEMORY IMPROVEMENTS

NetWare 4.1 is the first network operating system to offer memory protection as an option. You can create a secure domain in which to run the NetWare Loadable Modules and protect all other server operations from a failing module. If an NLM has proven its reliability, you can choose to run the module at a lower domain, which will increase its performance. The lower domain is not protected. Testing of such NLMs requires that they first be written to run in the secure domain.

NetWare 4.1 also manages the server memory more efficiently by allocating the memory from a single memory pool. Having a single memory pool enables the operating system to recover NLM resources more efficiently when they are unloaded. The NetWare 4.1 single memory pool performs all of the functions that five or more memory pools performed in previous versions of NetWare. These improvements in memory allow the server to load and unload server-based applications without running out of memory and prolong the server's total uptime.

WIDE AREA COMMUNICATION IMPROVEMENTS

NetWare 4.1 has made improvements in WAN communications in the areas of multiprotocol routing, TCP/IP transport protocol support, NetWare Link Services Protocol, packet burst, and large internet packets.

Multiprotocol Routing

NetWare 4.1 provides built-in multiprotocol routing services using the NetWare Multi Protocol Router (MPR). MPR is Novell's software bridge/router that runs on Intel-based personal computers. The bridge/router enables customers to connect remote offices using familiar transport protocols and PC technology. The NetWare MPR is ideal for connecting local area networks by routing and source-route bridging over leased lines, Frame Relay, X.25, and Point-to-Point (PPP). NetWare MPR also functions with third-party dedicated routers in a multirouter network.

NetWare/IP and TCP/IP Protocol Support

NetWare 4.1 can tightly integrate into existing TCP/IP environments using NetWare/IP. By installing NetWare/IP on NetWare 4.1 servers, customers can create an environment that supports both the TCP/IP and IPX transport protocols, or one that uses only TCP/IP.

NetWare/IP is ideal for wide area networks that already use TCP/IP. Many large companies with networks based on NetWare have standardized TCP/IP on their WANs.

These companies need efficient integration between TCP/IP and NetWare across their multiple geographic sites. NetWare/IP provides this integration and reduces the number of protocols required to run on a WAN.

NetWare Link Services Protocol

NetWare Link Services Protocol (NLSP) is a link state routing protocol for IPX. This routing technology — based on Open Shortest Path First (OSPF) — overcomes the limitations of the Routing Information Protocol (RIP) and the Service Advertising Protocol (SAP) in local area and wide area IPX networks.

NLSP improves overall network efficiency through better utilization of existing bandwidth. It substantially reduces the overhead traffic commonly generated by RIP and SAP, provides load sharing across redundant traffic commonly generated by RIP and SAP, and provides load sharing across redundant paths.

Packet Burst

NetWare 4.1 automatically supports packet burst, which improves data delivery over latent links. Packet burst technology eliminates the ping-pong effect of the send/return packets during communication. For example, the client sends a request via packet burst to the server. The server responds to the client with multiple packets that don't require a return receipt for every packet. The server accepts a return receipt for the entire transmission, thus reducing the total transmission time and minimizing the "chattiness" of the NCP protocol. Packet burst is particularly beneficial to communication across WANs in which line speeds can often result in bottleneck.

Large Internet Packets (LIP)

In previous versions of NetWare, data packets that were passed through a router were automatically sized to 512 bytes. This sizing was done regardless of whether the workstation, server, and router were able to receive larger packets.

NetWare 4.1 provides large packet capability through routers and over wide area links. The largest packet that both the workstation and server can support is negotiated and used for all communication transmissions. Having a large packet for communications reduces the total number of packets and increases the throughput of the network. NetWare 4.1 servers also use LIP for server-to-server communications.

In order to enable LIP type:

```
SET ALLOW LIP = ON
```

CONSULTING EXPERIENCE

When a workstation first attaches to a server, both the workstation and the server negotiate a physical packet size that they will use for communications. If the server and workstation are on the same token ring segment, they will establish a physical packet size of 4202 (the maximum size used for token ring). In previous versions of NetWare, when the workstation had to go through a router to reach the server, the server did not know what type of network topology was on the other side of the router. The server had to default to the least common denominator — ARCnet with 512-byte packets. Today, ARCnet is seldom used.

With Large Internet Packets, this 512-byte ARCnet limit is not assumed, and the workstation and server negotiate their packet size at 1514 bytes for Ethernet or 4202 bytes for token ring.

MESSAGING AND E-MAIL

NetWare 4.1 now comes with NetWare Message Handling Service (MHS) built in. MHS provides message storage and forward transfer services that can be tightly integrated with NDS. This messaging service may be configured to run on any NetWare 4.1 server and is the basis for building a fully interconnected message infrastructure for distributing e-mail. This infrastructure supports all of the popular e-mail programs such as Novell's GroupWise, Microsoft Mail, Lotus CC:Mail, or DaVinci.

NetWare 4.1 includes an e-mail solution called FirstMail, which is a basic NetWare MHS software package. The new FirstMail application provides basic functions for DOS e-mail users on the local network. It will support only the least demanding LAN e-mail users.

NETWORK MANAGEMENT

NetWare 4.1 provides powerful network and server management facilities. Novell has instrumented all of its core modules with built-in management logic, which enables

the system to more easily manage each component. NetWare 4.1 is remotely manageable by providing Simple Network Management Protocol (SNMP) alert information directly to management consoles. These management consoles range from Novell's NMS console, IBM's NetView to NetWare for SAA, or Hewlett Packard's HP Open View.

OTHER NEW FEATURES

Additive Licensing

NetWare 4.1 provides you with increased flexibility to control the number of user connections supported by the system. Additive licensing enables you to add NetWare 4.1 licenses together on the server to support more connections. Additive licensing also lets you build the size of the network you need. The licenses increment in a range from 5 to 1,000.

For example, if you have a NetWare 4.1 500-user server in production and you need more connections, you can add 5 more connections totaling a 505-user server, or you can add 1,000 more connections totaling a 1,500-user server, or any combination in between.

Easier Installation and CD-ROM Support

NetWare 4.1 is distributed on two CD-ROMs for easy installation. In order to install NetWare 4.1 from a CD-ROM, make sure that you have a DOS partition (we recommend 15 to 30 megabytes), that DOS is installed, and that the CD-ROM drivers for DOS are loaded. You then simply insert the NetWare CD-ROM into the drive, type INSTALL, and the bulk of the work is done for you. You then have two options for installation: Simple or Custom Installation.

CONSULTING EXPERIENCE

With NetWare 4.1 you have a choice of Simple Installation or Custom Installation. Whichever one you choose depends on your configuration. Using the Simple Installation option will limit your hands-on involvement to about five minutes. Copying all the files from the CD-ROM to the server will take about one hour. Using the Custom Installation takes less than 10 minutes of hands-on time. But, if you are adding servers to networks

and want them in different places in the NDS tree or need multiple NDS trees, you will need to use Custom Installation.

Choose the Simple Installation if:

- Your network consists of fewer servers (one to three) and fewer than 1,000 users.

- All network resources and users will reside in a single NDS container (O=Organization).

- The hard disk has an existing 15MB (or larger) DOS partition.

- The server will boot from the hard disk.

- You will not use disk mirroring or disk duplexing (fault-tolerant features included with NetWare 4.1).

- Each disk will contain a single NetWare volume.

- IPX will be the only protocol used.

Choose Custom Installation if:

- Your network consists of more than a couple of servers and more than 1,000 users.

- Network resources and users will be placed in different NDS containers.

- The server could be booted from a floppy diskette.

- The hard disk will be partitioned (more than one NetWare volume per disk).

- You will use disk mirroring or disk duplexing.

- Volumes will span across multiple hard disks.

- You will use more protocols such as TCP/IP or AppleTalk.

Although the stated minimum hardware configuration for a NetWare 4.1 file server is a clone 386/33 with only 8MB of RAM and a 380MB SCSI, our experience has shown otherwise. We recommend a minimum 486/50 with 32MB of RAM and 300MB SCSI hard disk for implementing any tree with multiple NetWare 4.1 servers.

NetWare 4.1 ships on two CD-ROM disks: one for the operating system and one for the documentation, making it easier to perform subsequent installation and to access the online documentation.

You can take advantage of NetWare 4.1's CD-ROM support to simplify installation on the network. Once the product is installed on the local server, the installation files can be copied to the server's hard disk for subsequent installations or the CD-ROM can be mounted as a NetWare volume. This capability is referred to as server-to-server installation and is one of the major improvements in NetWare 4.1 installations.

NetWare 4.1 documentation can also be made available electronically by mounting the online documentation CD-ROM as a volume. Another option, if you have room on the server's hard disk, is to copy or install the documentation files onto the server.

CONSULTING EXPERIENCE

In order to mount your CD-ROM drive as a NetWare volume, you need to have a full SCSI controller for the CD-ROM. If your CD-ROM is nonstandard SCSI, such as the new IDE combination controllers for hard disks and CD-ROM, the CD-ROM drive will work fine for installation but cannot be configured later as a NetWare volume.

International Language Support

NetWare 4.1 ships with English, Spanish, German, French, and Italian language support. On the workstation, you set a DOS environment variable NWLANGUAGE = xxx, where xxx is the desired language. The result is that the NetWare utilities, menus, messages, and help screens will be in the proper language.

A new command, LANGUAGE, can be set on the server console to select the proper language at the server. All the Novell server utilities will be viewed in the selected language. You can also run the installation program in the proper language by choosing the correct language subdirectory before invoking the INSTALL.EXE program.

CHAPTER 2

▶ ·

Installation and Management Utilities for NetWare 4.1

"The question of common sense is always 'What is it good for?' — a question which would abolish the rose and be answered triumphantly by the cabbage." J. R. Lowell

The NetWare 4.1 utilities help you install and manage NetWare 4.1. Your job as an administrator is made easier by being familiar with the utilities and their functions. This chapter will focus on the utilities that are new to NetWare 4.1 and will not attempt to cover utilities that were also available from previous versions of NetWare. For information on NetWare 3 utilities also being used in NetWare 4.1 refer to Novell's NetWare 4.1 documentation.

For reference, the NetWare 4.1 utilities are separated into two categories: Server utilities and the workstation utilities.

Server Utilities

Server utilities enable you to change server parameters, monitor the load on the server, and control its allocation of resources. The following list shows all the NetWare 4.1 server utilities. The menu-driven server utilities that are new to NetWare 4.1 are highlighted in bold. All of the server utilities can be run from the NetWare 4.1 server console using the LOAD <utility name> command. The server utilities include both the menu-driven utilities and the server console commands. In this chapter, we will focus mainly on the menu driven utilities. For a complete description of the server console commands, see Appendix B.

- **DSMAINT** (available on NetWire)

- **DSMERGE**

- **DSREPAIR**

- INSTALL

- MONITOR

- **NETSYNC**

- PSERVER

- SERVMAN

Workstation Utilities

Workstation utilities for NetWare 4.1 are generally stored on a NetWare 4.1 server and run from a DOS, Windows, or OS/2 workstation. NetWare provides several graphical utilities to assist you in managing the NetWare 4.1 environment. The NWADMIN.EXE NetWare Administrator utility enables you to manage your NDS tree by creating, moving, deleting, changing, and renaming objects and containers. The DOS equivalent to the graphics NWADMIN utility is the menu-driven NETADMIN utility.

Another utility new to NetWare 4.1 is the NetWare User Tool called NWUSER.EXE, which is available in both Windows and OS/2. Using this utility your users can display and modify or add to their workstation's drive mappings, server attachments, print queues, and so on.

Table 2.1 lists all the NetWare 4.1 workstation utilities, which include not only the command line utilities for DOS and OS/2 but also the menu-driven utilities for DOS and Windows. All of these utilities are executed from a workstation that is logged in to the NetWare 4.1 network. Some utilities (for example, CX and LOGIN) only require an attachment to a NetWare 4.1 server. The utilities that are new to NetWare 4 are highlighted in bold.

INSTALLING THE NETWARE 4.1 SERVER

The first utility that you need to become familiar with is the NetWare 4.1 INSTALL.EXE installation program. This installation program loads the server INSTALL.NLM program. Both of these installation programs are found on the NetWare 4.1 CD-ROM.

However, before you perform the installation of NetWare 4.1 hardware and software you should plan or design the NDS tree. We have found that well-designed trees, including partitions, replicas, and time synchronization design, are much easier to manage and support. The design decisions you make before installation will affect how the users interact with the network resources.

T A B L E   2.1

All the NetWare 4.1 workstation utilities. The utilities new to NetWare 4.1 are highlighted in bold.

DOS	OS/2	WINDOWS	MENU
ATOTAL	ATOTAL	**NWADMIN**	**AUDITCON**
CAPTURE	CAPTURE	**NWUSER**	COLORPAL
CX	**CX**	**NAL**	FILER
FLAG	FLAG		**NETADMIN**
LOGIN	LOGIN		**NETUSER**
LOGOUT	LOGOUT		**NPRINTER**
MAP	MAP		**PARTMGR**
NCOPY	NCOPY		PCONSOLE
NCUPDATE	**NCUPDATE**		PRINTCON
NDIR	NDIR		PRINTDEF
NLIST	**NLIST**		
NMENU			
NPATH	**NPATH**		
NPRINT	NPRINT		
NPRINTER	**NPRINTER**		
NVER	NVER		
NWXTRACT	**NWXTRACT**		
PSC	PSC		
PURGE	PURGE		
RCONSOLE			
RENDIR			
RIGHTS	RIGHTS		
SEND	SEND		
SETPASS	SETPASS		
SETTTS	SETTTS		
SYSTIME	SYSTIME		
UIMPORT	**UIMPORT**		
WHOAMI	WHOAMI		
WSUPDATE			
WSUPGRD			

The NDS tree is a representation of the network resources in your company. You should think through how your users will access the network resources and design your NDS tree accordingly. Although you can move the NDS objects after the tree is installed, a good tree design will reduce the number of changes. For more information on NDS tree design, refer to Chapter 5.

Server Hardware Requirements for NetWare 4.1

The minimum hardware recommendations given in the Novell manuals are the absolute bare minimum to install NetWare 4.1.

Novell recommends the following minimum hardware:

- A PC with a 386 or 486 (SX or DX) processor

- 8MB RAM

- 90MB of hard disk space on SYS for the NetWare 4.1 Directory Services and utilities

- A CD-ROM drive

- 3.5-inch drive

- One network board

- Low-cost VGA monitor and card

- Network cabling

CONSULTING EXPERIENCE

Our recommendations for the minimum hardware for NetWare 4.1 are:

- A PC with at least a 486/66 Mhz processor

- 32MB of RAM

(continued)

- 500MB of hard disk space on SYS for NetWare 4.1 Directory Services and utilities

- 30MB partition for DOS

- Quad speed CD-ROM drive

- 3.5-inch drive

- One network card that is at least a 16-bit card

- Low-cost VGA monitor and card

- Network cabling
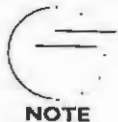
NOTE

For more exact calculations on RAM requirements for your system you can refer to the memory calculation in Chapter 11.

NetWare 4.1 Simplified Installation

The following step-by-step instructions are for installing a new NetWare 4.1 server using the simplified installation option. This section assumes that you intend to boot the NetWare server from the DOS partition and install the software from a CD-ROM.

1 • Create a minimum 30MB DOS partition on the machine that will be the NetWare 4.1 server. Refer to your DOS manual for detailed instructions on creating a DOS partition.

2 • Insert the NetWare 4.1 CD-ROM into a CD-ROM drive installed as a DOS device or a network device. For information on setting up your CD-ROM refer to your Novell or CD-ROM vendor documentation.

3 • Change to the root of the CD-ROM drive and run the NetWare 4.1 INSTALL program by typing **INSTALL** at the root.

4 • Choose NetWare Server Installation from the main menu as shown in Figure 2.1 below.

FIGURE 2.1

*NetWare 4.1 Server
Installation main menu
screen*

NetWare Install   NetWare Install

```
Select the type of installation desired
NetWare Server Installation
DOS/Windows Client Installation
Create Diskettes
```
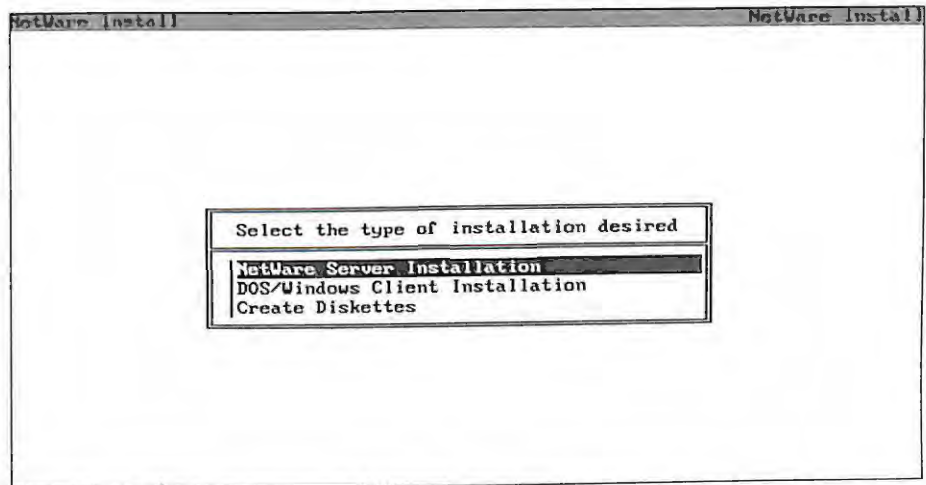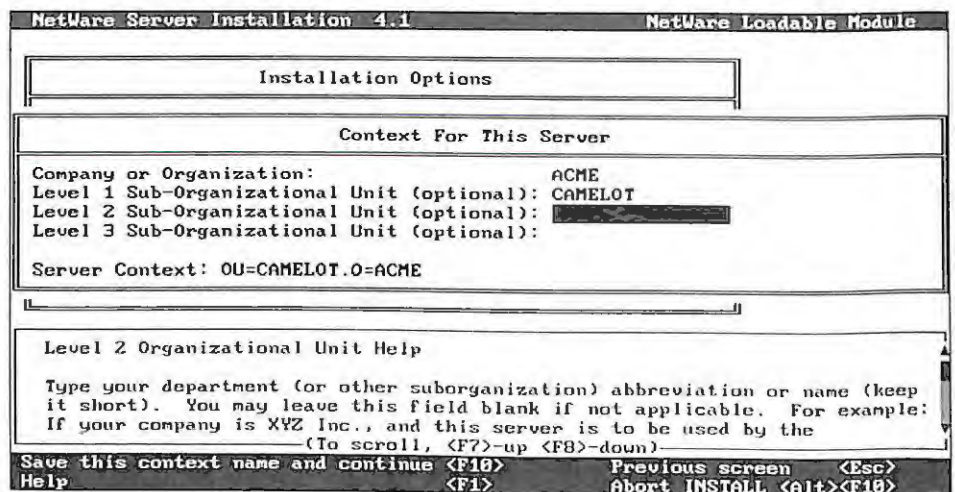
5 • Select Install New NetWare 4.1 Server from the installation screen, and then select Simple Installation of NetWare 4.1 from the installation screen. This section will review the Simple Installation method.

6 • Choose the server disk drivers. Select a driver that corresponds to the disk controller in your server hardware. If the required driver is not in the list, press <INS>, insert the appropriate diskette that contains the required disk driver, and select the driver.

7 • Select the appropriate parameters for your hardware configuration. You can usually select the hardware parameters from the installation menu unless the hardware settings on your hard disk controller have been changed from the manufacturer's default. If the hardware settings have been changed, please refer to your hard disk controller documentation for more information.

8 • After you have finished selecting your disk driver, select "No" to the prompt "Do you want to select an additional driver?"

9 • Choose the server network or LAN drivers that correspond to the network card in your server and press Enter. Verify the parameters to ensure they match the hardware in your server. You may need to make modifications as necessary.

10 • Install the Server License file by inserting the license diskette that came with the NetWare 4.1 operating system in drive A.

11 • The preliminary file copy for the NetWare 4.1 boot and startup files will now be copied.

12 • Install NDS. You would typically design or plan the NDS tree before installing the server. However, the simplified option automatically makes the decision that all the resources will be installed to a single O=Organization. This means that you do not have to plan the placement of the server in the tree.

Figure 2.2 shows the installation screen option for NDS.

FIGURE 2.2

NetWare Directory Services installation main screen

```
NetWare Server Installation  4.1                    NetWare Loadable Module
┌────────────────────────────────────────────────────────────────────────┐
│                         Installation Options                              │
├──────────────────────────────────────────────────────────────────────────┤
│                      Context For This Server                              │
│                                                                           │
│ Company or Organization:                      ACME                        │
│ Level 1 Sub-Organizational Unit (optional):   CAMELOT                     │
│ Level 2 Sub-Organizational Unit (optional):   ▓▓▓▓▓▓▓▓▓▓▓                 │
│ Level 3 Sub-Organizational Unit (optional):                               │
│                                                                           │
│ Server Context: OU=CAMELOT.O=ACME                                         │
│                                                                           │
├──────────────────────────────────────────────────────────────────────────┤
│  Level 2 Organizational Unit Help                                         │
│                                                                           │
│  Type your department (or other suborganization) abbreviation or name (keep│
│  it short).  You may leave this field blank if not applicable.  For example:│
│  If your company is XYZ Inc., and this server is to be used by the         │
│                  ──(To scroll, <F7>-up <F8>-down)──                       │
│ Save this context name and continue <F10>         Previous screen  <Esc>  │
│ Help                                 <F1>          Abort INSTALL <Alt><F10>│
└──────────────────────────────────────────────────────────────────────────┘
```

If this is the first NetWare 4.1 server on the network and you are installing a new NDS tree, follow the directions in step 13.

If this server is not the first NetWare 4.1 server and an NDS tree already exists, then skip the next steps and move to step 19.

13 • To install the First NetWare 4.1 server, choose the menu item "Is this the first NetWare 4 server?" (You will only see this item if there are no other trees advertising on your network.)

14 • Choose the time zone parameters by using the arrow keys to select the time zone where this server will be installed. You will be presented with another time zone menu for a Daylight Savings Time default.

15 • Next, enter the name of your organization in the field marked Organization Name. You can abbreviate the name as necessary.

16 • Enter the ADMIN password. You'll need to retype the ADMIN password.

17 • NDS installation should now be complete.

18 • Skip to step 22.

19 • To install this server in an existing NDS tree, choose the tree name from the list of NDS trees. If you have only one NDS tree installed on the network, it will be the only name that appears. If no NDS trees are displayed, check the network connection.

20 • Next, choose a time zone by using the arrow keys to select the time zone where this server will be installed. You will then be presented with another time menu to accept Daylight Savings Time. You can accept the default here.

21 • Enter the ADMIN password. This is the password assigned to the user ADMIN that is created when the first server is installed into this tree. NDS should now be installed.

22 • When the NDS installation is complete, the main NetWare 4.1 files will be copied. This procedure will last approximately 15 to 25 minutes depending on the server hardware you are running on.

23 • Next, you have the option to install other options. Please refer to the appropriate documentation for more details.

24 • Exit to the server console and restart the server by typing **DOWN** and then **RESTART SERVER**. Your installation of NetWare 4.1 is now complete.

NOTE

You can use the Custom Installation method to control the server setup and parameters for the new NetWare 4.1 server. The Custom Installation method gives you the most flexibility to change the defaults set by the installation program. For example, you can change the volume block setting before placing the server into the NDS tree. For more information about a custom installation, refer to your Novell documentation.

For more information about migration strategies using INSTALL and MIGRATE.EXE, refer to Chapter 16.

NETWARE ADMINISTRATION (NWADMIN.EXE)

NetWare 4.1 consolidates many of the NDS administrative functions into a single, easy-to-use graphics utility that greatly reduces the time you spend administering the network. The NetWare Administration utility, or NWADMIN.EXE, is a Windows-based utility that enables you to make changes to the Directory as easy as a point and click of a mouse. This utility is used by system administrators to manage objects in the NDS tree. Figure 2.3 shows the main screen or window for the NWADMIN utility.

Because the NWADMIN.EXE is a Windows-based utility it enables you to graphically manage all objects and properties in the NDS tree. The NDS tree can be created, changed, and browsed by selecting or clicking on the individual NDS objects.

The NWADMIN utility is executed at a client workstation from Windows. With this utility you can view, create, move, delete, and assign rights to the individual objects in the NDS tree. Typically, the individuals using NWADMIN will be system administrators with Supervisor rights over their container objects.

TIP

You may limit access to the NWADMIN utility by moving the utility from the SYS:\PUBLIC subdirectory to another subdirectory that only the administrators have access to. Most of your users will not need to use this utility.

*The main screen or window
for the NWADMIN utility*
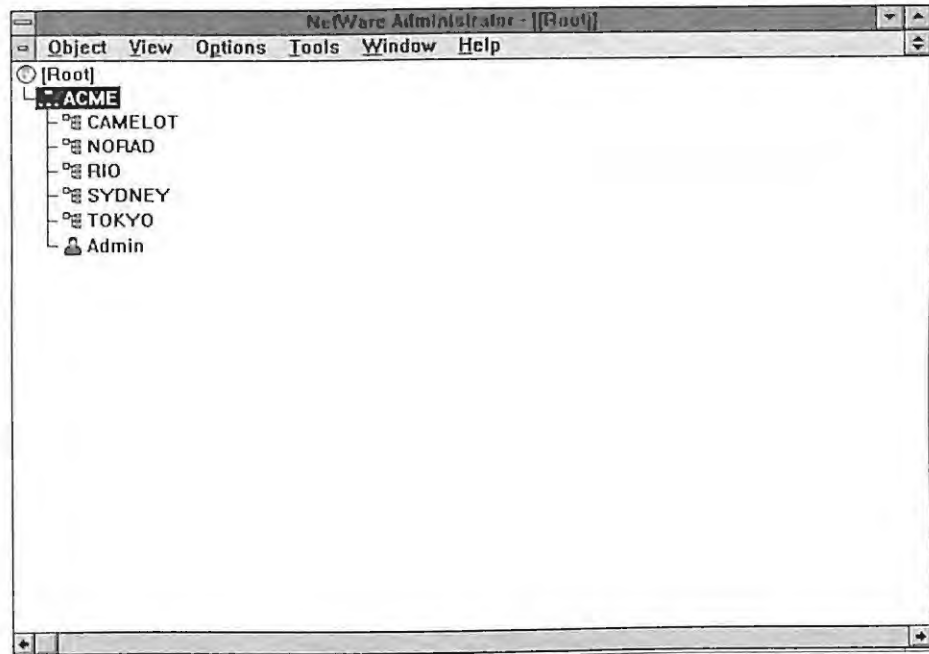
In order to execute this utility, you must have the NetWare DOS Requester or VLMs running on the workstation. Make sure that you have loaded the VLM client software or the NWADMIN utility will not load. For more information on client software, see Chapter 18.

You can use this utility to organize, manage, and browse the NDS objects that are contained in the tree by performing any of the following functions:

▶ Create and manage NDS objects.

▶ Assign rights in the NDS tree and file system.

▶ Set up the print services.

▶ Display or browse object and property information stored in the tree.

▶ Set up and manage the NDS partitions and replicas.

Using NWADMIN to Create and Manage NDS Objects

You can use NWADMIN to create, delete, modify, rename, move, and get detailed information about NDS objects. In order to create an NDS object you must first be logged in to the tree as an individual who has rights, such as the ADMIN (or other administrative) user at the container where you want to create the object.

For example, in order to create the user CN=GWASHINGTON in the container OU=RIO.O=ACME, move to that container in the tree and select the Create option from the Object menu. Figure 2.4 illustrates the New Object screen that is displayed after the Create option is selected.
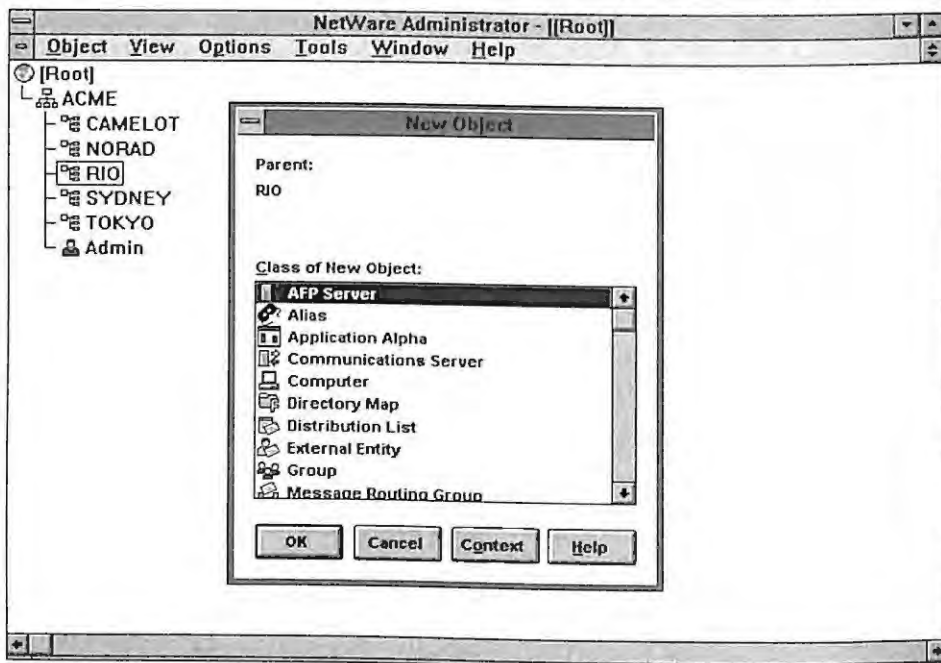
You can then select the User object from the list of objects, and you will be presented with the Detail information screen to fill out. You can access this screen any time for an object by right clicking (pressing the right mouse button) on the user object's icon. The Details dialog box appears that enables you to make changes. Figure 2.5 displays the details for the user GWASHINGTON in the container RIO.ACME.

*The Detail information
screen in NWADMIN for
the user
GWASHINGTON.RIO.
ACME*
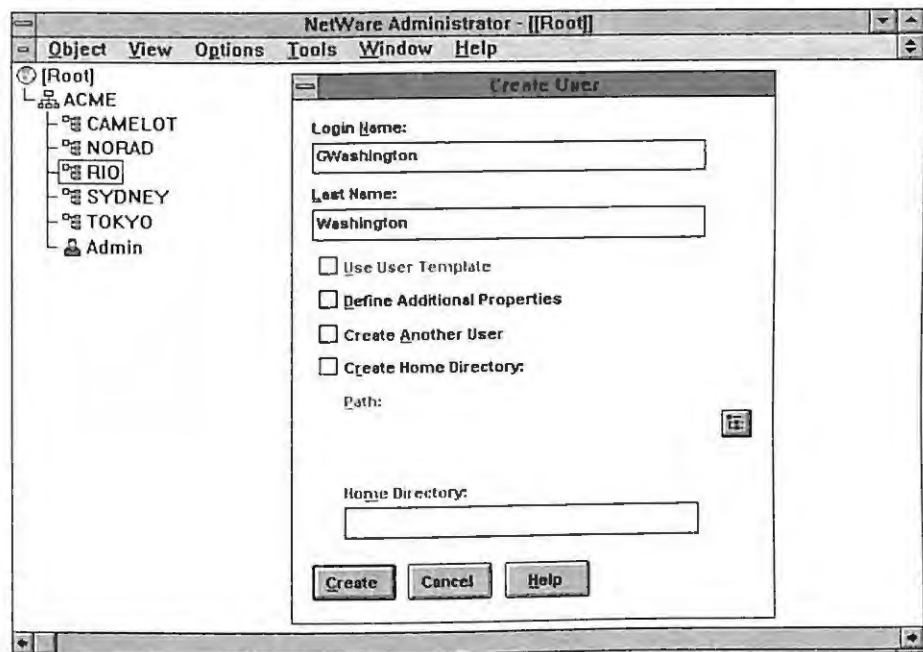
You can also use this utility to delete or rename any of the NDS objects. You can also move a selected NDS object from one location in the NDS tree to another. Figure 2.6 shows how the move operation for ALINCOLN might look. In this figure we are moving CN=ALINCOLN from the ADMIN.RIO.ACME to AUDIT.ADMIN.RIO.O=ACME.

Using NWADMIN to Assign Rights to the NDS Tree and File System

This object-oriented look of the utility enables you to make the assignment of rights much easier. Rights to users, groups, containers (OUs), files, and directories can be accomplished by simply dragging and dropping one icon onto the other. For example, when a user changes departments, you simply drag and drop the user to give that user the appropriate rights or move the user with the move command. Figure 2.7 shows the operation of dragging and dropping one object onto another to give it rights.

FIGURE 2.6

The NWADMIN screen for
moving CN=ALINCOLN
from the ADMIN.RIO.ACME
to AUDIT.ADMIN.RIO.ACME
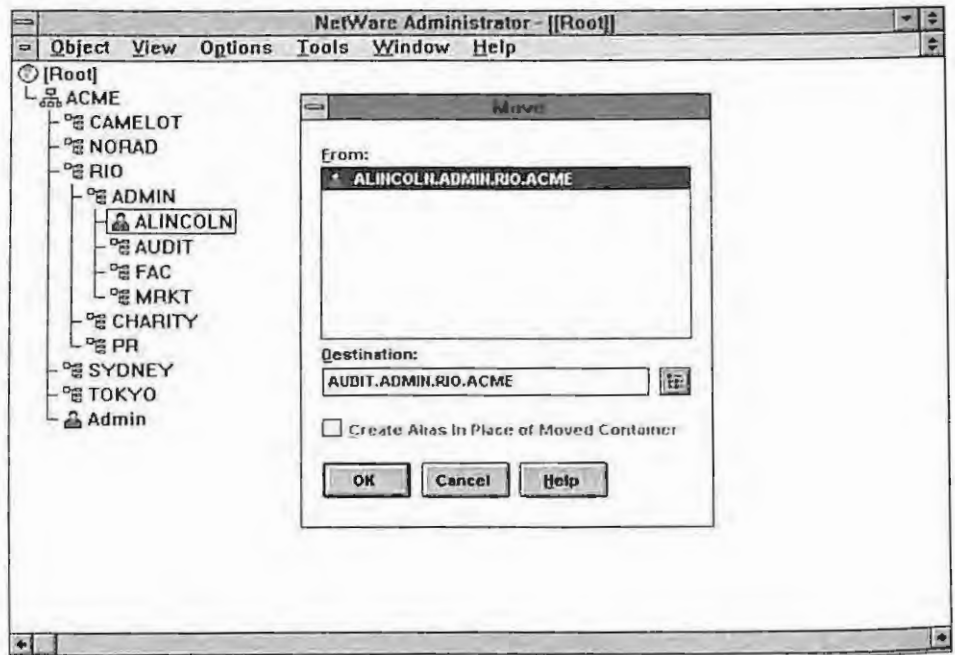
FIGURE 2.7

The NWADMIN screens
associated with the operation
of dragging and dropping
one object onto another to
give it rights
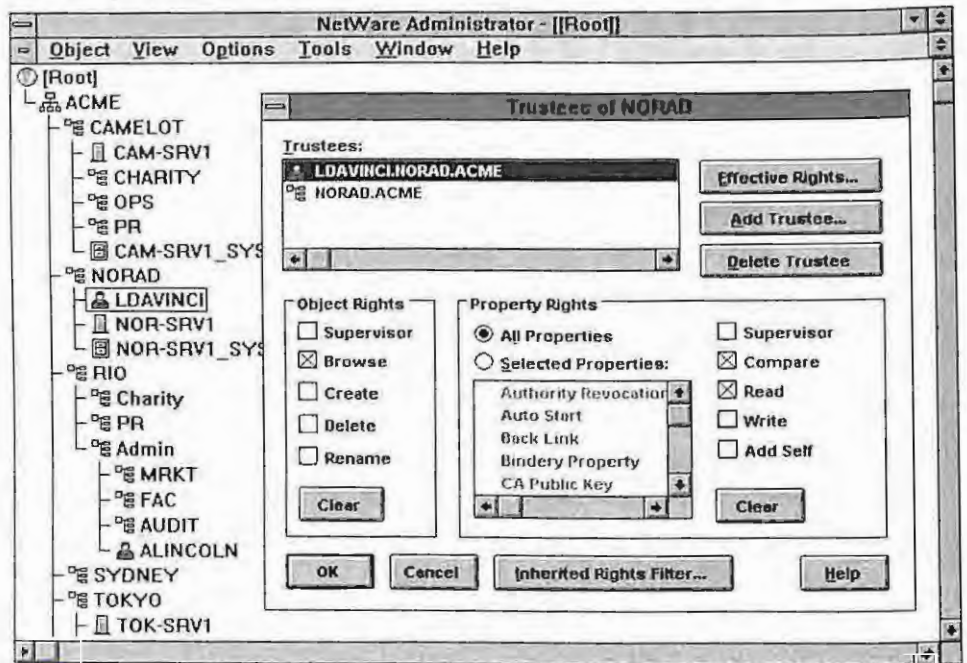
The operation of dragging and dropping is also used to assign rights to the files and directories in the file system. You can also enter into the Details screen for the object and assign file and directories rights in that screen. For example, in Figure 2.8, the NDS group object called CN=COMMON in the container ADMIN.RIO.ACME is granted rights to the COMMON subdirectory stored on the server CAM-SRV1.
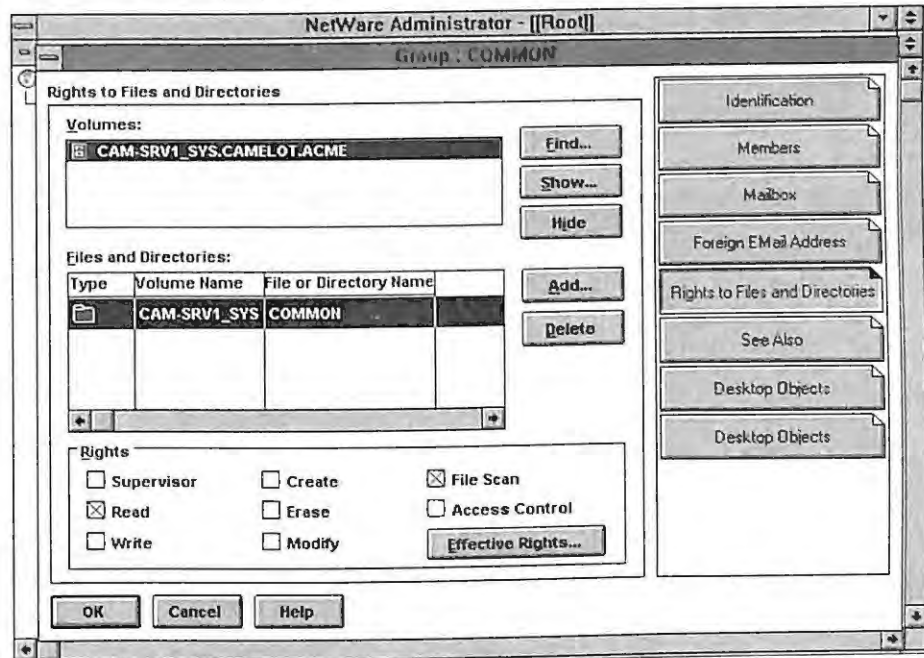
Adding file system directory or object rights in previous versions of NetWare used to require a multilayered menu utility and a tedious process of adding each user. Now all that is needed to add file system directory rights is to drag the user's icon over to the specific directory or object, and NetWare will simply ask you for a confirmation.

Because all the network resources in the NDS tree are represented by NWADMIN, managing multiple NetWare 4.1 servers is greatly simplified. Instead of managing a single server, you can focus your management efforts on a group of servers.

For more information on assigning rights, refer to Chapter 13.

Using NWADMIN to Set Up the Print Services

The NWADMIN utility can also be used to set up and manage the printing environment. NWADMIN is a functional equivalent to the PCONSOLE utility in DOS.

For example, in order to create a printer, select the desired container where the printer will be located. Select Create from the Option menu and then select Printer as the object to create. Figure 2.9 shows the dialog box that appears.
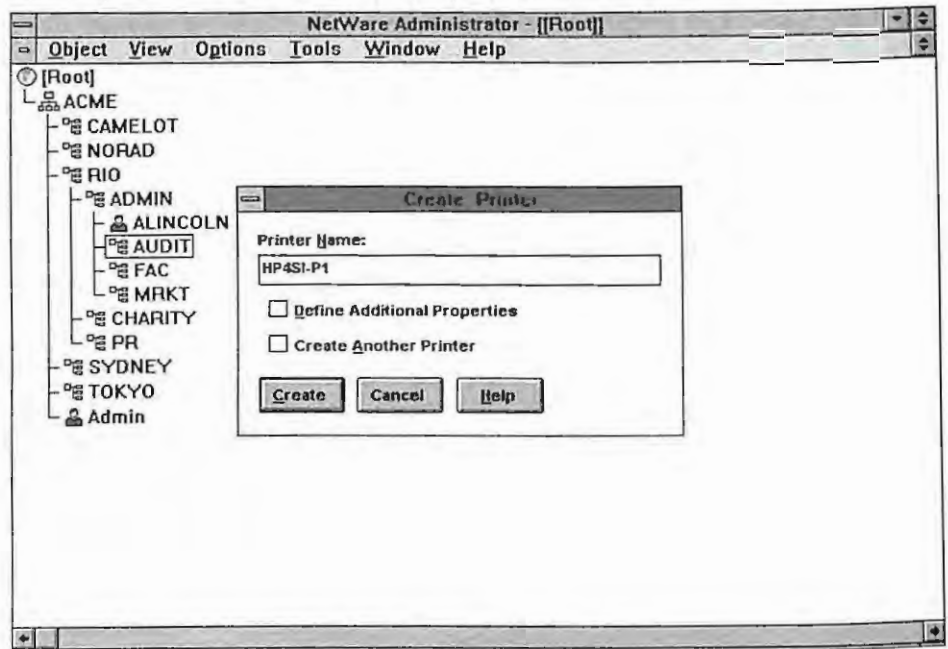
From the screen shown above, you can click on the Define Additional Properties button to define characteristics of the printer. If you select the additional properties, the Printer dialog box appears.

In order to create a print queue and print server object, you follow the same steps that you used to create the printer object. However, you select Print Queue and Print Server as the objects to create instead of Printer. The dialog boxes that are presented are very similar.

NOTE

For the print queues, NetWare 4.1 creates a subdirectory immediately under the root of the volume called QUEUES. Under this subdirectory, the operating system creates a numbered directory that matches the ID number of the queue. If the directory is created remotely it does not use the ID of the queue object at all.

Once all the corresponding print objects are created (printer, print queue, and print server), you can use NWADMIN to link them together. You perform this operation in a two-step process. First, you assign the print server to the print queue as shown in Figure 2.10. Second, you assign the printer to the print queue as shown in Figure 2.11.

FIGURE 2.10

Using NWADMIN to assign the print server to the print queue
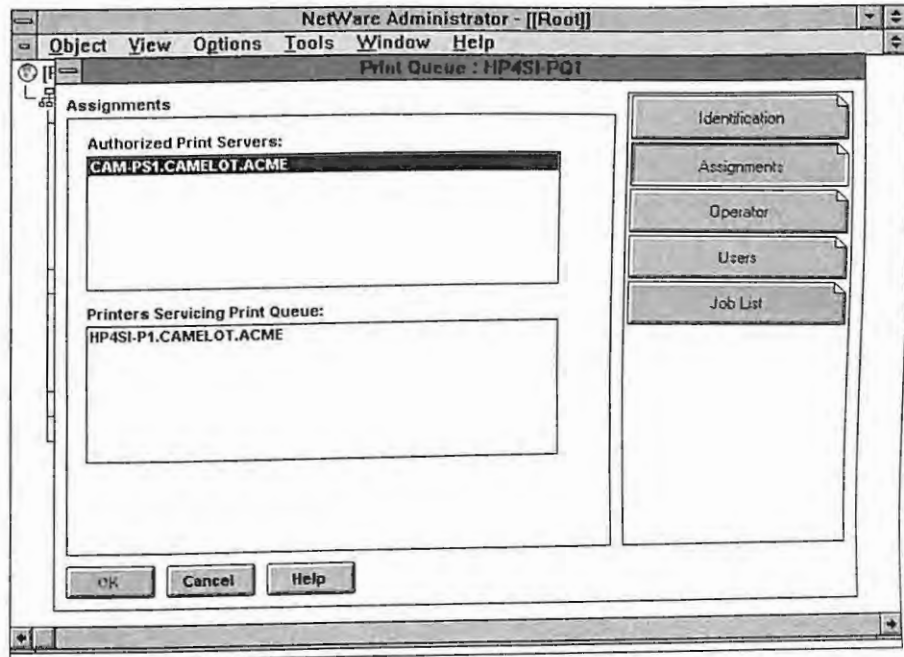
You can check the status of the printing setup using a new feature of NetWare 4.1, which displays the print layout. This feature is called Print Layout Page and is illustrated in Figure 2.12.

FIGURE 2.11

Using NWADMIN to assign the printer to the print queue
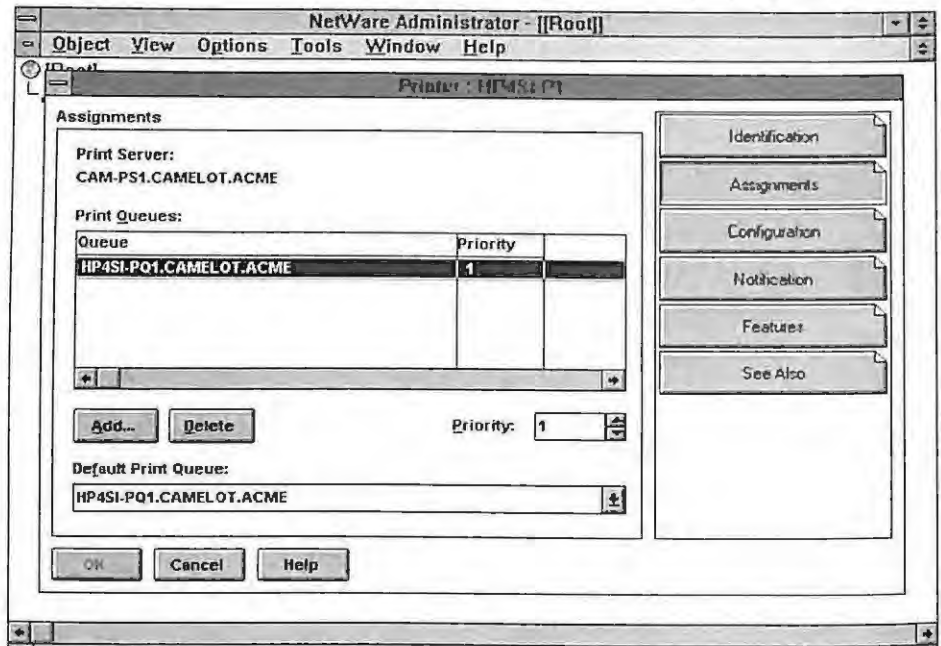
FIGURE 2.12

Using NWADMIN to display the Print Layout page
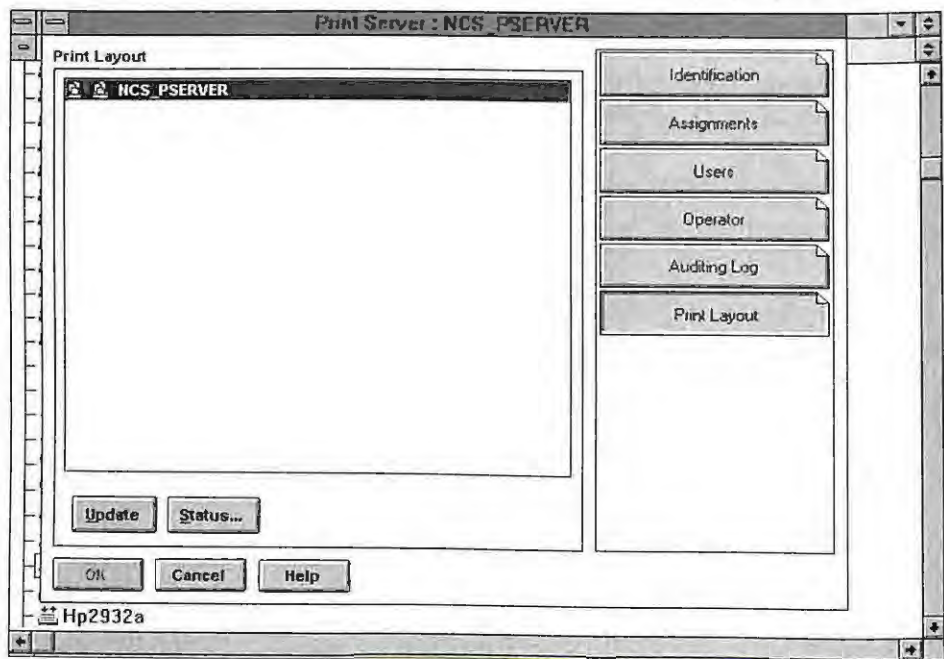
Using NWADMIN to Manage the NDS Partitions and Replicas

In order to manage your NDS partitions and replicas properly, there are several types of operations that you need to be able to perform. These operations are Create Partition, Merge Partition, Move Subtree, Add Replica, Remove Replicas, Change Replica Type, and Abort Partition. You can also delete a server object after removing the replicas from the tree. All these operations can be performed using the NWADMIN.EXE utility.

NOTE

Be sure to review Chapter 10 before performing any partitioning operations. Chapter 10 gives specific detailed instructions on how to check and verify partition operations and their states. It is important not to start an additional partition operation before the first partition operation has completed.

In the Tools menu of the NWADMIN.EXE utility, there is an item called Partition Manager. This option enables you to control partitioning and replica placement of the NDS database. Figure 2.13 shows you the Partition Manager option under Tools in the main menu of NWADMIN.

FIGURE 2.13

The Partition Manager option under Tools in the main menu of NWADMIN controls the partitioning and replica placement of the NDS database.
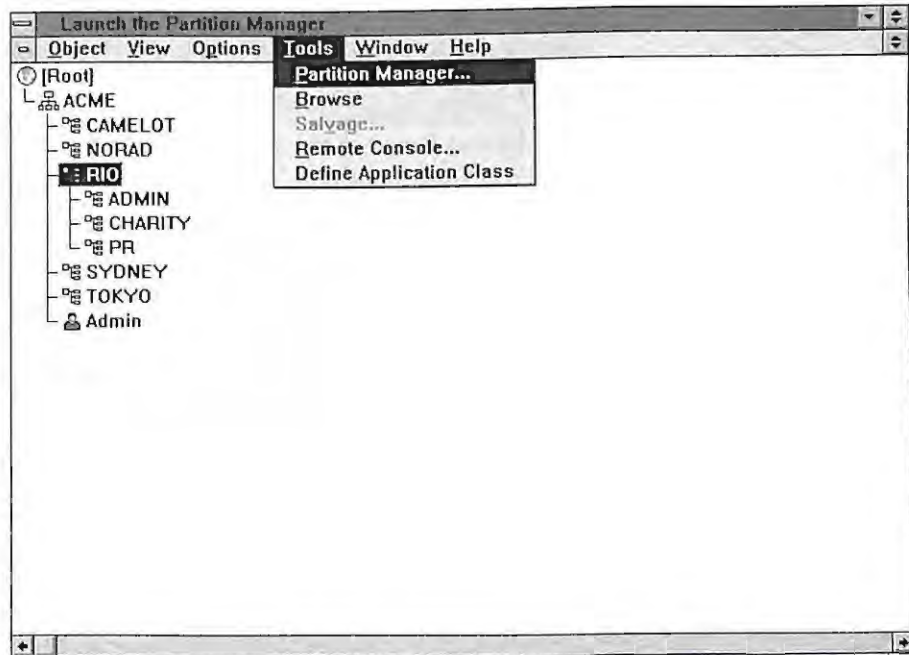
In Figure 2.14 you can see that the partitions are denoted with a small box icon to the left of the OU container object. In this example, the OU=NORAD and OU=CAMELOT are partitions.

FIGURE 2.14

Using the Partition Manager in NWADMIN, you can see the partitions denoted with a small box icon to the left of the OU container. OU=NORAD and OU=CAMELOT are shown as partitions.
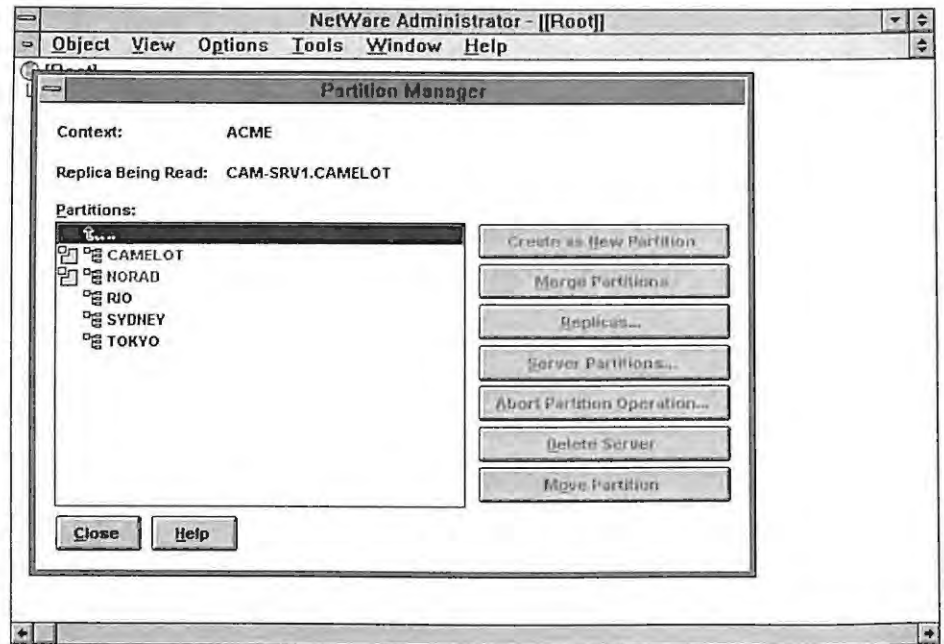
Using NWADMIN to Browse the NDS Tree

The NetWare Administration utility runs as a multiple-document interface application, which enables you to display up to nine different browsing windows at one time. The primary window illustrated in Figure 2.15 provides a background for viewing one or more secondary windows. The secondary windows support that browsing feature.

The browsers in NWADMIN are the secondary windows that appear in the utility. To open or enable a browser window, you need to select a container object from the primary window, and then select the Browse option from the Tools menu.

The new window is the browser window that displays the NDS objects from the selected container or current context. You can then select objects, directories, and files to perform the administrative tasks. Figure 2.16 shows an example of a browser window for the objects in the ACME tree.
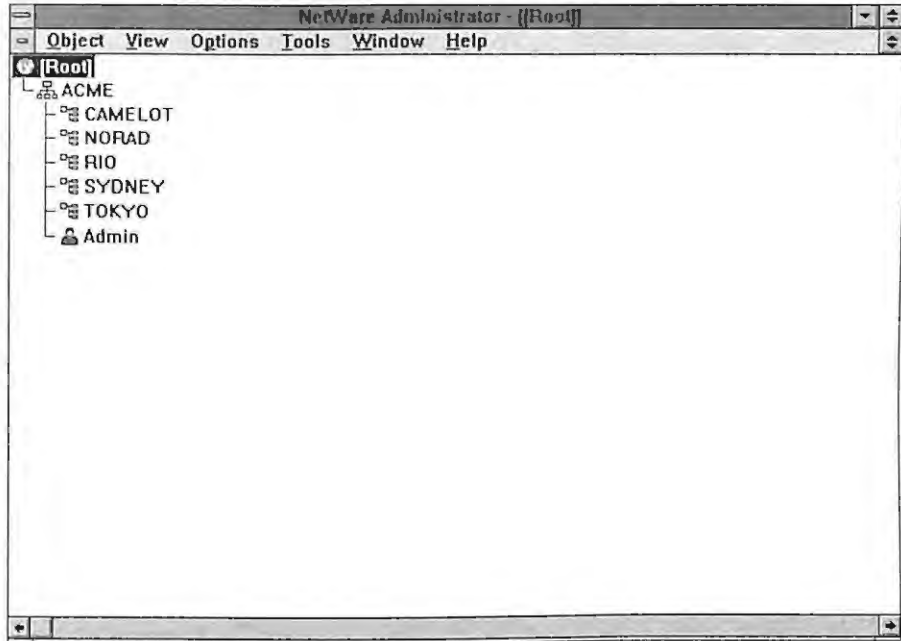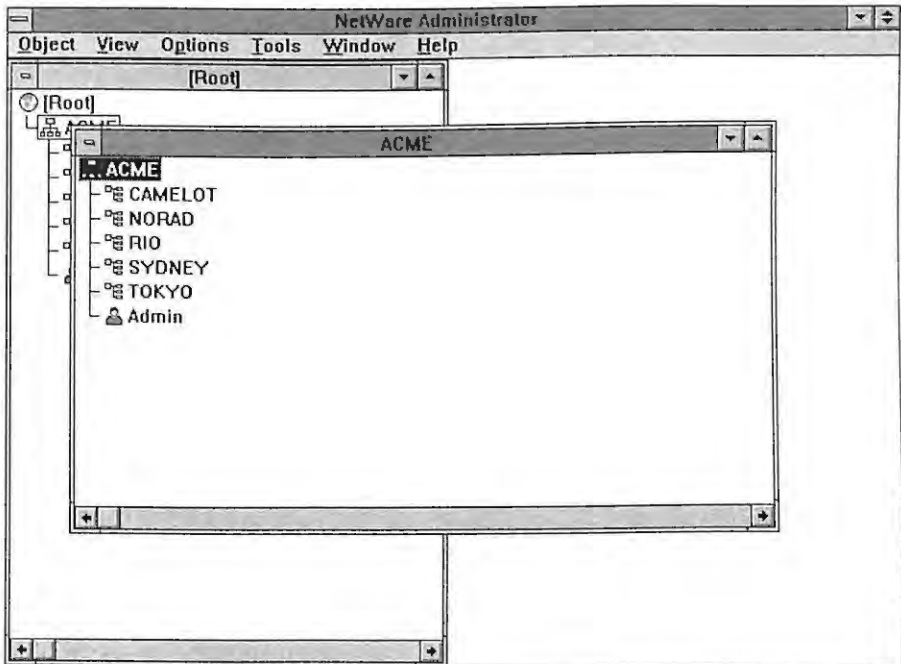
You can open up to nine browser windows, which enable you to view the individual container objects. To view multiple browser windows at the same time, select the Tile option from the Windows menu. Figure 2.17 shows three browser windows tiled for the objects in the ACME tree.

FIGURE 2.17

Three browser windows tiled for the objects in the ACME tree
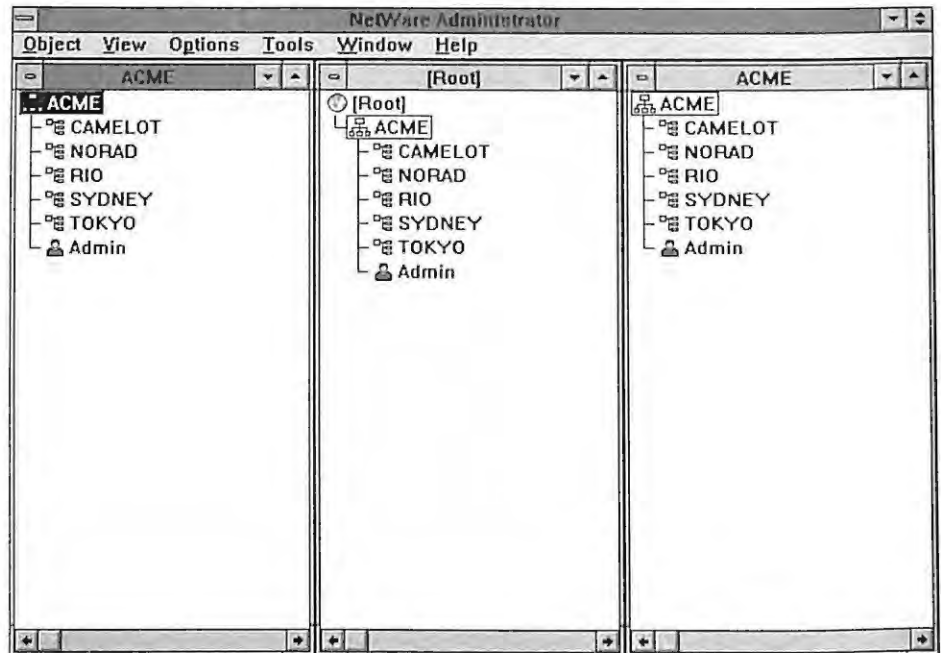
The title for each of the browser windows displays the context of the container where the browse windows is set. These will help you remember and organize the browser window on your screen.

Browse the NDS Tree Using the Container Objects in NWADMIN   You can browse the NDS tree by walking through the container objects. The container objects can be browsed by expanding or opening them and displaying their contents. There are several ways to expand the container objects within the NWADMIN utility:

▸ Double-click on the object. By double-clicking on the object the container will expand to show its contents and the subordinate objects. If the container is expanded, you can collapse the container by double-clicking again.

▶ Select a container object in the tree and choose the Expand option from the View menu to expand the contents of the container. You can collapse the container using the Collapse option from the same menu.

▶ Select a container object and press the "+" key on the keypad of the keyboard. The container object will expand and show its contents. (The "+" key above the "=" equal sign on your keyboard will not work for this operation.) You can collapse the container by pressing the "-" key on the keypad.

▶ Select a container object and press the right mouse button (or right click). Select the Browse option from the short menu that appears. This option launches a new browser window with the contents of the container object. Figure 2.18 shows the short menu that appears after you press the right mouse button.

FIGURE 2.18

A short menu appears when the right mouse button is pressed.
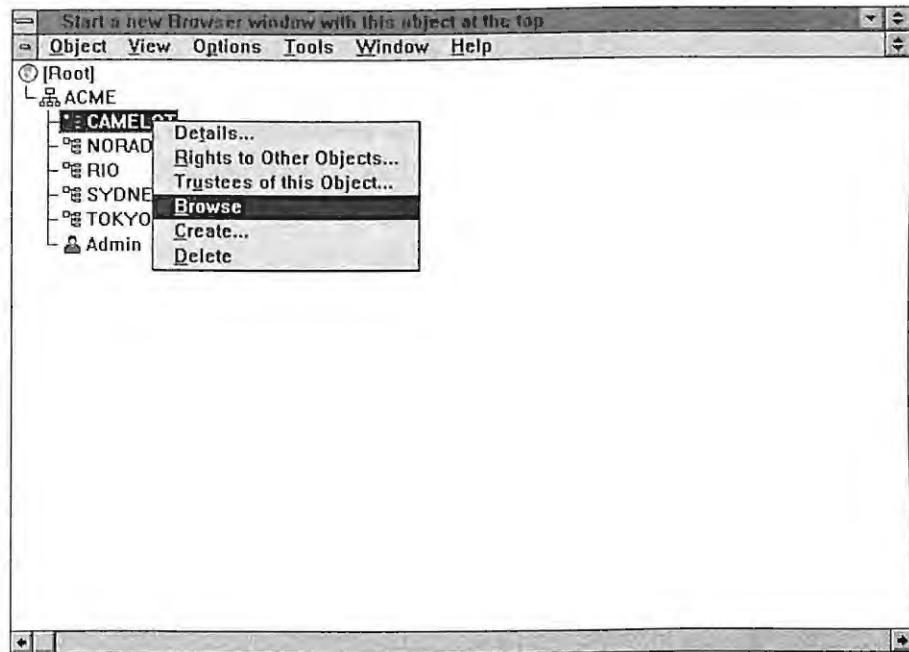
Figure 2.19 illustrates several container objects that have been expanded for the ACME tree.

FIGURE 2.19

Expanded container objects
for the ACME tree
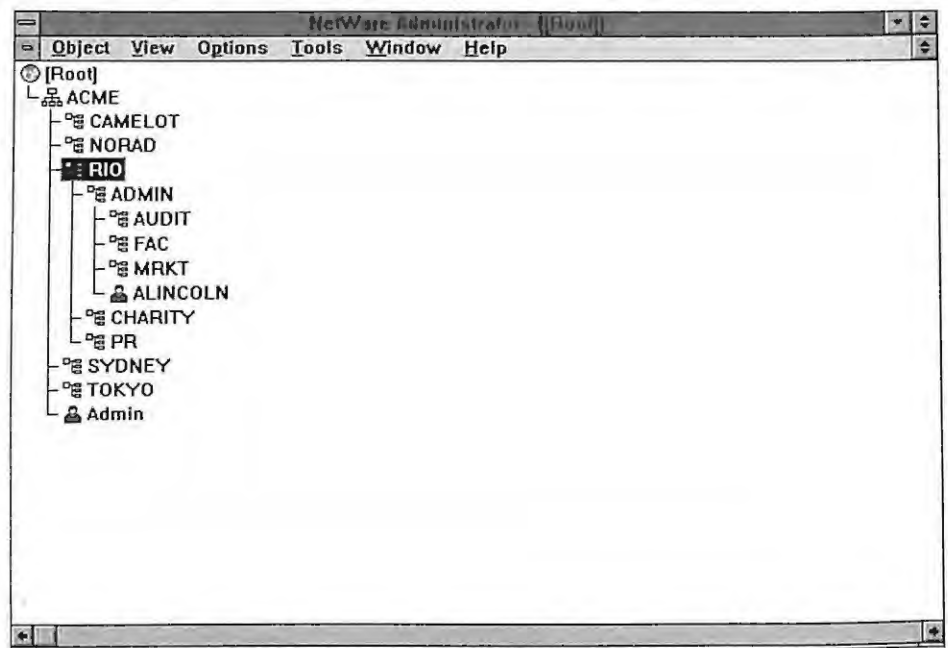
Browsing the Properties in NWADMIN   You can browse the properties for the individual object by opening the Object dialog box or selecting the Details option from the Object menu. You can open the Object dialog box using any of the following methods:

▸ Double-click on the object.

▸ Select the object and then choose the Details option from the Object menu.

▸ Select the object and press the right mouse button. Choose Details from the short menu that appears.

   The Object dialog box is organized or divided into pages that you select individually. The "Identification" page is the default or first page shown. Figure 2.20 shows the dialog screen for an object in the ACME tree.

FIGURE 2.20

The object or Details dialog screen for an object in the ACME tree

You can browse the specific information for an object by selecting the corresponding page. The page icons are located on the right-hand side of the Object dialog box and are specific to the type of object being displayed.

> **TIP**
>
> **The pages in the Object dialog box are all part of the same dialog box. When you select a different page you are still in the same dialog box. If you press OK or Cancel on any page you are affecting the entire dialog box, not just the individual page. Pressing OK will save modifications to all the pages and pressing Cancel will exit the dialog box without saving any changes to any page. In order to move between pages of the dialog box, select the desired page.**

Searching for Objects and Properties in NWADMIN You can find object and property information in the NDS tree using the Search feature. You can perform this function without having to expand each of the container objects. The search operation will check each of the objects in the tree unless you restrict or narrow the search criteria.

For example, in Figure 2.21, the search criteria is set up to find all the users in the ACME tree that have the department property equal to OU=CHARITY.

The search critera is set up to find all the users in the ACME tree that have the department property equal to OU=CHARITY.
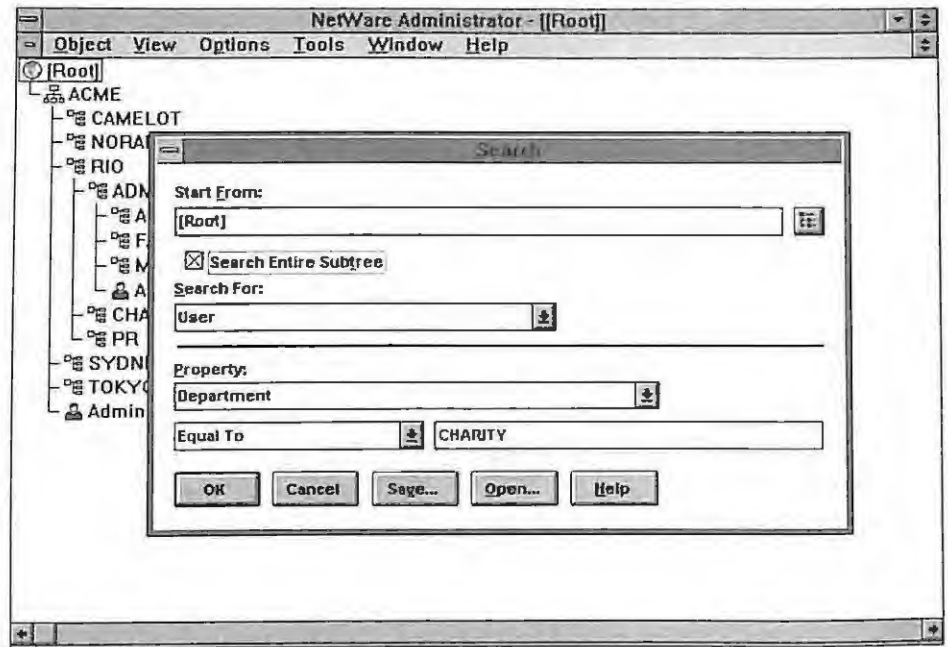
NETADMIN UTILITY

The NETADMIN.EXE is a DOS-based utility that enables you to manage NDS objects and properties in the Directory tree. With this utility you can view, create, move, delete, and assign rights to the individual objects in the NDS tree. This utility, like the NWADMIN utility, is intended to be used by the system administrators with access rights over their container objects. Figure 2.22 shows the main screen of the NETADMIN utility.

The main screen of the NETADMIN utility
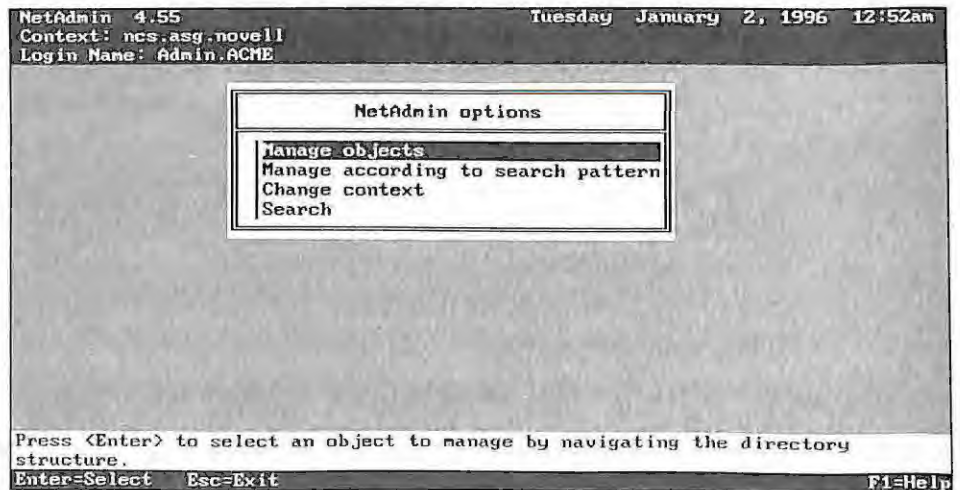
The NETADMIN utility has a browse screen that enables you to navigate the Directory tree and manage the objects. Figure 2.23 shows the browse screen for the NETADMIN utility for the ACME tree.
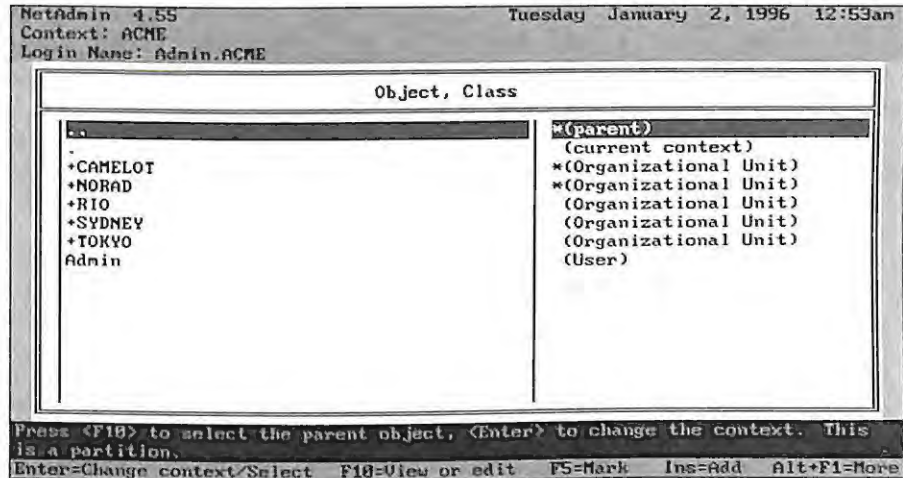
In order to browse down the NDS tree, you select the container objects and press Enter. The container objects are marked with a "+" character in front of the name. Objects without a "+" sign are leaf objects and do not contain other objects. In Figure 2.23 shown above, the CAMELOT, NORAD, RIO, SYNDEY, and TOKYO objects are the container objects. The user ADMIN is a leaf object.

To browse up the NDS tree to the parent container, select the ".." or parent marker. To view or edit the properties of the current container object select the "." or current container marker.

CX COMMAND LINE UTILITY

Change conteXt (CX) is a DOS-based utility that helps the user set the current context in the NDS tree. You can only use this utility to view the container hierarchy and leaf objects in the tree or to change to a new context. This utility is placed in the SYS:LOGIN subdirectory so that it can be executed before the users are logged in. Figure 2.24 displays the help screen for the CX utility.

```
CX                          Options Help                          4.20

Syntax:  CX [new context | /VER] [/R] [/[T | CONT] [/A]] [/C] [/?]

To:                                                        Use:
  View all container objects below the                      /T
    current or specified context.
  View container objects at the current                     /CONT
    or specified level.
  Modify /T or /CONT to view All objects                    /A
    at or below the context
  Change context or view objects relative to root           /R
  Display version information                                /VER
  Scroll continuously                                       /C

For example, to:                                     Type:
  View directory tree below the current context      CX /T
  View containers within a specific context          CX .O=Novell /CONT


>>> Enter = More    C = Continuous    Esc = Cancel
```

This section will not discuss how to set the context using CX, but will focus instead on the options available to you to browse the NDS tree. The browse options that display the tree information are:

- ▸ /CONT — Lists just the container objects at the current context

- ▸ /R — Lists objects relative to the [ROOT] object (This option also can change the context to the [ROOT].)

- ▸ /T — Lists all the container objects below the current context

- ▸ /A — Lists all the objects, including all the leaf objects

Figure 2.25 displays the output of the CX /CONT command for the ACME tree when the current context is set to [ROOT].

Figure 2.26 displays the output of the CX /T command for the ACME tree when the current context is set to [ROOT].

FIGURE 2.25

The output of the CX /CONT command for the ACME tree when the current context is set to [ROOT]

```
*** Directory Services Mapping ***

ACME
   |CAMELOT
   |NORAD
   |SYDNEY
   |RIO
   |TOKYO
C:\>
```

FIGURE 2.26

The output of the CX /T command for the ACME tree when the current context is set to [ROOT]

```
*** Directory Services Mapping ***

[ROOT]
   LACME
      |CAMELOT
          |CHARITY
          |PR
          |OPS
              |FIN
              |DIST
      |NORAD
          |CHARITY
          |PR
          |LABS
              |R&D
              |WHI
      |SYDNEY
          |CHARITY
          |PR
          |HR
              |MEDICAL
              |FOOD
              |SHELTER
      |RIO
>>> Enter = More    C = Continuous    Esc = Cancel
```

NLIST UTILITY

The NLIST utility in NetWare 4.1 enables you to browse the information in the NDS tree and replaces a number of NetWare 3 utilities, including USERLIST, SLIST, SECURITY, and others.

The NLIST utility enables any user to search and view information for most of the NDS objects such as the following:

▶ AFP Server

- Alias

- Computer

- Directory Map

- Group

- Organization

- Organizational Unit

- Print Queue

- Print Server

- Printer

- Profile

- Server

- User

- Volume

In order to search the Directory tree there are several options. Figure 2.27 displays one of the help screens for NLIST, which lists the display options for the objects in the tree.

The proper syntax must be entered for the NLIST utility to execute. The syntax is as follows:

```
NLIST [object type] [=object name] [/options]
```

Examples:

- NLIST User /R /S — Searches for all user objects in the tree starting at [ROOT]

▶ NLIST Server /B — Displays all the servers using bindery calls just like the
NetWare 3 SLIST command

```
NLIST                        General Help Screen                      4.19

Purpose: View information about users, groups and other objects.
Syntax: NLIST class type [property search option]
                         [display option] [basic option]

For details on:                          Type:
  Property search options                  NLIST /? R
  Properties                               NLIST /? P
  Display options                          NLIST /? D
  Basic options                            NLIST /? B
  All Help Screens                         NLIST /? ALL

Class types:
  * (all class types)       User            Print Queue
  Server                    Group           Printer
  Computer                  Volume          Print Server
  Directory Map             Profile         Organization
  Organizational Unit       Alias           AFP Server

Enclose in double quotes all class types or properties containing spaces.

C:\>
```

You can also use NLIST to search for and display information about the properties for
each of the objects. You will use the WHERE and SHOW options in the command line
to accomplish the browsing. For example, the syntax for the WHERE and SHOW options
for an object is as follows:

```
NLIST object class [=object name] [WHERE

[property]operator [value]]

[SHOW property[,property . . .]]
```

Examples:

▶ NLIST user=AEINSTEIN /D — Displays all properties for the user AEINSTEIN
in the ACME tree.

▶ NLIST user WHERE "Security Equal To" EQ CAMELOT_ADMIN — Shows all
the users that are security equivalent to the CAMELOT_ADMIN Organizational
Role relative to the current context in the ACME tree.

NWUSER AND NETUSER UTILITIES

The Windows-based NWUSER and the DOS-based NETUSER utilities enable the users to manage their own network environment and browse the NDS tree and its resources. Using these utilities the users can change their own context, manage server connections, manage the drive and search mappings, change passwords, change login scripts, capture printers and print queues, and send messages. Each user that has loaded at least the VLM client software will have installed these utilities automatically on his workstation.

Figure 2.28 illustrates the main screen for the Windows-based NWUSER utility.

FIGURE 2.28

The main screen for NWUSER utility
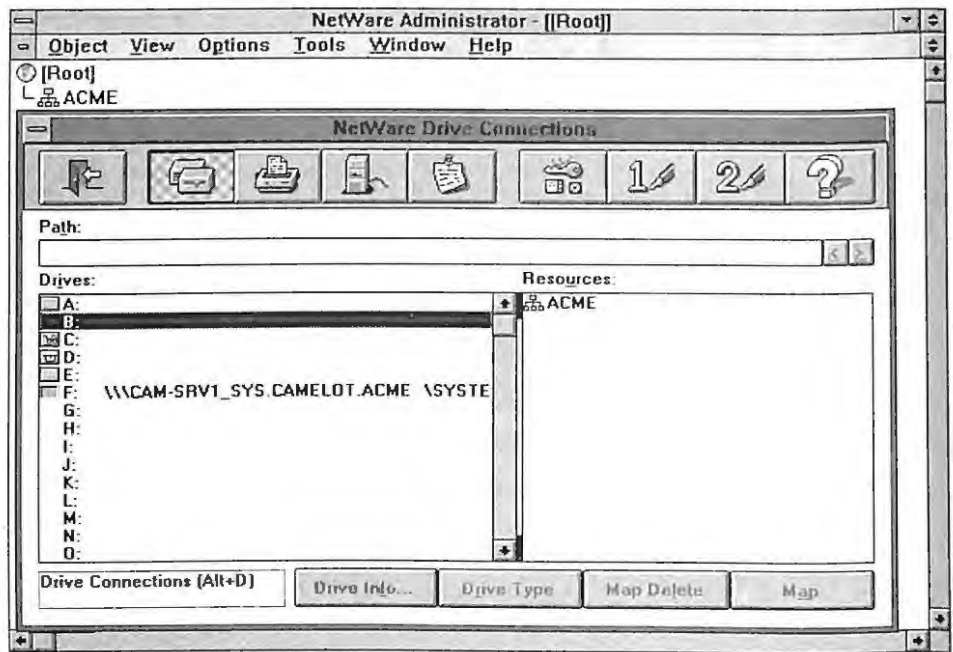
NETWARE APPLICATION MANAGER

The NetWare Application Manager (NAL) manages application objects at a user's desktop. The Application Manager currently consists of two components: an administration tool that snaps-in to NWADMIN and client software known as the NetWare Application Launcher.

The NAL enables users to run applications that were previously configured by a system administrator, and whose setup information is stored as an application object in the NetWare Directory tree. NAL displays icons for all available applications in a Windows

group, and users double-click an icon to launch particular applications. Users do not need to worry about drive mappings, paths, or rights; the administrator can manage the application launcher on a Container, Group, or User object level through the snap-in utility. This utility greatly simplifies the need for common applications and allows the user desktop configuration greater flexibility.

By creating application objects for your applications, you can manage which applications appear on the user's desktop. Using application objects simplifies the administrative tasks of assigning rights, customizing login scripts, and providing application support. Using NetWare Administrator, applications for an entire organization, group, or user can be centrally administered. By configuring executables as application objects, your administrators can perform upgrades and control versions of applications on the network very easily.

Assigning an application object to a container, group, or user object makes the application available to users who are trustees of the container, group, or user object. When launching the application, all network drive mappings, printer ports, and additional parameters are automatically configured. When the user exits the application, drive mappings and printer ports are returned to their original configuration.

Because application objects configure their required resources when launched, system and user scripts can be reduced. Application information and support contacts can be assigned to each application object. The application information is also available to users on their desktops.

For example, the following scenario illustrates the benefits of using the NetWare Application Manager utility. A NetWare administrator of a company with 300 employees installs a new application and wants to distribute it to the employees who need it. The application is a spreadsheet program that performs specific business macros unique to the company. The spreadsheet application requires no special drive mappings, printer ports, environment variables, or user configuration files.

Solution #1 Using Local Computer Resources

The administrator installs the application on each user's hard drive. This option is the most time-intensive; it requires the administrator to install, upgrade, and support the application and computer hardware at each user's location.

Solution #2 Using the Network

The administrator installs the spreadsheet on a NetWare server called APPS in the SYS:\APPS\SPREDSHT\V1.0 directory. Using the NWADMIN utility, the administrator

creates a group object called SPDSHEET GROUP. File system rights are granted to the group in the directory where the application is installed. Users requiring access to the application are made members of SPDSHEET GROUP. Though the application is now available, an icon representing the spreadsheet application may not appear on the desktop unless it is manually configured for each desktop. The administrator must either communicate that the application is available and explain where the application is located and how to set it up, or physically visit each desktop. This is a better solution but not the best because there are still a few problems:

► Even though the application icon may appear on the user's workstation, the application actually resides on the network. If the user is not logged in when he double-clicks an application icon, an error message will be displayed stating that the path is invalid.

► The user may delete the icon, then need the administrator's help to restore it.

► The icon references the executable file (quattro.exe) in the \\PRODUCTS\SYS\APPS\SPDSHEET\V1.0 directory. To move or rename the executable, or to upgrade to version 2.0 without removing version 1.0, the path must be changed at each workstation.

Solution #3 Using the NetWare Application Manager

The administrator installs the spreadsheet application on server APPS in the SYS\APPS\SPDSHEET\V1.0 directory, creates the SPDSHEET group, and adds as a member of the group each user who needs the application. Using NWADMIN, the administrator creates an application object called SPDSHEET. The new object contains the path to the program file and other information such as command-line parameters, the working directory, and a description of the application's purpose. The administrator associates the SPDSHEET application object with the SPDSHEET group. The SPDSHEET application icon appears on each user's workstation automatically. No setup is required at the user's workstation. The application may now be launched by double-clicking the icon.

Other benefits of the NetWare Application Manager are listed below:

► If the user is not logged in to the network, the icon is not displayed.

► The user cannot delete the icon or change any of the path information.

► If the administrator wants to move or rename the executable file, only the SPDSHEET object must be modified, and the user is unaware of any changes.

► The upgrade can be installed anywhere (that is, \\PRODUCTS\SYS\APPS\SPDSHEET\V2.0).Using the NWADMIN utility, version 1.0 can be replaced by modifying the path to the new application executable.

► Both versions can be available by creating a new SPRDSHT2 application object and associating it with SPDSHEET group.

PARTMGR UTILITY

The Partition Manager utility known as PARTMGR.EXE enables you to manage your NDS partitions and replicas. There are several types of operations that you need to understand and use. The NDS partition and replica operations are Create Partition, Merge Partition, Move Subtree, Add Replica, Remove Replicas, Change Replica Type, and Abort Partition. You can use the PARTMGR utility to perform the NDS partition and replica operations on the NDS tree. You can also use the Partition Manager option in NWADMIN to perform partition operations.

The PARTMGR.EXE utility is a DOS-based program that performs partition operations on the NDS database. The partition operations are create, merge, move subtree, add replicas, remove replicas, and change replica type. Figure 2.29 displays the main menu of the PARTMGR utility.

```
PartMgr  4.15                                    Tuesday  January  2, 1996  1:15am
Context:  ACME
Replica being read: CAM-SRV1.CAMELOT


                              ┌────────────────────────────┐
                              │  Partition Administration  │
                              ├────────────────────────────┤
                              │  Manage partitions         │
                              │  Change context            │
                              └────────────────────────────┘




 View, merge, or split partitions. View or modify replicas
 Enter=Select    Esc=Exit                                               F1=Help
```
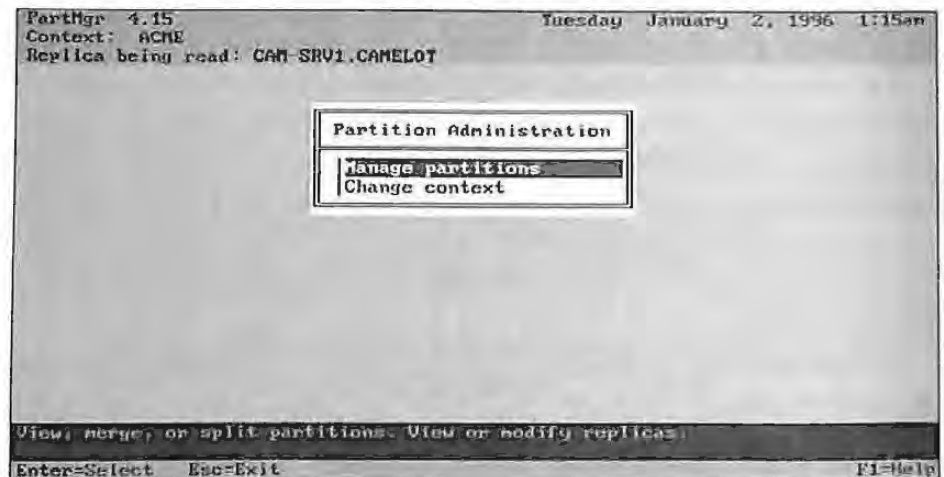
Refer to Chapters 6 and 11 for more information regarding partitioning and replication.

UIMPORT UTILITY

The UIMPORT utility enables you to import users from a database application to NDS. This utility can also create, delete, and update user objects and their existing properties. If you are using a database that has the capability to convert records to a comma-separated ASCII file, you can use UIMPORT to migrate this data to the NetWare Directory.

This utility is used primarily to create NDS user objects using records from another database, typically a database application that may contain lists of users. The process of using UIMPORT occurs in three phases:

1 • Create a data file based on your existing database.

2 • Create a control file to interpret and act on the data file.

3 • Import the records into NDS.

The UIMPORT utility consists of two files, the data file and the import control file. The creation of your data file is simply the generation of your ASCII comma-separated file from your previous database. The control file consists of some control parameters and field definitions to define where in NDS the information should be placed.

The Data File

The creation of a data file is accomplished by your database application when you save the information in a comma-separated ASCII file. A comma in the data file indicates a separation of fields in the NDS database. When a record is read through UIMPORT, a comma indicates a new property to be added in the user object. An example of the structure of your database might be as follows:

Last Name:
First Name:
Local Address
   Street:
   City:
   State of Country:
   Zip Code:
Description:
Job Title:

After you have created a data file in delimited ASCII format with any DOS editor you will have a file with records that look like the following:

```
Madison,James,"111  Calle  de  Carnival","Rio  de
Janeiro",Brazil,56665,Facilities,Administrator
```

We have named our file ACMEDATA as we exit and save from our DOS editor.

The Import Control File

The next file you need to create is the import control file. The import control file actually controls how the data file information will be written to NDS. The control file can be written using any DOS text editor. You will enter a set of control parameters first, followed by a list of field definitions. Control parameters define how the information in the data file is separated. The field parameters define how the information is to be written to NDS. Table 2.2 lists the control parameters and their functions.

TABLE 2.2

Control Parameters

CONTROL PARAMETER	EXPLANATION
Separator	Defines the type of separator used in the data file, such as a comma or a semicolon.
Quote	Defines the character used for string data in the data file.
Name Context	Defines the NDS context where the users will be created.
Replace Value	Enables you to overwrite or add data to multi-value fields, such as overwrite an existing telephone number in a user object.
User Template	Specifies the use of a user template in the creation of your NDS user objects.
Import Mode	Defines how user objects will be created. C = Create, B = Create and Update, and U = update data for existing objects.
Delete Property	Enables you to delete a property from a user object in NDS.
Create Home Directory	Enables the creation of a home directory for user objects.
Home Directory Path	Required if you create a home directory for users. The volume name is not necessary in the inclusion.
Home Directory Volume	Required if you create a home directory for users.

The field definitions define which fields should be used in NDS with the incoming data. The following fields can be selected based on your particular needs:

▶ Name

▶ Last Name

▶ Other Name

▶ Postal Address

- E-Mail Address

- Telephone

- Fax Number

- Job Title

- Description

- Department

- Location

- Group Membership

- See Also

- Skip

- Login Script

Using our previous data file as an example, we would have an import control file that could appear as the following created with any DOS editor utility (note that the Name Context is in the correct order):

Import Control

```
Name Context="O=ACME.OU=RIO.OU=ADMIN.OU=FAC"

User Template=y
```

Fields

```
Last Name

Other Name

Postal Address
```

```
Postal Address

Postal Address

Postal Address

Department

Job Title
```

We have named our file ACMECTRL as we exit and save from our DOS editor.

Importing Records into NDS

Once you have created a data file and a control file, you are ready to run the UIMPORT utility. At the DOS prompt of your workstation you simply type the following:

```
UIMPORT control_file data_file <enter>
```

In our example we type:

```
UIMPORT ACMECTRL ACMEDATA <enter>
```

The process will then initiate. Keep in mind that you must have Supervisor rights to the container of the context you have specified in the utility. Supervisor rights are necessary for security purposes and will prevent someone from creating objects and then populating them to the NDS database with approval.

If errors should occur during the process, you will receive error messages on your DOS workstation. You can use the > filename command to pipe error messages to a file. So, in our example we could type:

```
UIMPORT ACMECTRL ACMEDATA >LOGFILE
```

NCUPDATE

The NCUPDATE utility provides a network administrator with the ability to update the user's NET.CFG file if a change needs to be made to a container that affects the user's NET.CFG file. It is a convenient way to update the NET.CFG file with a new file that reflects the changes for a group of users.

AUDITCON

For some organizations it may be necessary to maintain an audit of network activity for a given server or perhaps a group of servers. This capability is provided in NetWare 4.1 by the the auditing utility. The AUDITCON utility allows independent auditors apart from your network administrators to audit network events. Network events that can be audited include the following:

- File and Directory Activities

  - Creating, modifying, deleting, salvaging, moving, or renaming

- Print Queue

  - Creating, deleting, and maintaining other activities

- Server Events

  - Downing a server, mounting or dismounting volumes

- NDS Events

  - Creating, deleting, moving, renaming operations, and changing to object security.

For more information on the AUDITCON utility refer to your Novell documentation for NetWare 4.1 or Novell's Dynatext under the topic "Auditing."

NPRINTER

NPRINTER.EXE is the replacement program for RPRINTER.EXE, which is no longer used. If you use the NLM-based print server, you will need to use the new NetWare printer program called NPRINTER to enable the print server to send jobs to a network printer. This printer can either be physically attached to the back of the NetWare server, attached to a DOS or OS/2 workstation, or directly to the network itself. Each printer requires the NPRINTER program in order to attach to the printer server and service print jobs.

There are a couple of different versions of the NPRINTER printer program that can be loaded in several different situations. For instance, NPRINTER can be loaded as a terminate-and-stay resident (TSR) program, as a NetWare Loadable Module (NLM), or in the memory of the printer device itself. NPRINTER can be loaded in the following three ways according to the printer configuration:

▶ Workstations with printer devices connected should run NPRINTER.EXE as a TSR to support the network printer. The printer is cabled to the parallel or serial port of the workstation.

▶ Printer devices that are connected to a NetWare server use the NPRINTER.NLM. The printer is cabled to the parallel or serial port of the NetWare server.

▶ Printer devices connected directly to the network run the NPRINTER program from firmware or flash-memory.

With NPRINTER loaded on a workstation, the user of that workstation can run applications and function normally. The user can even access the printer device as a local printer in stand-alone fashion, as well as a network printer. However, the users are not encouraged to use the printer as a local printer because it might conflict with the network printing activities. Naturally, the type of workstation and CPU speed affects the printing performance. For most large network sites this type of configuration is not used.

New to NetWare 4.1 is the ability of the users to specify a printer instead of a print queue when sending their print jobs. Users no longer have to know about print queues; they can simply send their print jobs to a printer by specifying the printer name. The network takes care of all print queue and print server activities. Sending a print job to a printer instead of a print queue is more intuitive to the user.

NMENU

The NMENU.BAT program is a descendent of the older MENU.BAT program and is backward compatible with the older NetWare operating systems. This utility enables you to provide a menu for your NetWare 4.1 users if you do not want to use DOS. Because so many workstations now use MS Windows, the use of menus is not as common.

For Windows-based users it usually makes more sense to look at Novell's Application Launcher, which enables you to create an NDS application icon in a Windows group. The user can simply click on the icon to execute the application. For more information on the NetWare Application Manager, see the "NetWare Application Manager" section earlier in this chapter.

NetWare 4.1 Server Utilities

The following section explains the more commonly used NetWare 4.1 server utilities and SET parameters available for network administrators. For information on DSREPAIR and DSTRACE set commands, see Chapter 10.

DSMAINT UTILITY

The DSMAINT.NLM utility (currently available only on NetWire and the Web) provides control of NDS when certain hardware maintenance operations are necessary. Because it deals with the Directory information on a specific server, the utility is run from that server's console. You may want to copy it to the SYS:\SYSTEM directory so it is available if needed.

An example of the DSMAINT main screen is shown in Figure 2.30.
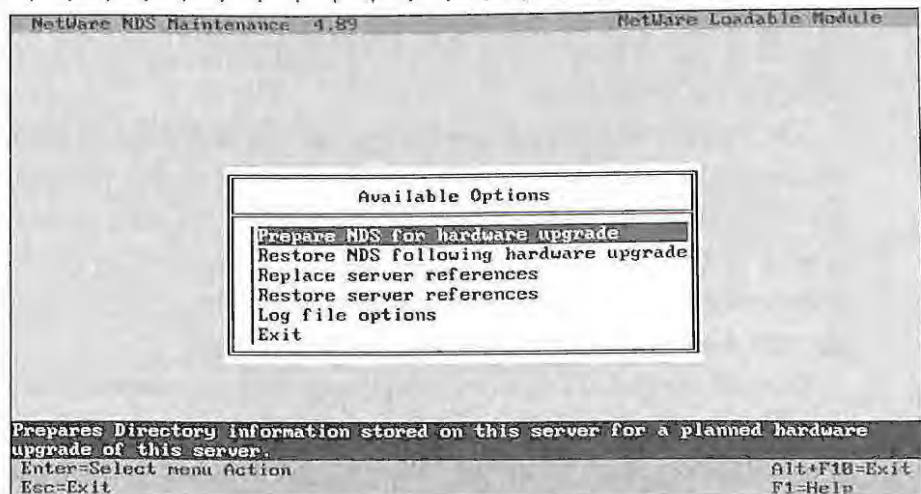
You can also use DSMAINT to maintain server references during a brief server shutdown and to more easily handle the transfer of NDS data when replacing an old SYS volume disk drive. Refer to the documentation that comes with DSMAINT.NLM for detailed instructions.

DSMAINT is included as part of the DS Enhancement Pack, which is available from NetWire and from Novell's Web server on the World Wide Web. The DS Enhancement Pack also contains a number of other utilities and tools to assist you in working with NDS.

The DSMAINT utility provides functionality to address two specific scenarios that NetWare administrators may experience. Each specific situation is addressed by a pair of features in the utility that work together. One command begins a process; another completes it.

Scenario One: NDS and Upgrading Server Hardware

There are times when a server requires an upgrade that does not affect the server as a Directory object. For example, the SYS: volume may be physically located on an old hard disk drive that needs to be upgraded.

In this situation, you no longer need to remove NDS from the server. You can use DSMAINT to prepare Directory information on the server for the upgrade. Then, after you have completed the upgrade, you can restore Directory information to the server with DSMAINT.

The Prepare NDS for a hardware upgrade option prepares the Directory information on this server for a planned hardware upgrade of this server. DSMAINT creates a file (SYS:\SYSTEM\BACKUP.DS) that stores all the Directory information on this server, including replica information. This file should be included in backup procedures before bringing the server down.

The Prepare NDS for a hardware upgrade option locks and disables the Directory on this server, preventing any data change. To other servers that normally communicate with this server, the server appears to be down. Any Directory information that normally is sent to the locked server is stored by other servers in the Directory; the stored information is used to synchronize the server when it comes back online. This is true anytime a server is unavailable for any period of time.

Because the global Directory is expecting the server to come back online quickly, you should not plan on taking several days to upgrade the server. Complete the upgrade promptly and restore Directory information on the server as soon as possible.

The Restore NDS after a hardware upgrade option uses the file created by the Prepare NDS for a hardware upgrade option (SYS:\SYSTEM\BACKUP.DS) to restore Directory information on this server. Before the Directory is restored, DSMAINT ensures that the server is in the same relative state as it was before the upgrade. DSMAINT ensures that the server's object and authentication keys are the same and that the server still exists in all the replica rings for copies that were on this server before the upgrade and as the same replica type.

CONSULTING EXPERIENCE

If you use backup software that needs to be logged in to the Directory, log it in before you use the Restore NDS after a hardware upgrade option. Because the option disables the Directory on this server, you cannot authenticate to this server after performing the option.

Procedure To upgrade the hardware, complete the following steps:

1 • Log in your backup software or if you have a current backup log in as ADMIN or administrative user. This step ensures that there is an authenticated connection with ADMIN rights to SYS:SYSTEM.

2 • Load DSMAINT and use the Prepare NDS for a hardware upgrade option; then, back up the server. If a backup was already performed, the BACKUP.DS file will need to be copied to the client's hard drive or to some location other than the server.

This step not only backs up the data, but it also gets a backup of the BACKUP.DS file in the SYS:SYSTEM subdirectory, which was created by DSMAINT.

3 • Bring down the server and perform the upgrade.

4 • Use the INSTALL utility to reinstall NetWare and place a temporary Directory on the server. Install the server to its own temporary Directory tree, not to your normal Directory tree.

The temporary Directory tree will be replaced in step 7.

5 • Copy the DSMAINT.NLM and restore BACKUP.DS to the SYS:\SYSTEM directory.

6 • Use the INSTALL utility to remove NDS from this server. This option is located under INSTALL's Directory options menu.

7 • Load DSMAINT.NLM at the server console and use the Restore NDS after a hardware upgrade option to restore the correct Directory information to the server.

8 • Restore data from backup performed in step 2.

9 • Load INSTALL and upgrade mounted volumes.

CONSULTING EXPERIENCE

This procedure may create a trustee assignment that did not exist before the upgrade. By default, the container object into which the server is installed receives Read and File Scan rights to the server's SYS:\PUBLIC directory. If these rights were previously removed you will need to remove them again.

Scenario Two: Maintaining Server References During a Brief Shutdown

At times, it is necessary to remove a NetWare 4.1 server object from the Directory for a brief period of time. For example, in the case of a corrupt authentication key, it is necessary to reinstall NDS on the server. During the remove NDS process, the NetWare 4.1 server object is removed from the Directory. When the NetWare 4.1 server object is removed from the Directory, objects that reference it in their required attributes can become Unknown objects. A similar type of problem can occur with services like printing that are associated with a physical server. Queues, for example, have a host server attribute. This attribute tells the printing utilities where to find the queue directory.

When the host server is deleted, the queue becomes unusable because the location of the queue directory is not known.

With DSMAINT you can avoid losing objects and ease reinstallation by replacing references to the server with references to another object that you create for this purpose. After installing NDS on the server again, you can use DSMAINT to replace these references to the server in other objects' Host Server, Host Device, or Message (Default) Server attributes.

The Replace server references option searches the Directory and replaces references to this server's NetWare 4.1 server object in other objects' Host Server, Host Device, or Message (Default) Server attributes with a reference to another Directory object.

The Restore server references option restores references to this server in other objects' Host Server, Host Device, or Message (Default) Server attributes. This option reverses the replacements made by the Replace server references option.

Procedure To reinstall NDS on the server, follow these steps:

1 • Begin by selecting an object for "holding" the references. This object can be an existing User object, but must not be a NetWare 4.1 server object. The user object you have logged in would be appropriate.

2 • Now select the Replace server references option. You are required to enter the full name of the container where you want to begin searching for objects that reference this server's NetWare 4.1 server object. You also need to enter the full name of the object you want DSMAINT to use as a replacement value (such as a TEMP User object).

3 • At this point, you can remove and reinstall NDS.

4 • Once NDS is properly operating, you can select the Restore server references option to reverse the replacements made by the Replace server references option. You will again be required to provide the full name of the temporary object that is holding the references.

NOTE

DSMAINT automatically removes volume IDs from the physical volumes on the server so that Volume objects are not removed during a remove NDS procedure.

DSMERGE UTILITY

The DSMERGE utility enables you to merge NDS trees into the same tree. The DSMERGE.NLM provides the mechanism for combining two trees through four basic functions:

▸ Checks status of NetWare 4.1 servers in the Directory Tree before the merge

▸ Checks time synchronization status of NetWare 4.1 servers before the merge

▸ Renames a directory tree when necessary

▸ Performs the merge of two trees when the previous criteria are satisfied

The main screen of DSMERGE enables you to perform these operations and is displayed in Figure 2.31.

FIGURE 2.31

The main DSMERGE screen

Check Status of NetWare 4.1 Servers

Before and after the merge of two trees it is important to check the status of your NetWare 4.1 servers. Before the merge, you want to check the versions of each NetWare 4.1 server and their current status. Table 2.3 lists the possible statuses and brief descriptions.

TABLE 2.3

NetWare 4.1 server statuses

SERVER STATUS	DESCRIPTION
DS Locked	The server responds with an error message indicating that the NDS name service is locked and not accessible.
Inaccessible/Down	The server does not respond and is considered to be inaccessible.
Error <-N>	N is the Directory Services error code number. This status indicates that the server is not responding because of the error condition listed in the value N.
Wrong Tree	The server is responding with a tree name that does not match the tree name of the local server container the master replica of [ROOT].
UP	The server is up and functioning.

The check phase of DSMERGE also ensures that the current server on which you are running this utility holds a replica of the [ROOT] partition. If it cannot find a replica of [ROOT] the operation will abort.

Next, the DSMERGE utility will attempt to find all NetWare 4.1 servers in the tree by searching a list of servers to collect the server name, version of NDS, and tree name. If a server in the list cannot be found or has an incorrect tree name, the utility will issue an error message.Once all the servers have been contacted, the operation will display the list of servers on the screen and their statuses. An example of checking the servers before a DSMERGE operation is shown in Figure 2.32.

```
Netware 4.1 DS Merge Tree  1.60                    NetWare Loadable Module
Tree name: ACME_TREE
Server name: CAM-SRV1.CAMELOT.ACME

              ┌──────────────────────────────────────────────┐

    ┌─────────────────────────────────────────────────────────────────┐
    │         Time Synchronization Information For Tree ACME_TREE        │
    │                                                                   │
    │  Server name              │  Type      │ In Sync │ Time Delta     │
    │  CAM-SRV1.CAMELOT.ACME     │  Single    │ YES     │        0       │
    │  TOK-SRV1.TOKYO.ACME       │  Secondary │ YES     │        0       │
    │                                                                   │
    └─────────────────────────────────────────────────────────────────┘
              └──────────────────────────────────────────────┘

    This list shows each server recorded in the tree's root partition along with
    time synchronization information.
                              Up=Previous server          Alt+F10=Exit
    Esc=Return to main menu   Down=Next server             F1=Help
```
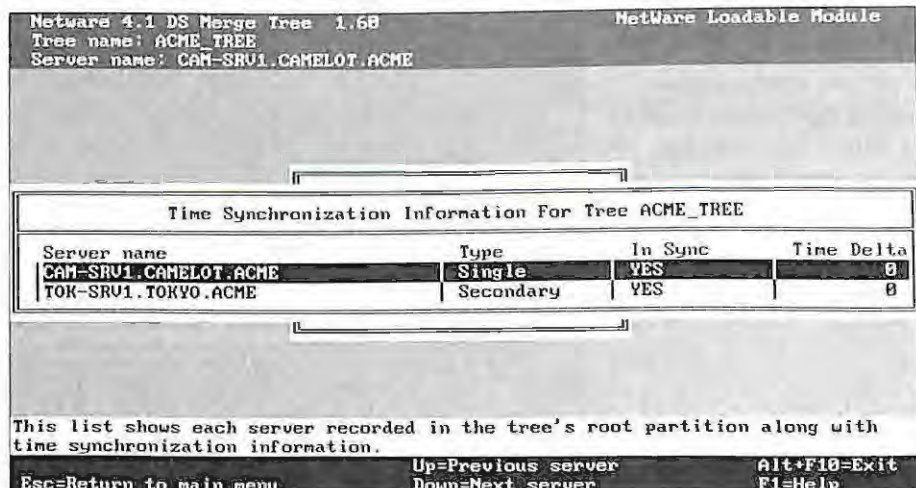
Check Time Synchronization

Another important check that the DSMERGE utility performs is the status of time synchronization on all servers to be merged. Both trees must be synchronized and they must refer to the same time source. Once DSMERGE is loaded on a server containing a replica of the [ROOT] partition, you can initiate the time synchronization command as shown in Figure 2.33.

The option will check the time status of all servers for server name, NDS version, and the time synchronization status. If there is a time delta between a NetWare 4.1 server and the local server running DSMERGE, the difference is displayed as shown in Figure 2.34.

FIGURE 2.33

Time synchronization status screen in the DSMERGE utility

FIGURE 2.34

If time synchronization differs, the DSMERGE utility will show the time delta between servers.

Your merged Directory tree must only have one reference or single reference time source. If the two trees you are merging both have a reference or a single reference server, you must assign one of them as a secondary server prior to initiating the merge.

Rename a Directory Tree

Another useful feature found in the DSMERGE utility is the capability to rename an NDS tree. You must use this option if you are merging two trees that have the same name. Keep in mind that currently this is the only Novell utility that will enable you to change the name of your NDS tree. You must load the DSMERGE.NLM on the server that contains the master replica of the [ROOT] partition. The utility will first verify that you are running on the server containing the master replica of [ROOT].The master replica can calculate a list of servers that the utility can contact to change the names of servers holding a replica of [ROOT].

If a server in the list cannot be contacted, a warning message will be issued. You can proceed to rename the tree if servers are unavailable, but you must enter a confirmation to continue the operation.

The operation actually begins by changing the name of the local tree's [ROOT] object to the new tree name. If an error should occur during this process, the operation is aborted. After the local server has changed the name of the [ROOT] object, a command is issued to change all servers in the local tree to the new tree name. Keep in mind that once this process has completed there are many more synchronization activities being performed in the background. Once initiated, this process is irrevocable. After completion of the operation, you should check the status of the servers once again to verify that all servers have received the new name.

Perform the Merge Operation

Once you have checked and verified the status of your NetWare 4.1 servers, along with time synchronization, you are ready to perform the merge operation. After the completion of DSMERGE, you will have a single tree with a common [ROOT]. The operation merges the selected source tree with a target tree. The end result is a tree with the target tree's name of [ROOT].

The DSMERGE.NLM must be loaded on a server containing the master replica of [ROOT] of the source tree. The operation occurs in four phases:

Phase 1: Merge Check   This check ensures that your source server running DSMERGE.NLM contains the master replica of the [ROOT] partition. If this check

is successful you will see a screen similar to the one shown in Figure 2.35. This screen asks for the tree name of both the source and target trees along with their administrative names and passwords.

Phase 2: Merge Preparation   This phase modifies the source tree partitions. The utility will do a schema comparison. If the schemas do not match, you will need to use DSREPAIR to import a remote schema from the other tree. It will also split the [ROOT] partition from all other objects in the tree, making a partition that contains only the [ROOT] object. All other Read/Write or Read Only replicas of [ROOT] will also be removed from any servers in the tree. Once this operation is completed the utility can actually merge the trees. Keep in mind that the partitioning operations mentioned here could also be performed beforehand with the Partition Manager or PARTMGR utilities. The first two phases are displayed in the utility before you continue as shown in Figure 2.36.

FIGURE   2.35

The Merge Trees Information screen

```
Netware 4.1 DS Merge Tree  1.68                    NetWare Loadable Module
Tree name: UN_TREE
Server name: UN-SRV1.UN

                                                          Available Trees

                                                          ACME_TREE

                        Merge  Trees  Information

     Source tree:          UN_TREE
     Administrator name:   admin.un
     Password:

     Target tree:          ACME_TREE
     Administrator name:   admin.acme
     Password:


Type the name of the target tree or select it using the cursor positioning keys,
then press <Enter>.
  Enter=Select tree                   Ins=Enter server address      Alt+F10=Exit
  Esc=Return to merge form            Down=Next tree                 F1=Help
```

Phase 3: Merge Trees   Once the first two phases are completed, the merge operation will begin. Once again, the operation will gather the list of servers known by the source server. Each server in the source tree will be contacted to obtain its server address and distinguished name in preparation for merging the source tree to the target tree.

The merge operation actually changes the name of the source tree [ROOT] object to that of the [ROOT] in the target tree. Once this process occurs the source server receives a new name and issues a command to change the tree name of all servers in the source tree.

IPR2017-01290
Ubisoft EX1008-1 Page 130

FIGURE 2.36

The first phases of the DSMERGE process are displayed for information purposes and are followed by another screen showing the next two phases.
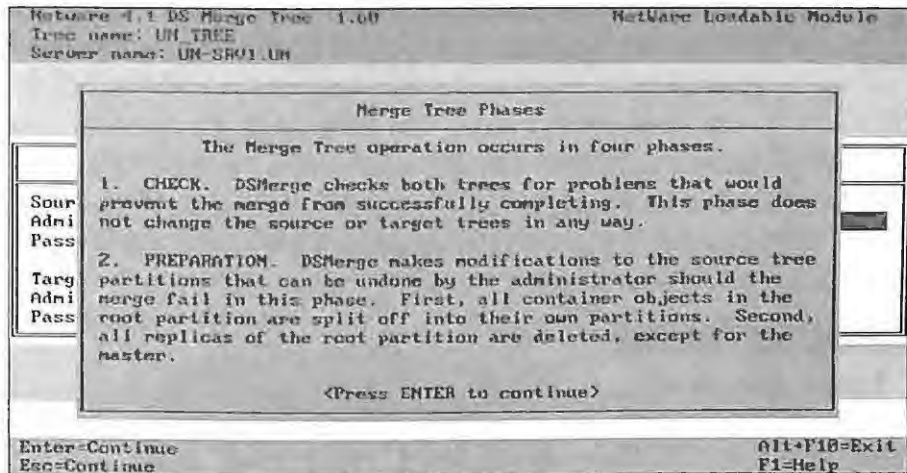
Phase 4: Merge Completion The merge completion phase includes the continuation of renaming the tree and servers in the source tree. A copy of the target tree's [ROOT] replica will then be copied over to the source server. This operation may take some time depending on the size of the target's [ROOT] replica. Any objects that are no longer needed will be removed by the background synchronization processes.

Some manual tasks may have to be completed after the two trees are merged together. These tasks would include placing copies of the [ROOT] replica where needed in your tree. You may have to restore any objects that were deleted before the DSMERGE utility was run. Because the source tree has had significant changes to the partitions below [ROOT], you will have to use the Partition Manager or PARTMGR utilities to place partitions back again. Follow the guidelines discussed in Chapter 6 of this book.

Finally, you may have to check workstations that have the PREFERRED TREE statement set in their NET.CFG. Each workstation with the source tree name will have to be changed to reflect the target tree name. This can be accomplished through the use of batch files or an update program, such as the UCUPDATE utility, contained in the user's container login script.

NETSYNC UTILITY

NETSYNC is a management utility designed to expand the benefits of NDS to NetWare 3 file servers. With NETSYNC you have the ability to synchronize NetWare 3 users and groups with objects that are contained in a NetWare 4.1 server's bindery context.

Therefore, when you create or modify a user in the bindery of a NetWare 4.1 server, the changes are synchronized with all NetWare 3 servers participating in the NETSYNC cluster.

A NetWare 4.1 server can have up to 12 NetWare 3 servers attached in what is known as a cluster. Because all servers in the cluster are synchronized you need to create a user only once because the user will exist on all NetWare 3 servers that are actively attached. An example of a cluster of five NetWare 3 servers is shown in Figure 2.37.

FIGURE 2.37

A cluster of five NetWare 3.12 servers is connected to a single NetWare 4.1 server. The NetWare 4.1 server can have up to 12 NetWare 3.12 servers attached.
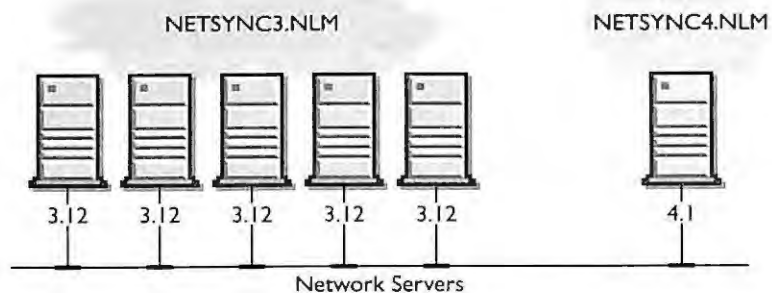
You can also use the NWADMIN or NETADMIN utilities found in NetWare 4.1 to easily manage NetWare 3 users and groups. Printing is also manageable with the NetWare 4.1 versions of PCONSOLE and NWADMIN.

Using NETSYNC

NETSYNC is intended as a temporary solution for central administration of a mixed NetWare 3 and NetWare 4.1 environment. For those servers running versions of NetWare 3 that cannot immediately be moved to NetWare 4.1, this utility provides a good, albeit, temporary solution. It can also be used in environments that are running Novell's NetWare Name Service domain and cannot move all the servers in the domain to NetWare 4.1 immediately.

The NETSYNC NetWare Loadable Module can be loaded on a single NetWare 4.1 server and up to 12 NetWare 3 servers to provide communication between the binderies of the NetWare 3 servers and the emulated bindery of NetWare 4.1. NETSYNC on the NetWare 4.1 server performs the following functions:

- Copies all users and groups from the NetWare 3 server's binderies into the bindery context of the NetWare 4.1 server when NETSYNC.NLM is first enabled on the NetWare 4.1 server.

- Downloads all objects in the NetWare 4.1 bindery context to the connected NetWare 3 servers. A combined bindery of NDS users and groups along with the NetWare 3 binderies is downloaded to each NetWare 3 server. This process gives all servers (NetWare 3 and the NetWare 4.1 server) a common, synchronized bindery.

- Monitors any changes to the bindery context of the NetWare 4.1 server. The NETSYNC utility will monitor any user or group changes made on the NetWare 4.1 server and synchronize these changes with all NetWare 3 servers connected in the cluster.

- Synchronizes bindery information from any NetWare 3 server that has been temporarily removed from the cluster. Any changes that may have been made to the bindery during the server's absence will be sent down by the NetWare 4.1 server.

Installing NETSYNC

Installing NETSYNC is not difficult. All of the components are already stored on your NetWare 4.1 file server. The SYS:SYSTEM directory contains the necessary NetWare 4.1 files, and the SYS:SYSTE\NETSYNC directory contains the NetWare 3 files. At least one NetWare 4.1 server is required but you can have up to 12 NetWare 3 servers connected to one NetWare 4.1 server. For authentication of each NetWare 3 server you will also need an available licensed connection to the NetWare 4.1 server.

To load NETSYNC simply type **LOAD NETSYNC4** at the NetWare 4.1 server console. Always load NETSYNC4 server first and authorize the NetWare 3 servers before beginning the installation of NETSYNC3 on the NetWare 3 server. If you don't load the NETSYNC4 server first, the NetWare 3 servers are not authorized to connect to the NETSYNC cluster. This installation program will automatically create the SYS:SYSTEM\NETSYNC directory that contains log files and the other NETSYNC file that will be copied automatically when each NETSYNC module is loaded on the NetWare 3 servers.

The loading of your NetWare 3 servers requires you to first authorize each server to connect to the NetWare 4.1 network. When you have loaded NETSYNC4, you will see the screen shown in Figure 2.38. Pressing Enter to the Edit Server List option enables you to enter NetWare 3 servers by name to begin the authorization process. You can enter up to 12 NetWare 3 servers on the list.

As an added measure of security you will also need to enter a NETSYNC password. This password will then be required as each NetWare 3 server initially requests entry into the cluster. The password is used only once.

FIGURE 2.38

The NETSYNC4 utility enables you to authorize NetWare 3.1 servers for synchronization with a NetWare 4.1 host server.
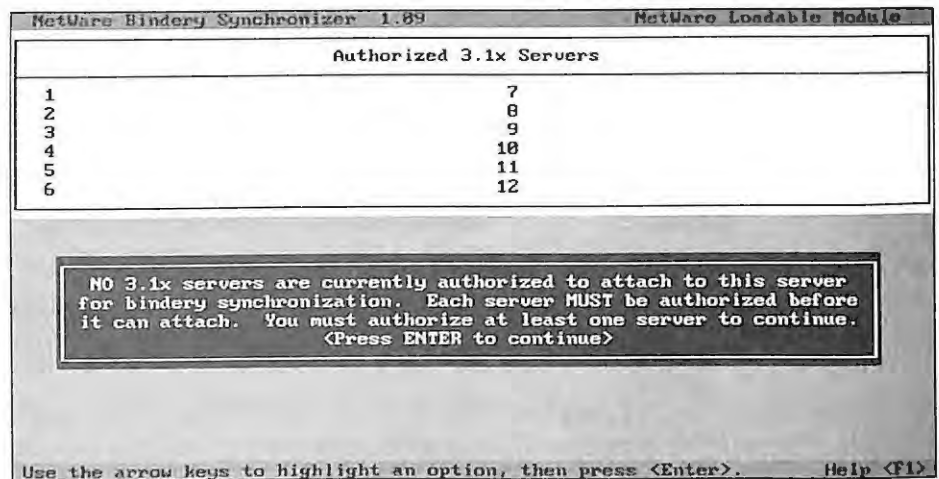
> **You can use a different password for each NetWare 3 server. However, since this password is used only once, for simplicity we recommend the same password for all NetWare 3 servers in the cluster. In addition, do NOT use your ADMIN or Supervisor password as your NETSYNC password as this may pose a security risk.**
>
> TIP

Next, you will be asked if you want to copy the program files to the NetWare 3 server by pressing the Down arrow key. This option appears when you first authorize a NetWare 3 server for installation of NETSYNC.

Loading NETSYNC3

Every NetWare 3 server that participates in the cluster must load the program NETSYNC3. Again, be sure that you have completed the steps to loading NETSYNC4 on your NetWare 4.1 server before starting this phase. The first step is to reboot the

NetWare 3 servers because they will execute some new files that have been downloaded by the NETSYNC4 process earlier. Once your NetWare 3 server is back up and running you can now type **LOAD NETSYNC3** at the server's console.

Once the NETSYNC3 program is loaded you will be asked a couple of questions regarding NetWare Name Services. If you are not running NNS, select NO to these questions and proceed. If you are running NNS, refer to Novell's documentation for information.

The utility will ask you to enter the name of the NetWare 4.1 host and the password. After successful entry of your password the utility will upload the NetWare 3 bindery information to the NetWare 4.1 bindery context. The bindery type objects at the bindery context are in turn loaded onto the NetWare 3 server.

For further configurations of NETSYNC you can use the NETSYNC Options menus for NETSYNC3 and NETSYNC4, such as viewing or editing the log file, editing a server list, and other activities.

Managing Objects with NETSYNC

Upon successful installation of NETSYNC, you will be able to manage your NetWare 3 objects with the NetWare 4.1 NWADMIN or NETADMIN utilities. DO NOT go back and use SYSCON to modify user and group information on NetWare 3 servers in the cluster. The use of SYSCON on the NetWare 3 server will not synchronize changes to NDS and should be avoided in this configuration.

NETSYNC also provides printing support. After you have completed your installation of NETSYNC3 and NETSYNC4, all workstation print utilities are copied to the NetWare 3 servers, and PRINTCON and PRINTDEF databases are updated to the NetWare 4.1 formats. You can also choose to move your NetWare 3 print servers into a single print server on the NetWare 4.1 server. Your printers are moved into Directory Services where they can be managed from a single NetWare 4.1 print server.

NetWare 4.1 print utilities are immediately available for use in your NetWare 3 environment once NETSYNC is operational. For more information on using print utilities with NETSYNC, refer to your Novell documentation.

NETSYNC and the Bindery Context

NDS provides compatibility to NetWare 2 and NetWare 3 using a feature called bindery services. This feature enables bindery versions of NetWare and other bindery-based applications to access the NDS tree as if it were the bindery. Bindery services is accomplished by setting a bindery context on the server, which specifies the name of the NDS tree container(s) to search as the bindery.

**In order to set the bindery context on a server, you can enter
SET BINDERY CONTEXT = "OU=OPS.OU=CAMELOT.O=ACME"
at the server console. The server bindery context can also be set
using the server-based SERVMAN utility.**

Prior to NetWare 4, object information was not available from a distributed directory. A NetWare 3 server stored only information related to its own server in the bindery. Therefore, applications were written that accessed each NetWare 3 server's bindery for user or connection information. NDS provides bindery services that enable objects in a container to be accessed by bindery-based servers and clients as well as NDS objects. When a bindery-based application makes bindery calls to the server, the server sees the objects in the container where the server bindery context is set.

**In terms of bindery context on the server, NDS will see only the
objects that previously existed in a NetWare 3 bindery as the bindery
objects. These objects are limited to user, group, print queue, and
print server. In addition, NetWare 4.1 has added the profile object to
assist with migrations from NetWare Name Services (NNS).**

The default server bindery context is set to the container where the server was installed. The context can be changed to look to another container or OU if you want. Previous versions of NetWare 4, up to and including NetWare 4.02, allow you to set only a single context for your server. NetWare 4.1 enables you to set up to 16 contexts for a single server.

Since NetSync synchronizes NetWare 3 user and groups with objects in a NetWare 4.1 bindery context, you must enable the host 4.1 server with bindery services. The NetWare 4.1 server periodically queues up and signals the NetWare 3 servers to retrieve the changes in the cluster. If more than one NetWare 4.1 server is using the same bindery context, you will have all servers with the same context synchronizing with the NetWare 3 servers in the cluster. Although there is no theoretical limit on how many servers can be using the same bindery context, the NetSync operation is CPU intensive. Thus, the more objects to synchronize, the more overhead is placed on a server.

In addition, if you desire to set multiple bindery contexts for a server, the process is also governed by the number of objects and the speed of your links during synchronization.

SERVMAN

The SERVMAN utility monitors your NetWare 4.1 servers and is broken down into four main parameter groupings. Keep in mind that the SERVMAN utility is a menu interface for all the NetWare 4.1 SET parameters and makes administration of these parameters a little easier. The four groups are listed below.

Server Parameters

Server parameters enable you to view and modify operating system parameters. Server parameters include packet communication options such as maximum and minimum packet receive buffers and watchdog packets. Figure 2.39 shows the main screen for server parameters. To make a change to any of these options, simply highlight the option and press Enter. You can then backspace over the old option and enter the new one. If the option is toggled, then simply highlight the option and press Enter to toggle to the desired value.

FIGURE 2.39

The Server Parameters screen
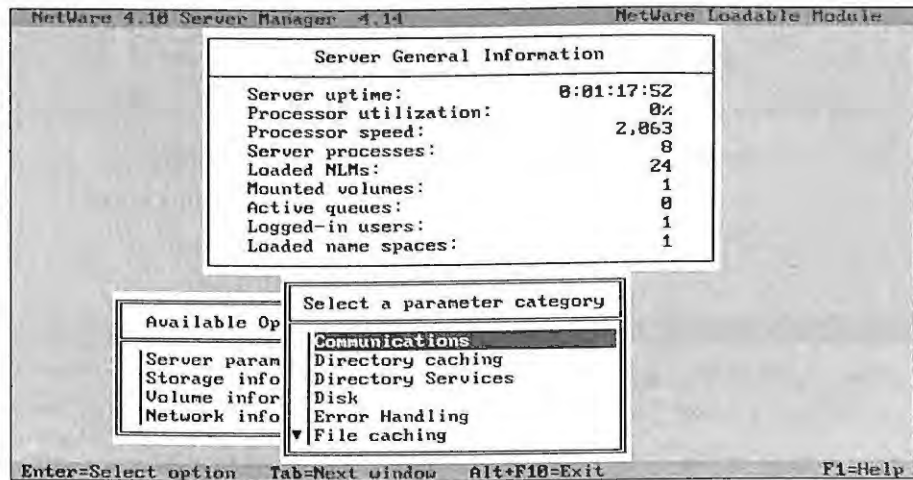
NOTE For a detailed description of using **SERVMAN** for time synchronization control found under the **SERVER PARAMETERS** screen, refer to Chapter 7.

Storage Information

The storage information option enables you to view adapter, device, and NetWare

partition information. An example of the main screen for storage information is shown in Figure 2.40.

FIGURE 2.40

The Storage Information screen
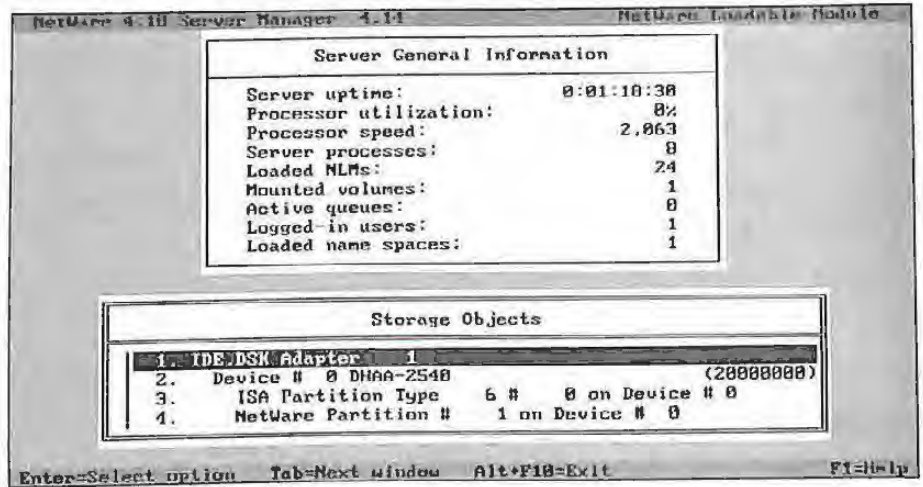
Volume Information

The volume information option displays the NetWare volume statistics for currently mounted volumes. These statistics provide information such as compression status, number of directory entries, and size of suballocation units. An example of a volume information screen is shown in Figure 2.41.

FIGURE 2.41

Volume Information screen for the SYS volume
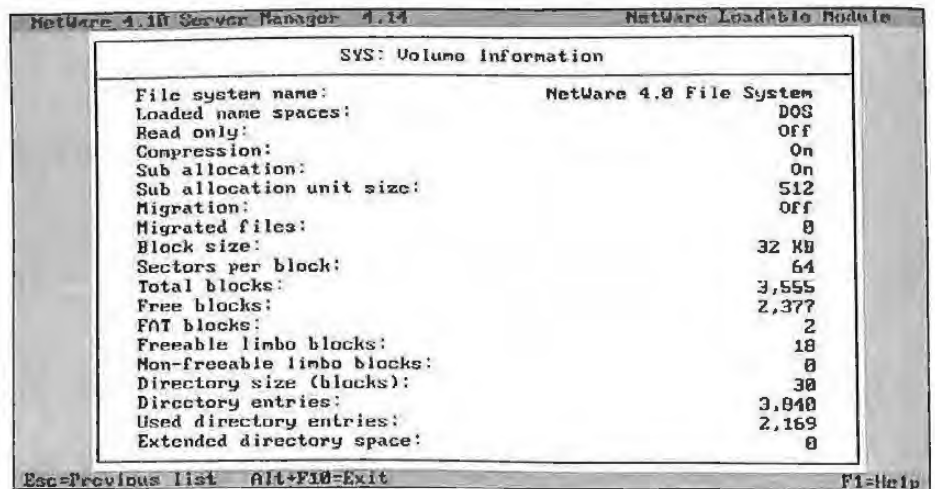
Network Information

The network information parameters enable you to view packet information such as the numbers of packets transmitted and received. This information is helpful if you are trying to diagnose a problem with network traffic. An example of this statistics screen is found in Figure 2.42.

FIGURE 2.42

The Network Information screen

```
NetWare 4.10 Server Manager  4.14                    NetWare Loadable Module

                          Network Information

            Packets transmitted:                 2,780
            Packets received:                     1,498
            Packets waiting to be sent:               0
            Unclaimed packets:                        0
            Get ECB buffers:                      3,258
            Get ECB requests failed:                  0
            Maximum number of LANs:                 256
            Current number of LANs:                   1
            Loaded protocol stacks:                   1


              Available Options

              Server parameters
              Storage information
              Volume information
              Network information


Esc=Previous list    Alt+F10=Exit                              F1=Help
```

CHAPTER 3

▶ ·

NetWare Directory Services Objects and Properties

"If two objects have the same logical form, the only distinction between them, apart from their external properties, is that they are different." Wittgenstein

NetWare Directory Services (NDS) is a collection of objects that follows a set of rules regarding how they are created and used. Becoming familiar with NDS objects and properties and the ways in which they are used to build the NDS tree structure will enhance your understanding of NetWare 4.1. Your understanding of the schema will help you to determine the structure of additional objects that may be added to your NDS schema through the installation of add-on or third-party applications. In addition, an understanding of the internal structure of both the NDS objects and the schema will give you greater flexibility in managing and designing your NetWare 4.1 network.

This chapter first begins with a basic discussion of the NDS objects and properties (or attributes). Next, we will explain the NDS schema — its components and rules and how the NDS objects are created. Schema extensions are discussed to help you understand what happens when additional object classes or properties or modifications are added to existing classes. We will then explore the purpose and use of each individual NDS object with examples of how they can effectively be used in your tree. Where pertinent we have also included consulting experiences that will help you in understanding how various objects have been used in actual customer environments.

Introduction to NDS Objects

Your primary responsibility as a network administrator is to maintain the servers, volumes, users, groups, printers, and other resources in the network. In order to help with this task, NDS enables you to view all the network resources as objects within a distributed or network-wide name service or NDS tree. You decide which network resources are created and placed as objects in the name service. These objects might include users, groups, printers, servers, volumes, computers, and so on.

Each of the entries in the NDS tree consists of the object and the properties or data stored as a property. For example, the User object has up to 75 properties, and some of the properties have multiple values. Table 3.1 shows an example of a User object. This table is not a complete example or list of all the properties, but notice that some of the properties have multiple values.

TABLE 3.1	OBJECT	PROPERTY	VALUE(S)
Example of a User object	User	Login Name	GWASHINGTON
		Given Name	George
		Last Name	Washington
		Title	President, Statesman, Farmer, Cherry Tree Killer
		Location	Washington D.C, Mount Vernon
		Telephone	(not invented)
		Fax Number	(not invented)
		Description	President of the United States

NDS Objects

The NDS objects are the entities that store the information or data about a network resource. The NDS objects represent both physical and logical entities on the network. Table 3.2 illustrates examples of both logical and physical network resources that can be represented by NDS.

TABLE 3.2	LOGICAL ENTITIES	PHYSICAL ENTITIES
Example of logical and physical entities	User	Printer
	Group	NetWare Server
	Print Queue	Volume
	Alias	Computer

Since there could be many NDS objects in the network, the objects are organized into a hierarchical structure called the NDS tree. A good analogy here is a file system. The file system is a hierarchy of subdirectories and files. The reason for the hierarchy is that the files are more manageable with this structure than storing them at the same level. The same is true with the Directory.

NetWare Directory Services classifies all objects as either *container* or *leaf objects* (*non-container objects*). The container objects are analogous to subdirectories in a file system, and leaf objects are analogous to the files.

CONTAINER OBJECTS

The container objects are those objects that are allowed subordinate objects. The container objects form the hierarchy of the NDS tree and are typically named after the locations, divisions, departments, and workgroups in your company. The container objects enable you to group the other NDS objects together in the tree. A branch or subtree of the NDS tree consists of a container object and all the objects it holds, which can include other container objects.

There are several types of container objects defined for your use. The container object base classes are as follows:

- ▶ TOP [ROOT] — a special object at the top of the tree

- ▶ Organization (O=) — represents the name of your company

- ▶ Organization Unit (OU=) — represents locations, departments, divisions, or workgroups

- ▶ Country (C=) — represents a 2-letter country object

- ▶ Locality (L=) — is named by "L" or "S" for state or both. Currently the Locality is not enabled by the Novell utilities.

Although, the NetWare 4.1 Directory defines five container objects, the current utilities support only the O=Organization, OU=Organizational Unit, and C=Country. The TOP [ROOT] class is supported but only the NDS system can create it. Future releases of the NetWare utilities might implement the L=Locality and S=State, as well as third-party products. Both can also be currently represented by using the OU=Organizational Units, which serve the same purpose.

Figure 3.1 illustrates how the container objects are the building blocks of the NDS tree. The container objects also help you organize and manage the leaf objects or non-container objects.

FIGURE 3.1

The container objects are organized into a hierarchical structure called the NDS tree.
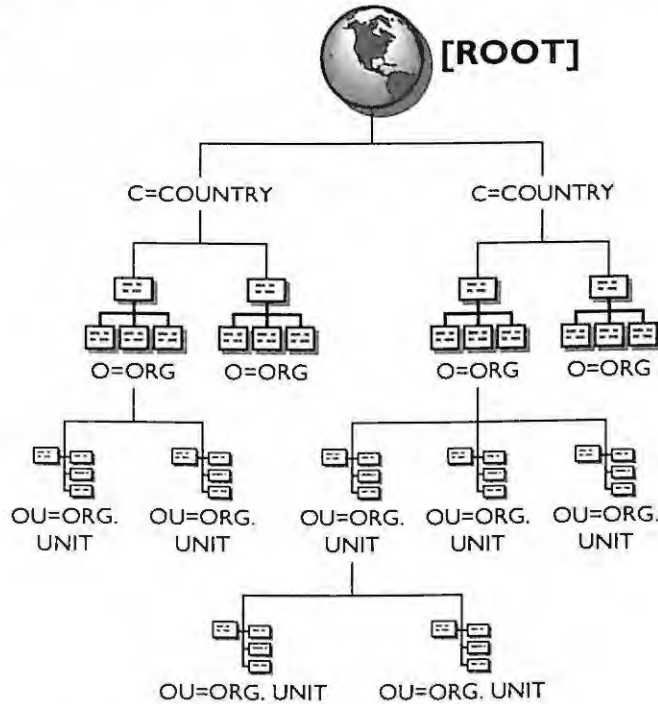
LEAF OBJECTS (NON-CONTAINER OBJECTS)

The leaf objects are located at the ends of the NDS tree branches and don't contain any other objects. These are the objects that are used to represent the network resources. The following is a list of some of the leaf objects for NetWare 4.1.

- AFP Server

- Alias

- Bindery Object

- Bindery Queue

- ► Computer

- ► Directory Map

- ► Group

- ► NCP Server

- ► CommExec

- ► External Entity

- ► Organization Role

- ► Queue

- ► Print Server

- ► Printer

- ► Profile

- ► User

- ► Unknown

- ► Volume

- ► Messaging Server

- ► Message Routing Group

- ► List

In the NDS tree, you can place container objects and leaf objects in different arrangements, according to your company's needs. Figure 3.2 illustrates how container objects organize the leaf objects.

FIGURE 3.2

The container objects
organize the leaf objects in
the NDS tree.
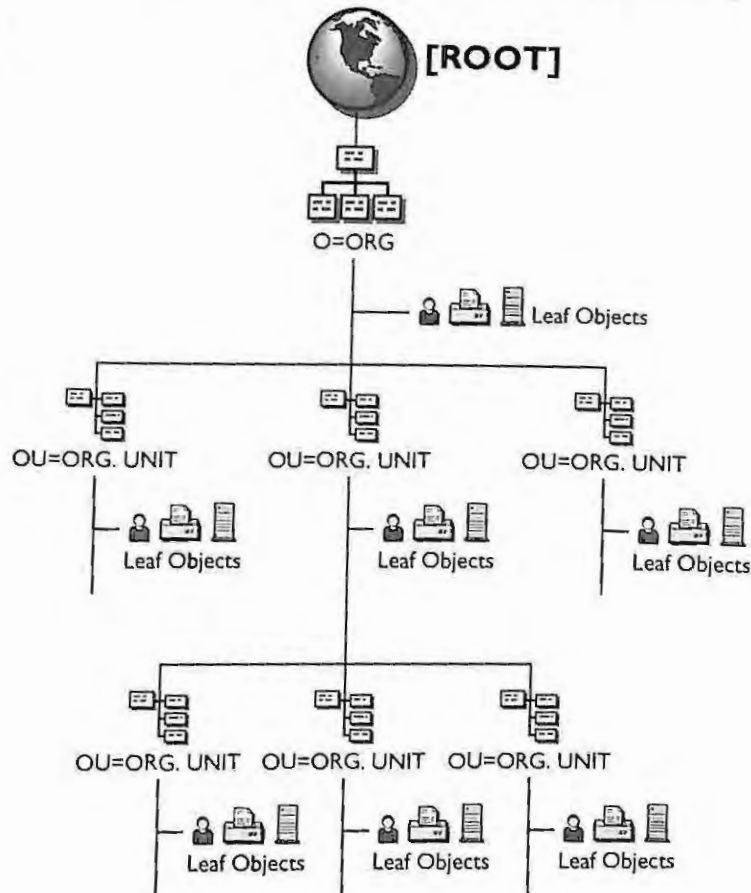
NDS PROPERTIES

The properties (also referred to as attributes) of the objects specify the type of information that the object can store. The terms property and attribute share the same concept and are used interchangeably throughout the rest of this book.

Different types of objects have different types of properties. Thus, it can be said that the properties define or describe the object. For example, a User object has the following properties: Login Name, Last Name, Group Membership, Telephone, Fax Number, and so on. On the other hand, a Printer object has the properties: Name, Network Address, Location, Print Server and so on.

There are two categories of properties for the object: mandatory and optional property. The mandatory properties are required during the object creation and cannot be removed. The Novell utilities take care of determining which properties are mandatory. Typically,

these are properties that name the object necessary for operation. Many properties are mandatory to comply with the X.500 standard. For instance, the surname(or last name) of a User object is mandatory. The host server property is mandatory when creating a Volume object. An object's optional properties can be entered as the network administrator deems appropriate.

The value of the property is the data information stored in each property. This is the object information that the users can read and write in many cases, although some properties are not readable or writable by the client. One property type can have a single value, while another can have multiple values. For example, the Telephone property of the user can hold several different telephone numbers.

There is a specific syntax for the value or data type of each property. One use of the syntax is to define the acceptable characters for the value field(s). More details on the syntax for each property can be found in the next section of this chapter.

The NDS Schema

The NetWare Directory Services schema supports object class inheritance, and each object and property is defined by a set of rules in the Directory. The schema rules dictate the requirements, limits, and relationship of the objects and properties that can exist in the Directory. NDS enforces those rules for each client request to change (add, delete, modify, and so on) an object or property.

The NDS schema is automatically stored on every NetWare 4.1 server you install, even if you don't have NDS replicas stored on that server. The NetWare 4.1 installation automatically places the schema on the server for the following reasons:

▶ The schema enables each server to know how to create all the NDS objects and properties.

▶ The schema placed on each server permits the administration utilities to access the schema during the creation of an object or objects.

The schema consists of three major components: the object class, the property or attribute type definitions, and the attribute syntaxes. These three components work together to establish the rules that control the creation of a particular object type in the

Directory. In Figure 3.3, you can see the relationship between these components. For instance, each object class is defined in terms of attribute or property definitions and other classes. The attribute types, in turn, are defined in terms of the attribute syntaxes. The attribute syntaxes are the data type of the values or information that can be stored in the attribute.

FIGURE 3.3

Relationship between the three components of the schema: objects, attributes (or properties), and attribute syntaxes

Object Class

Attribute Definitions (or Property Types)

Attribute Syntaxes
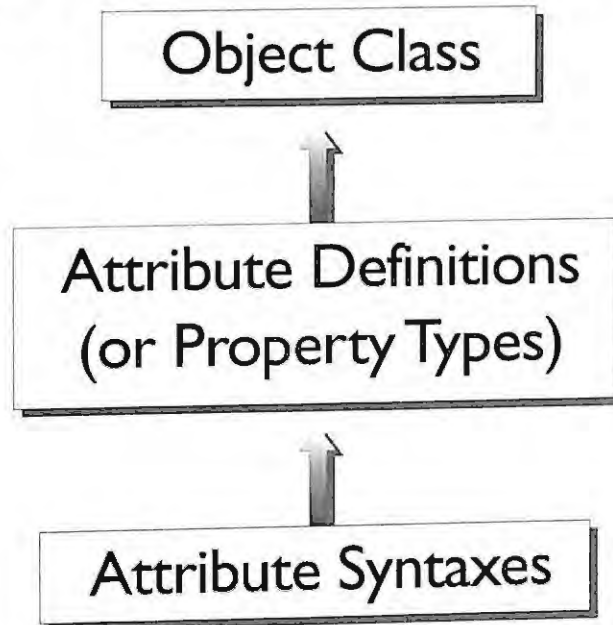
Keep in mind that, for the most part, the objects defined in the base schema for NetWare 4.1 are taken from the X.500 specifications for appropriate objects in a directory service. Other objects were defined that describe existing NetWare 3 bindery objects. Some objects may currently have little use and are intended for future enhancements to the Directory and thus were placed in the schema for future use.

OBJECT CLASSES

The object class defines the types of NDS objects that can be created and stored in the Directory. The object class is used as a set of rules to create the individual objects and to determine its set of characteristics. This means that every object in the tree belongs to an object class that specifies which attributes can be used or must be used as well as where

objects can be created in relation to other objects. Figure 3.3 illustrates how the object classes are constructed from the attributes types. Keep in mind that object classes can refer to other object classes as well.

The base schema that ships with NetWare 4.1 contains approximately 32 object classes. These object classes cannot be removed or deleted. The following is a list of 32 object classes stored in the schema.

- ▶ AFP Server

- ▶ Alias

- ▶ Bindery Object

- ▶ Bindery Queue

- ▶ CommExec

- ▶ Computer

- ▶ Country

- ▶ Device

- ▶ Directory Map

- ▶ External Entity

- ▶ Group

- ▶ Locality

- ▶ Message Routing Group

- ▶ Messaging Server

- ▶ NCP Server

- ► Organization

- ► Organizational Person

- ► Organizational Role

- ► Organizational Unit

- ► Partition

- ► Person

- ► Printer

- ► Print Server

- ► Profile

- ► Queue

- ► Resource

- ► Server

- ► Top

- ► Unknown

- ► User

- ► Volume

- ► List

Each of the object classes in the schema is determined by the distinctive information held in five components. The five components or type of information that make up an object class definition are:

- ▸ Structure Rules — Defines the relationship of the objects in the Directory tree. The relationship is constructed based on two items: Naming Attributes (how the object is named) and Containment Class (which object classes the object can be subordinate to).

- ▸ Super Classes — Builds the internal structure of the schema by relating the object classes to each other. Some object classes are simply the building blocks for the other object classes, or are classes from which the object class inherits its definition.

- ▸ Mandatory Attributes — The attributes listed in the mandatory attribute list for the object class are the attributes required during the object's creation and throughout the life of the object.

- ▸ Optional Attributes — The attributes listed in the optional attribute list for the object class are the attributes that are not required but may be added.

- ▸ Object Class Flags — There are five object class flags that are either set to ON or OFF. These object class flags include the container flag, effective status, and nonremovable flag.

As an example, the following is a list of the Group object class and its definitions:

Object Class:	GROUP
Class Flags:	Effective Class
Super Classes:	Top
Named By:	CN (Common Name)
Containment:	Organization
	Organizational Unit
Mandatory Attributes:	CN
Optional Attributes:	Description
	Full Name
	GID
	Locality Name
	Member
	etc.

The following is a list of the Organization object class and its definitions:

Object Class:	ORGANIZATION
Class Flags:	Effective Class, Container Class
Super Classes:	Top
Named By:	O (Organization Name)
Containment:	Country
	Locality
	Top
Mandatory Attributes:	O
Optional Attributes:	Description
	Detect Intruder
	E-Mail Address
	Facsimile Telephone Number
	etc.

Structure Rules

The structure rules for an object define the relationship between the objects in the NDS tree. The relationship between object classes is determined by two items. These items are Naming Attributes and Containment Class. The naming attributes determine how the object is named. The containment class determines where the object may appear in the Directory tree relative to the other object classes.

Naming Attributes   The naming attributes (or properties) determine how an object is named in the NDS tree. An Object is identified in the tree by its common name and also by the names of all its parent containers. An object's name includes the object name type and a value connected together with an equal (=) sign. Some object classes are container objects while other objects are the leaf objects (or non-container objects). Table 3.3 lists the object name types that are currently used in the naming attribute of the object class definition.

The first four name types — C=Country Name, L=Locality Name (S=State or Province), O=Organization Name, and OU=Organizational Unit Name — are all the container objects. The CN=Common Name and Bindery Type are used for the leaf objects.

An object's leaf-most name is referred to as its partial name or Relative Distinguished Name (RDN). For example, the user, MTERESA, in the ACME tree has an RDN of CN=MTERESA. If we refer to the user's distinguished name or full name, it would be defined as the RDN, MTERESA, plus the names of all the parent containers above the user MTERESA. The distinguished name would be written as follows:

```
CN=MTERESA.OU=FOOD.OU=SYDNEY.O=ACME
```

List of the object name types that are available for use in object name type definitions

NAME TYPE		NAME VALUE
C	=	Country Name
L	=	Locality Name
S	=	State or Province Name
O	=	Organization Name
OU	=	Organizational Unit Name
CN	=	Common Name
Bindery Type	=	Bindery Type

Therefore, the structure rules effectively control the formation of the distinguished names and relative distinguished names for each of the objects created in the NDS tree.

Nearly every object class will have one or more object name types defined in the Naming portion of the object class definition unless inherited by a super class. For example, the Organization object class is named by the "O" name type. The naming attribute of the Organizational Unit object class is defined as an "OU" name type. The Country object class has the "C" name type. It is mandatory that at least one naming attribute be defined or inherited.

The Named By attribute is not always a reflection of the class to which an object belongs. For example, leaf objects such as User or Printer are named by the Common Name (CN). The CN= name attribute does not by itself give you any indication to which class the object belongs.

An object class can allow for multiple object name types in the Named By attribute in the definition. In these cases any one of the object name types can be used to name the object. For classes having multiple naming attributes, any combination of the naming attributes can be used. The attribute values are connected together using the plus (+) sign.

A common occurrence of this peculiar naming convention occurs in the case of Bindery Objects. In the definition of the bindery object class, there are two object name type attributes called Bindery Type and Common Name (CN). A bindery object would then be defined as:

```
CN=ATPSUSER+BINDERY TYPE=83
```

The Locality object class definition also has a multivalued Named By attribute definition. It is named by either the name type L=Locality Name, or S=State or Province Name. Therefore, the RDN for a Locality object could appear as:

```
L=NORAD+S=COLORADO
```

Containment Classes   As we mentioned earlier in this chapter, some object classes are designed to contain other objects. The containment class list specifies where the individual object classes may appear in the NDS tree. Each object class has a definition or list of the container objects that it may be created in or subordinate to. This list, called the containment class list, builds the relationship between the object classes in the NDS tree.

An individual object can only reside or be created directly in a container that is found in its containment class list. For example, the Group object class can only be placed in or created subordinate to an Organization or Organizational Unit object. The containment class list for the Group object lists this relationship as illustrated in the list of Group object classes shown above. The following example for the Group object class is taken from the list of Group object classes:

Object Class:	GROUP
Named By:	CN (Common Name)
Containment:	Organization
	Organizational Unit

The containment class list simply specifies that the Group objects can be placed under either the Organization or Organizational Unit objects in the tree. Therefore, the containment class specifies where an individual object may appear in the Directory tree. The schema prohibits objects from being created directly below or subordinate to container objects whose class definition is not in the new object's containment list.

The following is a list of the container objects that are used in the containment class lists for each of the object class definitions:

- Top

- Country

- Locality

- Organization

- Organizational Unit

The containment class dictates the structure of the NDS tree and changes as the tree expands toward the bottom. For example, the Organization and Country object classes are the only object definitions that have Top in their containment class list. The Top object class is implemented as the [ROOT] object in the tree. This means that only Organization and Country objects can be subordinate objects to [ROOT]. The Organization and Locality object class definitions are the only object classes that have the Country object listed in their containment class. Thus, Organization and Locality objects can only be subordinate objects to Country. Continuing down the tree, the Organizational Unit has the Organization, Locality, and Organizational Unit (itself) defined in its containment class.

As the tree expands toward the bottom most of the leaf object class definitions have the containment class for both Organization and Organizational Unit. This means that most leaf objects will reside either within the Organization or Organizational Unit container objects.

Super Classes

The internal structure of the schema is built using the super classes specification for the object class. The super classes help designate the structure of the schema internally and do not dictate the hierarchical structure for the NDS tree. As stated above, the NDS tree structure is dictated by each object's definition of the containment class.

The super classes are those object class definitions in the schema from which other objects classes may be constructed. A complete definition for an object class is derived from the components of the object class itself plus all the components of its super class objects. An example of the super class structure for a User object class is illustrated in Figure 3.4.

The NDS Directory allows for classes to inherit from other class objects in a unidirectional manner. This means that object class B can inherit from object class A, but object class A cannot inherit from object class B. The concept of inheritance among objects in the schema stipulates that an object class inherits the features of its super classes. This means that all the definitions of the super class objects are inherited in the new object class. If object class B inherits from object class A, B must receive all the definitions from A. The Directory does not have the capability to inherit only parts of the super class definition. A class may inherit from multiple super classes.

FIGURE 3.4

*Super class structure for the
User object class as defined
in the schema*

Top
Effective Class
Super Classes: (none)
Containment: (none)
Named by: (none)
Mandatory: Object Class
Optional: ACL
Back Link
Bindery Property
Obituary
Reference

Person
Non-Effective Class
Super Classes: Top
Containment: (none)
Named by: (none)
Mandatory: CN
Surname
Optional: Description
Full Name
See Also
Telephone Number

Organizational Person
Non-Effective Class
Super Classes: Person
Containment: Organization
Organizational Unit
Named by: CN
OU
Mandatory: (none)
Optional: EMail Address   Postal Address
Facsimile Telephone No.   Postal Code
Locality Name   Postal Office Box
OU   S
Physical Delivery Office   SA
Name   Title

Organizational User
Effective Class
Super Classes: Organizational Person   Login Grace Remaining   Password Expiration
Containment: (none)   Login Intruder Address   Time
Named by: (none)   Login Intruder Attempts   Password Minimum
Mandatory: (none)   Login Intruder Reset   Length
Optional: Account Balance   Time   Password Required
Allow Unlimited Credit   Login Maximum   Password Unique
Group Membership   Simultaneous   Required
Higher Privileges   Login Script   Password Used
Home Directory   Login Time   Print Job Configuration
Language   Message Server   Private Key
Last Login Time   Minimum Account   Profile
Locked By Intruder   Balance   Public Key
Login Allowed Time Map   Network Address   Security Equals
Login Disabled   Network Address   Server Holds
Login Expiration Time   Restriction   Type Creator Map
Login Grace Limit   Password Allow   UID
                                       Change
                                     Password Allow
                                       Interval

Mandatory Attributes

The attributes listed in the mandatory attribute list for the object class are the attributes that are required. If an attribute is mandatory, every object created from the class must have at least one assigned value for each mandatory attribute.

Optional Attributes

The attributes listed in the optional attribute list for the object class are the attributes that are not required. If an attribute is listed as optional, then when the object is created the attributes may or may not be created.

Object Class Flags

There are five object class flags that are either set to ON or OFF. These object class flags include the container flag, effective status, and nonremovable flag among others.

Container Flag   The container flag is set to ON only for those object classes that are designated as the container object classes in the Directory tree. The container flag would be set to ON for the following object classes: Top, Country, Locality, Organization, and Organizational Unit.

The container flag is set to OFF for all the object classes that are the leaf objects in the schema or non-effective classes.

Effective and Non-Effective Status   Each object class is categorized as an effective or non-effective class. Effective classes are visible in the utilities and are used to create new objects in your NDS tree.

Non-effective object classes are used as super classes to build the other object class information. Therefore, non-effective classes are used only in the definition of other objects and not to create objects in the Directory tree. The non-effective object classes are not visible using the management utilities. Non-effective object classes in the schema are used to create other object classes in the Directory tree. (Later in this section where each object is listed is a notation of each object's effective or non-effective class.)

Nonremovable Flag   The nonremovable flag determines if the object class can be removed from the schema definitions. The nonremovable flag is set to ON for all the object classes that are defined in the base schema that is shipped with NetWare 4.1. The object classes added later to extend the schema have the nonremovable set to OFF.

PROPERTY OR ATTRIBUTE TYPE DEFINITIONS

All the properties or attributes found in the Directory consist of an attribute type and an attribute value. The attribute identifies the nature of the information that will be stored by the attribute and dictates the syntax used to enter the value.

An attribute definition is based on special constraints and on a specific attribute syntax that determines the values the attribute can store. There is a one-to-one relationship between attributes and their syntaxes. There are approximately 142 available attributes that are defined in the base NDS schema that ships with NetWare 4.1. Various combinations of these attributes apply to all object classes. You can view some of the available attributes or properties for each object through the use of the NWADMIN utility. The following is a list of the attributes defined for the Group object class:

- Access Control List

- Authority Revocation

- Auto Start

- Back Link

- Bindery Property

- CA Public Key

- Certificate Validity Interval

- ▸ Common Name

- ▸ Default Queue

- ▸ Department

- ▸ Description

- ▸ Desktop

- ▸ E-mail Address

- ▸ Equivalent To Me

- ▸ Full Name

- ▸ GID

- ▸ Last Referenced Time

- ▸ Location

- ▸ Mailbox ID

- ▸ Mailbox Location

- ▸ Member

- ▸ Object Class

- ▸ Obituary

- ▸ Organization

- ▸ Other Name (other values of common name)

- ▸ Owner

- Reference

- Revision

- See Also

The attribute type definition consists of Attribute Syntax and Attribute Constraints.

Attribute Syntax

The attribute syntaxes are data types for values that will be stored in the Directory. The attribute syntaxes are predefined. The NDS schema is extensible, as discussed in the next section, meaning that you can create new object classes and attributes based on the attribute syntaxes. You can also add attributes to existing class definitions. Although the schema is extensible, the attribute syntaxes cannot be alternated. There are currently 28 static attribute syntaxes defined in the schema. The following is a list of the attribute syntaxes that are defined in the schema:

- Back Link

- Boolean

- Case Exact String

- Case Ignore List

- Case Ignore String

- Class Name

- Counter

- Distinguished Name

- E-mail Address

- Facsimile Telephone Number

- ► Hold

- ► Integer

- ► Interval

- ► Net Address

- ► Numeric String

- ► Object ACL

- ► Octet List

- ► Octet String

- ► Path

- ► Postal Address

- ► Printable String

- ► Replica Pointer

- ► Stream

- ► Telephone Number

- ► Time

- ► Timestamp

- ► Typed Name

- ► Unknown

ATTRIBUTE CONSTRAINTS

Attribute constraints are all restrictions that affect the attribute values for each of the attributes. These constraints include whether the attribute values can have a single value or multiple values, whether the attribute values are immediately synchronized between replicas, or whether the attribute is hidden or nonremovable. The constraints also manage access control to the attribute value. For example, an attribute can be constrained to allow only someone with sufficient rights to the object to add an attribute.

The attribute syntax also dictates the matching rules that are used when comparing two values of the same syntax. The primary matching rules are: equality, substrings, and ordering. The equality checks for equal values of the attribute values and only applies if the attribute types are the same. It can't compare unless the attribute is the same. Therefore, the syntax would be the same. The substring compares two string attribute values where a wildcard "*" can be used. The ordering rules compare for "less than," "equal to," and "greater than."

For a complete definition of the attribute syntax IDs, data formats, and matching rules, please refer to Novell's Software Developer's Kit (SDK).

SCHEMA STRUCTURE

Each object class along with its properties or attributes and the attribute syntaxes form what is called the NDS Base Schema. As shown in Figure 3.5, the schema structure can be represented as a pyramid in which the foundation consists of the attribute syntaxes. These attribute syntaxes are used to define an attribute; in turn the attribute defines the object classes, and the object classes determine the schema structure.

THE NDS SCHEMA IS EXTENSIBLE

NDS provides a mechanism for making changes or extensions to the schema. The purpose of this section is to give an overview of how this can be accomplished in your NetWare 4.1 environment. For specific information on API functions and making API calls, see the NetWare Directory Services API Reference Manual from Novell. By using the client Application Programming Interface (API), the schema can be expanded beyond the base schema, which is provided by Novell. Attributes can be added to existing object classes, or new object classes or attributes can be created.

FIGURE 3.5

The schema structure is represented as a pyramid in which the foundation is the attribute syntaxes, which define the attributes. The attributes define the object classes, and the object classes determine the schema structure.
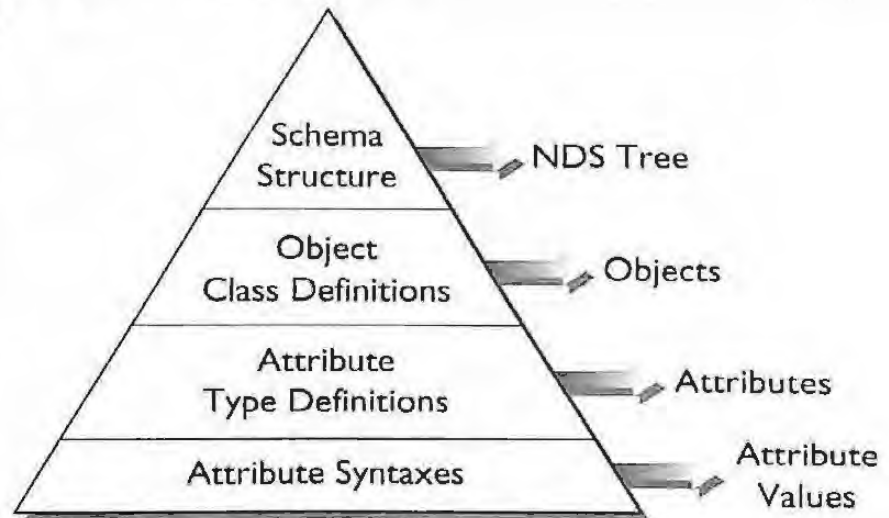
The first step to extending the schema is to create new attribute definitions and/or new object class definitions. The extensions to the schema can be accomplished through the use of programming, with Novell's API calls designed for this purpose. After completing this task, you have modified the schema. All changes or extensions are automatically replicated on all NetWare 4.1 servers in the Directory tree.

Third-party applications will also provide a means for extending the schema. Novell's NetWare Application Launcher, for example, will make an extension to the schema to define an Application icon. Prefixes for schema extensions can be registered with Novell Developer Support. This registry ensures that schema prefixes among vendors are not duplicated on the schema.

Any user performing modifications to the schema must have write rights on the [ROOT] object. Before you create a new attribute or object class, determine first if the attribute or object class already exists in the schema. If you cannot find the attribute or object class you need or if the class is missing an attribute, you can then take the steps necessary to extend the schema.

Each time there are modifications to the schema, these changes will be synchronized on all NetWare 4.1 servers in the tree. The synchronization process requires no administrator intervention as it is part of the NDS background processes that occur. The schema synchronization occurs through the use of the trickle-down algorithm because the synchronization occurs from the top and proceeds downward.

Therefore, the schema updates will propagate across replicas of the same partition and downward in the tree structure from the [ROOT].

The trickle-down method causes a NetWare 4.1 server receiving an updated schema to build a distribution list in order to update servers that it knows about. It will then check all servers in its distribution list to determine if they have received an updated copy of the schema. If they have not received a copy, that server will send an updated copy of the latest schema to other servers. This process extends only down and across, not up.

NDS Objects and Their Uses

You will discover if you haven't already that some NDS objects will be used much more than others. Obviously, the User object will be the most commonly used object in your tree. Some other leaf objects may never be used in your tree. That's okay. There are probably no NetWare 4.1 networks that will have a need for all the objects defined in NDS. But, they are in the schema for a general audience that may need them at some point.

To make your job easier, we have first grouped the objects in the following sections according to how they are used.

- ▶ Required Objects — These are the objects that either you must define or are defined for you during the installation of NetWare 4.1.

- ▶ Commonly Used Objects — These objects include both the container and non-container objects along with examples of how the objects can be used. These examples are based on actual consulting experiences.

- ▶ Less Commonly Used Objects — These are the objects that may have less importance for you. Where appropriate we have listed consulting recommendations with these objects as well.

REQUIRED OBJECTS

The following objects are required in your NDS tree. When you install your first NetWare 4.1 server, these objects will be present in your NDS tree. All of these objects will require you to name them at some point in the NetWare 4.1 installation.

As shown in Figure 3.6 of Novell's installation utility, you will be prompted for a name of the tree, organization, and organizational units. You can also create a Country object, which is not mandatory.

Novell's Install utility requires you to name all required objects during the installation.

```
NetWare Server Installation 4.1                    NetWare Loadable Module

                         Installation Options

                       Context For This Server

  Company or Organization:                        ACME
  Level 1 Sub-Organizational Unit (optional): CAMELOT
  Level 2 Sub-Organizational Unit (optional):
  Level 3 Sub-Organizational Unit (optional):

  Server Context: OU=CAMELOT.O=ACME


  Level 2 Organizational Unit Help

  Type your department (or other suborganization) abbreviation or name (keep
  it short).  You may leave this field blank if not applicable.  For example:
  If your company is XYZ Inc., and this server is to be used by the
                      (To scroll, <F7>-up <F8>-down)
Save this context name and continue <F10>          Previous screen    <Esc>
Help                                    <F1>       Abort INSTALL <Alt><F10>
```

Top and [ROOT]

Effective Class

Purpose   The Top object class is the super class of all other object classes in the schema. This object is also the container object for all other objects in the Directory hierarchy. The Top class is an effective class object. The only instance of the Top object is the [ROOT] object.

The [ROOT] object is at the top of your inverted tree structure. The [ROOT], from a visual standpoint in the utilities, is the starting point of your tree and branches downward as shown in Figure 3.7. The name of the [ROOT] object is your tree name, although this name is not indicated or displayed in the utilities. When you install the first NetWare 4.1 server in a tree, you are prompted for a tree name, which means that you actually assign a name to the [ROOT] object. Each tree will contain one and only one [ROOT] object.

FIGURE 3.7

ACME tree structure shows
the [ROOT] object at the top
of the inverted tree structure.
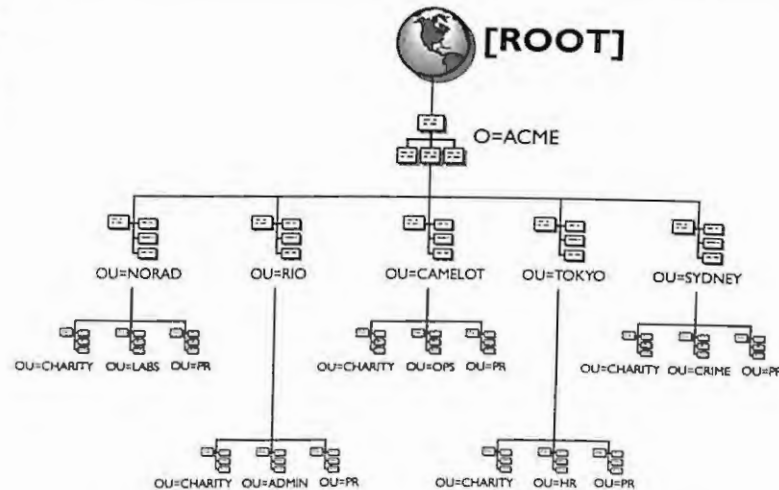
The [ROOT] object's name, which is the tree name, is broadcast on your
network using the Service Advertising Protocol. If you have multiple trees running
on the same network infrastructure, you must ensure that the names are unique.

Sample Uses Choose a name for [ROOT] that will clearly identify the
organization or company for the tree. Keep in mind that renaming the [ROOT]
object or tree name can only be done with the DSMERGE.

CONSULTING EXPERIENCE

Most companies simply use the same name as their company or Organization object
and add an _TREE. For example, our tree name for ACME is ACME_TREE. The organ-
ization name is O=ACME. You want to choose a name that you can recognize as the
tree name if you are using software analysis tools on your network and looking at server-
to-server communications. Figure 3.8 illustrates how the NDS tree for ACME has been named.

Example Uses Again, [ROOT] is the only instance of the Top object class in the NDS
tree. However, in the schema, all object classes must have the super class of Top defined,
including any new classes created by extending the schema. Top is the only object that

has no super class. All object classes inherit from the class Top. All object classes that you may define through schema extensions must contain the Top class.

FIGURE 3.8

The NDS tree for ACME is named ACME_TREE.
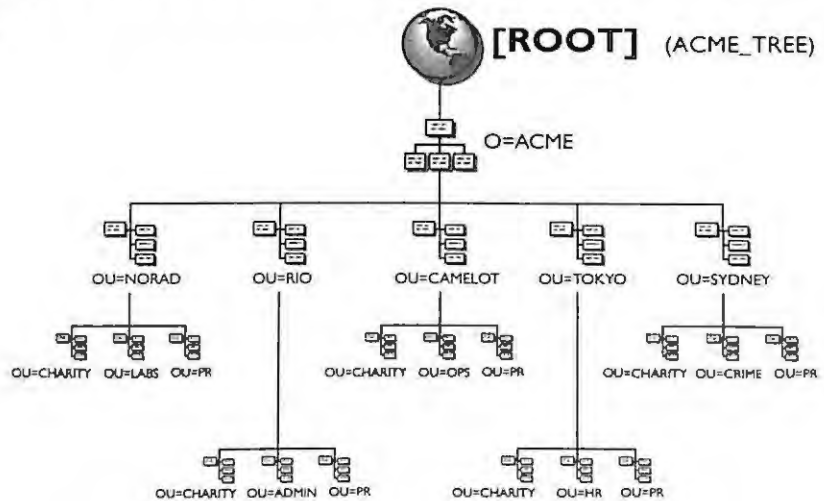
Organization

Effective Class

Purpose This object class is used to define organization objects in the tree. An organization is located directly under the [ROOT] object or C=Country object. This object is used to define the name of a company, and you are required to define at least one organization in your NDS tree. Multiple O=Organization objects may be used directly below [ROOT] or Country as shown in Figure 3.9.

FIGURE 3.9

Multiple O=Organization objects defined for a large conglomerate company
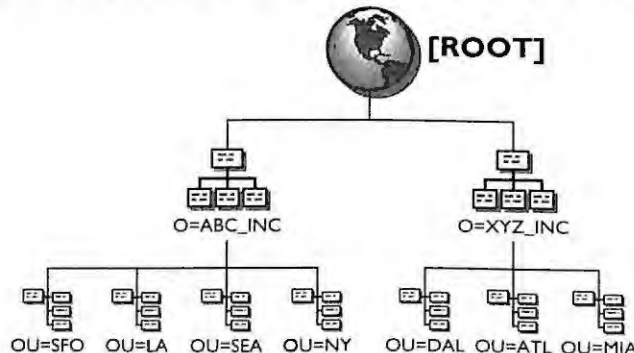
Sample Uses   The ACME tree is defined as the name of its organization O=ACME. The organization represents the name of your company or organization and should be an overall descriptor of the business. If you are a member of a nonprofit group or university, the name of your Organization object can be the same name as your university or group.

CONSULTING EXPERIENCE

Most companies typically define only one Organization object in their NDS tree. If all your business units are connected with the same network, then a single Organization object more accurately represents your network. Our guideline for using multiple Organization objects is based on the network infrastructure in which you operate. For example, if you are a large conglomerate with varied businesses that have decided to communicate information together using NetWare 4.1 for the exchange of information, you can use multiple Organization objects as shown in Figure 3.9 above. If each company is separately managed with a separate network infrastructure, you can create multiple Organization objects for each business unit under the [ROOT] object. You can also use the Organizational Unit objects to represent your business units.

For more information on designing multiple Organization objects, see Chapter 5.

User

Effective Class

Purpose   Obviously, the most common object in your tree, the User object represents every user that is part of your NetWare 4.1 network. The User object is similar to the user object found in NetWare 3 environments but contains more attributes.

Sample Uses   During the installation of your first NetWare 4.1 server, you will be prompted to assign a password for the first user ADMIN created on the network. This user, named ADMIN, has object Supervisor rights assigned to the [ROOT] object of your tree. The ADMIN user has all rights (NDS and File System) initially to the entire

tree. At this point in your installation, it is the only object with such complete and extensive access to your network. The importance of maintaining the ADMIN user or another object with the same rights at [ROOT] is well known. If you should delete this object without making other rights assignments, say to an organizational role, your access to the tree is lost. Your options in this situation would be to reinstall NDS or call Novell Technical Support to provide you with reentrance into your tree.

CONSULTING EXPERIENCE

After you have installed the first couple of NetWare 4.1 servers, ensure that the ADMIN password is protected. Following these steps will diminish the likelihood of losing access to your tree:

1 • Your first NetWare 4.1 server installation will prompt you for a password for the ADMIN user. Remember that the first password you assign to the NDS for ADMIN is also assigned to the bindery Supervisor object created by the install utility. If you are doing a bindery upgrade, the supervisor's password is taken from the bindery. If the bindery supervisor's password does not exist, then the NDS ADMIN password is used. If you later change the ADMIN user's password, the bindery Supervisor password does not change. Choose a password that will not easily be guessed. Change your passwords periodically for greater security. The ADMIN user password is changed through NWADMIN or NETADMIN utilities. The bindery Supervisor password can be changed through the NetWare 3 SYSCON utility.

2 • Create an organizational role in the O=Organization level of your tree and assign this role object Supervisor rights at the [ROOT] object of your tree.

3 • Do not make this organizational role object a security equivalent to the original ADMIN user. If the original ADMIN were to be deleted, your organizational role would have no access to the tree because the security equivalency would be lost. For more information on how to define security for organizational roles, refer to Chapter 13.

NCP Server

Effective Class

Purpose A NetWare (NCP) server object will automatically be created for any server being installed or upgraded to NetWare 4.1. The server object holds key information such as the network address and the version of NetWare you are running on that machine. This object type is used to represent any server that provides NCP (NetWare Core Protocol) transport and session services and can represent either bindery or NDS-based NCP servers. This subclass of the Server object provides further definition of NCP services available on a particular server.

Sample Uses The supported Services attribute can be used to list NCP-based features and services available for this network address. As part of an NDS search engine, you can search for available NCP services.

Another useful feature is the Operator attribute, which is used by an NCP server as an access control list. If a particular object is part of this access control list, that object can perform remote console operations. These remote operations do not mean using RCONSOLE; they have the capability to exercise the console APIs, which RCONSOLE does not use.

The server object is basically managed by NDS and requires little attention from the administrator.

CONSULTING EXPERIENCE

Be sure to follow the naming standards presented in Chapter 4. This is a resource on which you will no doubt want to perform Directory queries. Consistent and appropriate naming will help this process considerably.

Volume

Effective Class

Purpose   The server volume object is automatically created when you install a NetWare 4.1 server into the tree. A file server must have at least one volume

called the SYS volume. The server may have additional volumes that are defined during the installation of NetWare 4.1. A volume object will be created for each mounted volume during the installation. The naming convention is <file server name> _ <volume name>.

Sample Uses   This object type exists primarily to differentiate it from other types of resource objects and to allow greater flexibility for volume management if needed. It is also used in MAP commands and is required by the FILER utility to grant file system rights.

CONSULTING EXPERIENCE

NetWare 4.1 gives you flexibility to place print queues on any NetWare 4.1 volume. We recommend that you always create at least one other volume in addition to the SYS volume so that your print queue will not fill up the SYS volume and cause NDS to become disabled.

Use the volume restrictions to limit the size of your volumes before they fill up completely. You can also limit the disk space used on the Home Directory if you are short on space or want to limit each user on the volume.

COMMONLY USED OBJECTS
Below is a list of the most commonly used NDS objects. Most, if not all, of these objects will be used in your NDS tree. Small sites with perhaps one server may not use all of these objects. In fact, it is only necessary to make use of the objects that will best serve the purposes of your network environment.

Bindery Object

Effective Class

Purpose   This leaf object represents any object other than user, group, queue, profile, and print server created through bindery services. The bindery object has the format of Common Name + Object Type and is used to provide backward compatibility for bindery-oriented applications or utilities.

Sample Uses   You will see bindery objects appear after a migration from NetWare 3 to NetWare 4.1 if an object is not identifiable by NDS. Some applications will create bindery objects to suite their particular purposes. For applications that still require the bindery services, a bindery object could exist in the tree. Although these objects appear in utilities such as NWADMIN, they are nonmanageable from NWADMIN or NETADMIN and are present only for information purposes.

CONSULTING EXPERIENCE

Do not delete any bindery object until you verify its purpose in the Directory. Some installation utilities will create a bindery object that if removed will cause problems for the application. Always check first before deleting bindery objects that appear in your containers.

Organizational Unit

Effective Class
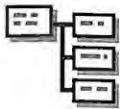
Purpose   Nearly all NDS trees, except environments with a single NetWare 4.1 server, will use Organizational Units. They are, however, optional. Organizational Units subdivide the tree into either locations or organizations, which could be departments, divisions, or workgroups. The Organizational Unit is also referred to as a container and contains the login script property, also known as the container login script. All users in a container will execute that container's script if it is available.

Sample Uses   As shown in Figure 3.10 the ACME tree uses containers to make logical divisions in the tree. Notice that the first level of OU's represent geographic sites based on the network infrastructure, and the subsequent levels represent the departments of ACME. Many trees will be designed in this fashion with several levels of nested organizational units. For more information on designing a NetWare 4.1 tree, see Chapter 5.

The use of organizational units makes NDS tree subdivision clear and logical.
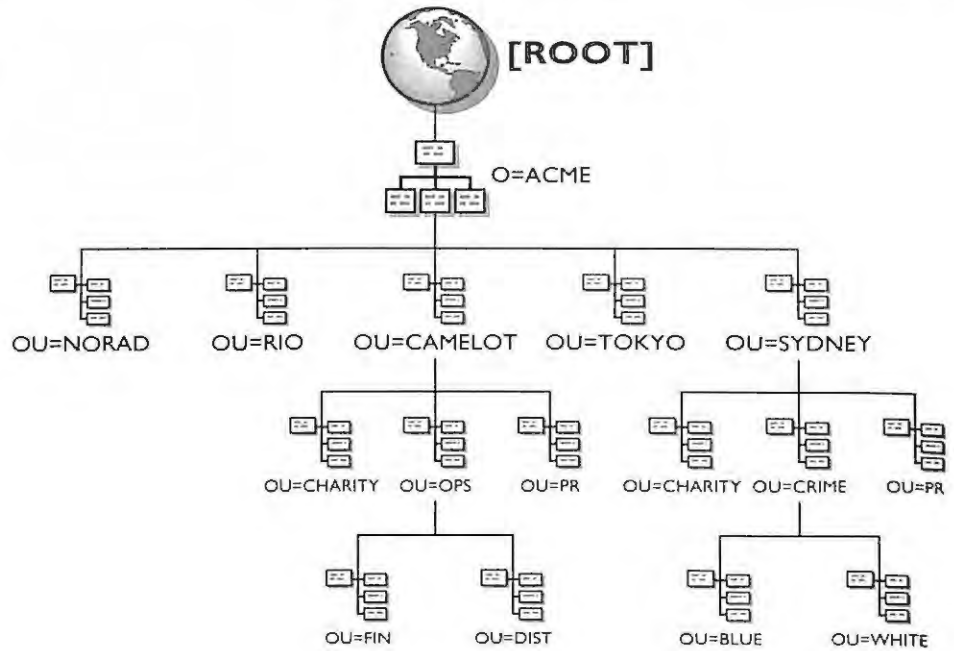
CONSULTING EXPERIENCE

Most companies use the Organizational Unit to represent geographic locations, divisions, and workgroups. You generally use an Organizational Unit to group common users and resources in the same container. Organizational Units should not be created to represent a single person or a single resource. They are used to provide access to a group of users in a particular part of the tree. NetWare 4.1 does provide the capability to move containers to new locations in the tree should your needs change.

Organizational Role

Effective Class

Purpose   An Organizational Role object is typically used to define a role or position within an organization object or container object. The role is extremely

useful because the object or file rights are granted to the role itself and not to the occupants who may belong to the role.

NOTE

However, this rule is different for object creation. For example, an object is created by a user assigned to the organizational role. The object creation rights to the new object go to the user creating the object, not to the organizational role.

Sample Uses   Organizational Roles are especially useful in maintaining subadministrators. The occupant can be moved in and out of the role quickly to facilitate short-term assignments. For example, if the regular administrator is absent for any length of time, another user can be moved into the role temporarily to manage the network. The implementation of this object is just like a group. The organizational role occupants derive their rights through security equivalence to the role object.

As shown in Figure 3.11, we have created an organizational role for each city contained in the ACME tree. This administrator will have rights to manage their entire container, such as adding and deleting users or servers.

FIGURE 3.11

Organizational Roles created for the ACME tree
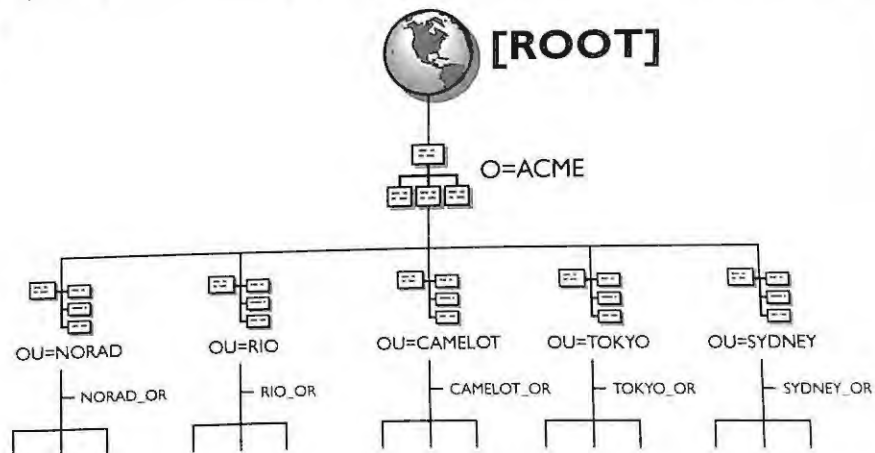
CONSULTING EXPERIENCE

The use of an organizational role is highly recommended as a way to administer rights in your network to various administrators in your network. You can more easily track who has received rights as administrators in your tree by checking who has been made an occupant of a role. Assigning rights to individuals through means other than the organizational role is not recommended because it is more difficult to track which individuals have been granted rights in your tree. After a few months or even weeks, you'll forget who has been assigned rights. For more details on setting up an organizational role for your tree, see Chapter 13.

Group

Effective Class

Purpose   This class is used to represent a set of users from any part of the NDS tree. The membership of the group is static and is only modified by your administrative action.

Groups function the same way they did in NetWare 3. There is very little difference between using group objects and organizational unit (OU) objects. Both have the same function, which is to place common users close together. It is true that users who are members of both objects receive rights by security equivalence, but there are some differences. Because of security equivalency, any member of an OU will receive whatever rights the OU possesses. Users inside groups also receive whatever rights the group possesses, and just like a container there is no IRF provision. For more information on NDS security refer to Chapter 13.

Groups are used to differentiate rights within a particular Organizational Unit. Rather than create multiple subcontainers, you can simply use groups within a container for users that need a specific environment created.

Sample Use   In order to provide rights to a smaller subset of users in the CHARITY.SYDNEY.ACME OU, you can use a group. Figure 3.12 shows an organizational unit called OU=Charity and is populated with users and resources. Within this OU are two sets of users. Each group is accessing different software on

the server and needs different rights assignments. Within the container or OU login script (known as the system login script in NetWare 3), the IF MEMBER OF GROUP statements are used to determine which group a user belongs to for rights assignments. When each user logs into the network, the login script determines if a user is a member of a group and sets the appropriate environment variables.

CONSULTING EXPERIENCE

Groups are great to use in a container to provide further differentiation of rights to a subset of users. We recommend that groups be used in a container instead of creating more containers. Additional containers would require the use of additional container login scripts. Limit the number of groups to fewer than 15 in a container login script, if possible. The more groups, the longer the container login script will be, and a slower login will occur. However, if you have 15 or more groups you should create additional containers because the login script must check your users against each group membership list. Groups should be contained locally. Avoid the use of groups that contain users from multiple organizational units.

FIGURE 3.12

The Organization Unit Charity with two groups created to differentiate users within the same container

[ROOT]

O=ACME

OU=NORAD OU=RIO OU=CAMELOT OU=TOKYO OU=SYDNEY

OU=CHARITY OU=HR OU=PR

SIRLANCELOT

WP_GROUP

HPSI GROUP
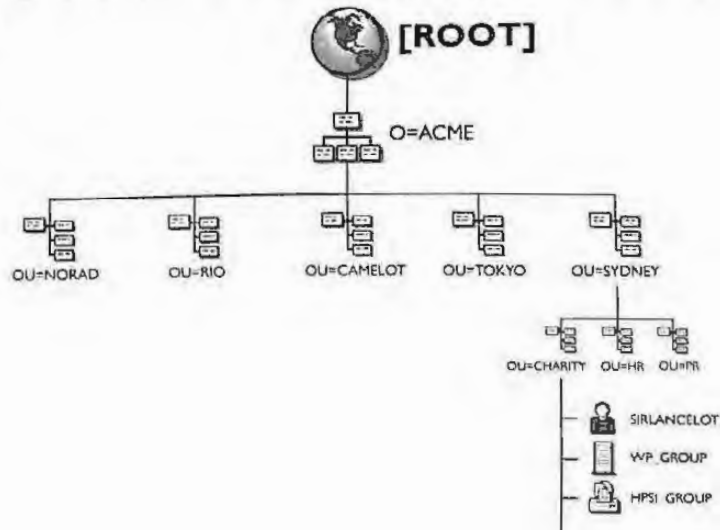
Directory Map

Effective Class

Purpose   The Directory Map object is a pointer that refers to a file system directory on a NetWare volume. It is used in login scripts or by the MAP command to point to a directory that contains a particular application.

Sample Use   Let's assume you have WordPerfect installed in a directory called WP60. If you were to upgrade your software to WordPerfect 6.1, you could choose to rename the subdirectory to WP61. You would also have to change every login script to reflect this change for the new subdirectory. Through the use of a directory map you can eliminate making changes to your login scripts. Your directory map object is used in your login scripts and points to a subdirectory on a NetWare volume. Your only change is to have the directory map point to WP61. All container login scripts are left intact.

CONSULTING EXPERIENCE

Directory map objects are useful for providing a standard container login script on multiple servers. We recommend that you have standard file structures for all your NetWare 4.1 file servers so that you can facilitate easier use of directory maps. If you maintain a directory map for an application, you can make a change to the directory map pointer and have it be effective for many servers. The directory map requires the administrator to make each user security equivalent to the map object and for containers with large numbers of users. This could be a drawback from an administration standpoint. You can, however, grant rights on the directory pointed to by the directory map to the container to simplify this situation.

Alias

Effective Class

Purpose An alias is another name for an object. An alias is an object that points to another object you specify in the Directory tree. An alias can point to either a container object (an object that holds other containers) or a non-container object (one that does not have any other containers).

Sample Uses An alias is a name containing at least one Relative Distinguished Name. As a network administrator you may grant users access to a particular resource contained in another OU, such as a printer. You can create an alias to reference that printer, for example. The alias can be considered a relay to another object in a different part of the tree.

You can also alias one OU to another OU, giving one OU to the other OU's resources. It appears as if the alias places the alias's container inside of the other container.

The object being aliased is known as the primary object. When you create an alias object you may or may not name it with some indication that it is an alias to another object. For example, the name might include the word "alias," such as Alias_Blair.

CONSULTING EXPERIENCE

The alias object is very useful in moving subtrees or renaming containers so that you can easily make the transition. The alias can assist you in this migration path. In addition, the alias can be useful for some companies wanting to create mobile users who do not want to remember their user context. The alias can be created at the top of the tree below the Organization object, for example, to shorten the user's context. If user David wants to log in to the network he only has to remember that his context is David.Acme because that alias points to his actual context in the tree. This example is shown in Figure 3.13. Keep in mind that this example refers to a very limited number of users, such as mobile users. We are not implying that you alias every user in your tree.

FIGURE 3.13

Mobile user David has a shortened context through the use of aliasing.
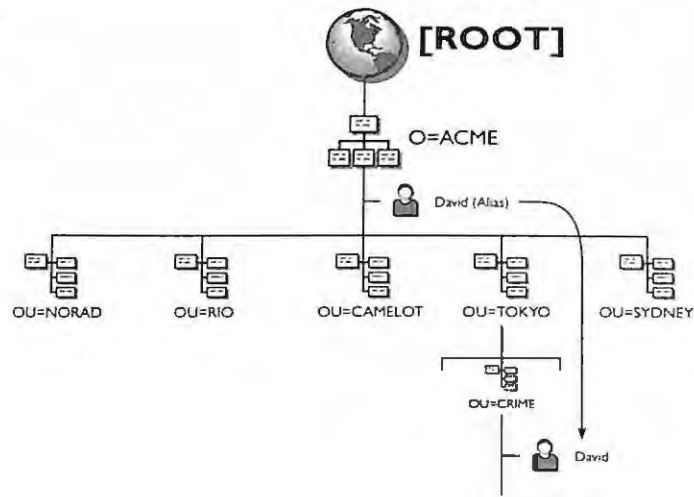
Print Server

Effective Class

Purpose   The print server leaf object represents a server that takes print jobs out of a print queue and sends them to a network printer. This object is used in conjunction with the Printer and Print Queue objects. A print server object must be created for every actual print server that exists on the NetWare 4.1 network.

Sample Uses   All NetWare 4.1 networks using printers will have at least one print server object. You can use this object to define which printers to assign to users based on their position or location in the NDS tree. You can also define the print server operators.

Printer

Effective Class

Purpose   The Printer object is used in conjunction with the print server object. You use this object to manage a printer. The assignment for a print queue can be made within the object.

Sample Uses   You can attach printers in several different ways to the network, namely, directly to the network, to a printer port of a NetWare server, or to a printer port of a PC.

CONSULTING EXPERIENCE

You can place the Printer object along with the Print Queue object as high as possible in your NDS tree to support many users. If your printer is for only your container, then place the printer in your container. If more containers need access to the printer, place the object in the next highest level in your tree. Keep in mind that the placement of printers higher in the tree also adds overhead to NDS in terms of name resolution. We also recommend that you configure printers as network-direct printers in queue server mode as explained in Chapter 14.

Queue

Effective Class

Purpose   This object represents a print queue defined on a NetWare 4.1 server. The queue actually represents the directories where the print jobs are sent to be serviced by a printer.

Sample Uses   Place Print Queue object along with the Print object at the highest level possible in your tree to service the most users.

CONSULTING EXPERIENCE

All NetWare 4.1 print queues should be placed on a volume other than the SYS volume for greater fault tolerance of the server. A queue object must be created for NetWare 4.1 printing. A Print Queue object is assigned to a printer.

Profile

Effective Class

Purpose   The Profile object is used as a special purpose scripting object that is executed by LOGIN.EXE and is a shared login script after the execution of your container login script. The profile script can contain special drive mappings or environment settings you want a select group of people to receive.

One of the properties of a user object is the profile. When a user is first created using the NWADMIN utility, you can specify that the user be part of a profile. You can also add a user to a profile anytime after by going back and adding the profile to a user object's properties.

Sample Uses   There are three notable uses for a profile script:

► Creating a global login script

► Creating a location login script

► Creating a special function login script

Creating Global Login Scripts

NetWare 4 does not use a global system login script. Each Organizational Unit created will have its own login script referred to as the OU login script. The order of execution of login scripts is:

1 • OU login script

2 • Profile login script, if used

3 • User login script, or

4 • Default login script if no other script is available

Therefore, if you want to create a more global login script and include multiple Organizational Units, you could employ the profile object to set up a specific environment for a group of users. Keep in mind that this type of solution is expensive in terms of NDS overhead that will be created on your network.

Location Login Scripts

A profile can also be used for determining allocation of resources based on location. For example, each department of a company may have three printers and three print queues. A profile enables you to assign a particular group of users to a specific print queue. You can use a profile login script to capture to a particular print queue and the users will automatically capture to that print queue.

Special Function Scripts

A profile can be used as a special function script to assign users access for applications. For example, you can create a profile script that will be used by only administrators. This script may give these users a specific drive assignment to a help desk utility. In this scenario, you would move the help desk utility out of the SYS:Public directory into a new subdirectory you create called HELPDESK. When a user logs into the network, the admin Profile object is executed, and the user is assigned a drive mapping to the HELPDESK directory. Only users who execute the profile script will be assigned rights to access the help desk utility.

CONSULTING EXPERIENCE

Typically, the profile object is used as a global login script for a workgroup or organization. Keep in mind that the profile object will execute from one container, although users may be participating from other containers and may see some performance issues if the script is executing over a wide-area connection. In addition, you will notice increased overhead of NDS traffic as NDS searches the tree to locate a particular object.

Unknown

Effective Class

Purpose   This object class represents any object created by the server to restore any object whose base class is currently undefined in the schema. Objects of this class type are created by only the server, not the client, and usually occur when a mandatory property of an object has been lost.

Unknown objects are created during synchronization of objects as placeholders, when a mandatory attribute of an object has been lost, and when an object class has been deleted and the server had an object of that class.

Sample Uses   This object is automatically created by NDS. If an unknown object appears in your tree, check to see if some other object has been deleted. In the case of a deleted server object, the server's volume objects will appear as unknown objects and must be deleted by an administrator. In other circumstances an unknown object may appear temporarily as the NDS background processes work through a synchronization process.

CONSULTING EXPERIENCE

Before deleting an unknown object, first check to make sure it is not needed. When you have verified that the object is not needed, then remove it from your tree. There is no point in maintaining unknown and unused objects.

LESS COMMONLY USED OBJECTS

AFP Server

Effective Class

Purpose   The AFP Server object represents an Appletalk Filing Protocol-based server that is part of your NetWare 4.1 network as a server node or router for connected Macintosh workstations.

Sample Use   This object serves only as a descriptor object and does not provide any management capability for your AFP server in a NetWare 4.1 environment. You can use this object for informational purposes about Appletalk in your NetWare 4.1 environment.

Bindery Queue

Effective Class

Purpose   The Bindery Queue object represents a leaf object queue that has been placed in the Directory through an upgrade or migration to NetWare 4.1.

Sample Use   A bindery queue is used to support other queues upgraded from NetWare 3.

Country

Effective Class

Purpose   This object is used to define country entries in your NDS tree. The Country Name attribute is restricted to two characters as defined by the ISO3166 standard. Directory Services does not check to see if characters are correctly defined, just if there are two.

Sample Uses   For companies desiring to use the Country object, there are several considerations. First, the object must be placed directly below the [ROOT] object. Specifying this option during installation of NetWare 4.1 is shown in Chapter 16. Second, consider the ramifications of adding the Country object to your tree in terms of adding another layer to your tree and lengthening your users' context. Also, carefully plan your tree design so that your users' context will be logical and make sense. For example, let's assume we added the Country object to the ACME tree under [ROOT]. Since our company is global in nature, which country do we choose? If we say that C=US, notice what the context would be for users not reporting under the US region. This example is shown in Figure 3.14.

CONSULTING EXPERIENCE

Our recommendation is to avoid using the Country object in your tree. Even with emerging gateway technology to interconnect disparate directory databases, the gateway will handle the distinction between databases using and not using the Country object. The Country object is also not necessary for companies wanting to connect to a public data provider for Directory Services.

FIGURE 3.14

The Country object can cause your users' context to look different than the context you are actually trying to represent.
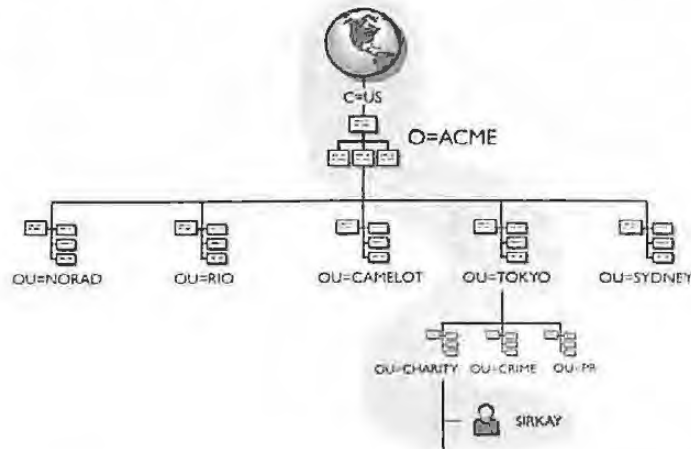
Distinguished name: CN=SIRKAY.OU=CHARITY.OU=TOKYO.O=ACME.C=US

CommExec

Effective Class

Purpose The CommExec object is used to manage Novell's NetWare for SAA NetWare Loadable Module on a NetWare 4.1 Server.

Sample Use This object is used only with NetWare SAA and provides management capabilities and rights privileges for the [ROOT] level of your NDS tree.

Locality

Effective Class

Purpose The Locality object is used to define geographic locations in the NDS tree such as states or regions or counties. The Locality object is currently not enabled by Novell's NWADMIN or NETADMIN utilities and therefore is not visible through the current utilities. Programs are available to define and view this object class.

Sample Uses   This object must be named by one or both of its attributes L=Locality Name or S=State or Province Name. Most companies will not need to use this object. Some Directory Services that provide connection services through NDS will use the Locality object.

CONSULTING EXPERIENCE

The Locality object exists in the NDS schema.

Computer

Effective Class

Purpose   This object class can represent both computers used as NetWare servers and computers used as client workstations.

Sample Use   A key attribute of this object is the Operator, which can be used to identify individuals or groups that typically handle the day-to-day hardware maintenance of this computer.

Device

Non-Effective Class

Purpose   This object subclass is used to represent a physical device such as a modem or printer that you want to have defined in your tree.

Sample Uses   One attribute contained in this object is the Locality Name, which can be used to identify the physical location of a device. In addition, as more devices become NDS enabled, the Device object will possibly come into greater use for managing those objects.

External Entity

Effective Class

Purpose This object is used to store information about non-native NDS objects in the Directory tree.

Sample Use Some situations may require that an NDS object have information about another object that does not exist on the Directory Tree. A messaging service, for example, can use the External Entity class to store information about e-mail users who exist on other systems outside of the NetWare 4.1 tree.

List

Effective Class

Purpose This object is used to represent an unordered set of object names in the Directory. It could be a list of leaf objects or other objects that you want to logically group together based on some type of search criteria.

Sample Use The Member attribute is used to define the objects that are members of the list. The members can be any individual objects, even including Group objects. However, the key difference between this object and a group object is that membership in a list does not imply security equivalence as it does in a group. This object can be used to logically list objects for NDS searches.

Message Routing Group

Effective Class

Purpose Closely related to the Messaging Server, the Message Routing Group object is used to represent a group of messaging servers that communicate with each other to transfer messages.

Sample Use The Member attribute is used to define the messaging servers that belong to the Message Routing Group.

Messaging Server

Effective Class

Purpose   This object is used to represent messaging servers such as Novell's MHS servers that may exist in the Directory tree.

Sample Uses   If you are using Novell's MHS services for NetWare 4.1 you will use this object. Important attributes such as the Message Routing Group, Messaging Server Type, and Supported Gateway will be configured to define the types of services provided for e-mail communications on your network.

Organizational Person

Non-Effective Class

Purpose   This object defines anyone who either represents or is in some way associated with a particular organization, such as an employee. This object is part of two subclasses defined in the X.500 standard: Organizational Person and Residential Person.

Sample Uses   The User object is a subclass of the Organizational Person. The user class inherits from the organizational person.

Person

Non-Effective Class

Purpose   This object contains the more common attributes of the Organizational Person and Residential Person objects.

Sample Uses   The X.500 specification defines two subclasses of Person: Organization Person and Residential Person. The current NetWare 4.1 schema does not include Residential Person. The separation of Person from Organizational Person has been done for future compatibility with x.500.

Resource

Non-Effective Class

Purpose This object class is used to identify the logical resources available on your network. The resource class is similar to the device class in that a device is a physical unit, and a resource is a nonphysical or logical unit.

Sample Uses A very useful attribute in this object is the Host Resource Name, which can be used if a host's local identification differs from a more global resource identification. If a resource is being used by multiple localities you can define a Locality name, Organization name, and Organization Unit name as separate identifiers. If you define appropriate values, you can initiate NDS searches for a particular resource name, locality, or organization.

This chapter has presented a basis for understanding and using NDS objects in your tree along with the schema structure, which is the foundation of NetWare Directory Services. Extensions can be made to the schema to add new object classes or to add properties to existing object classes.

With this information you will be able to create a useful naming standard and to define appropriate objects for use within your tree.

Directory Services Naming Conventions

"It is impossible that every particular thing should have a distinct peculiar name."
John Locke

Closely tied to the NDS structure is the concept of naming. Object naming is fundamental to the Directory because it provides a definition of the objects in the tree as well as their relationship to the other objects in the tree. In addition, resource searching, a key function of the NDS name service, is provided through the use of object naming within the Directory. Good naming implies improved search capabilities now and in the future as more applications take advantage of NDS capabilities.

NetWare 4.1 uses a set of rules known as the Directory schema to define the naming hierarchy for the entire NetWare 4.1 network. The schema is replicated across all NetWare 4.1 servers that exist in the same tree. By contrast, the NetWare 3 bindery is a flat naming structure that is defined by the individual NetWare 3 server and does not provide a hierarchy of bindery objects. A visual example of this concept is shown in Figure 4.1.

FIGURE 4.1

A comparison between the NetWare 3 bindery and the NetWare 4.1 Directory
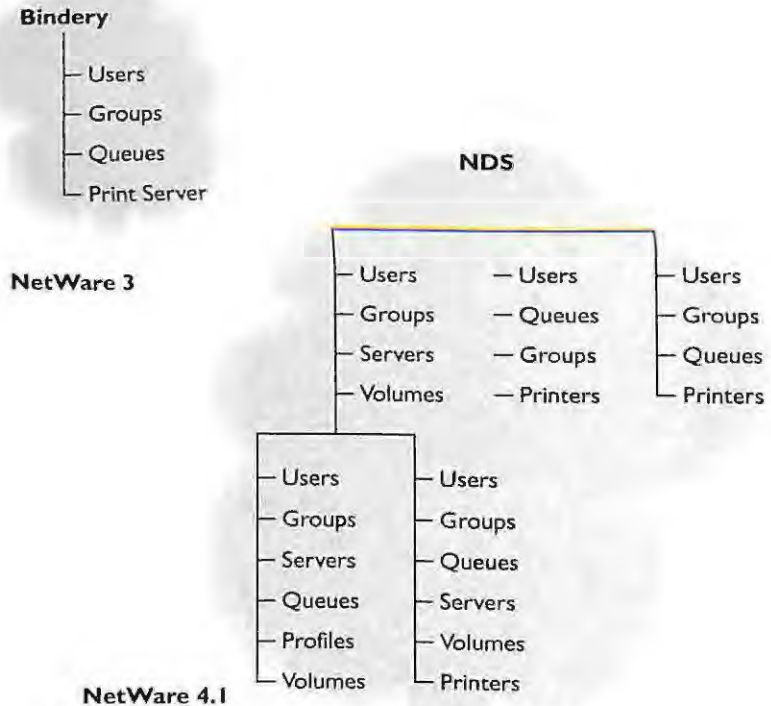
This section will first consider the basic elements provided within NDS for searching and locating objects within the Directory. These elements include the concepts of name types, distinguished names, relative distinguished names, contexts, typeful and typeless naming, and other methods of naming.

Next we will discuss how to create a naming standard for your particular needs and will include sample standards that can be implemented at your organization.

NDS Naming Rules

NDS is a collection of objects that follows a set of rules regarding their creation, naming, and use. Becoming familiar with how each of the NDS objects and properties are used in building the NDS tree structure and how they should be named will improve your NDS tree design.

Your understanding of the schema will help you to understand the internal rules of the objects, including the name types. Also, your understanding of the schema will help you to determine the necessity of creating additional objects through schema extensions or through the use of existing objects and properties.

Next, we will explain briefly the NDS schema — its components, rules, and how the NDS objects are created. For a more detailed understanding of the NDS schema refer to Chapter 3.

NDS SCHEMA

NDS consists of objects and properties that are defined by a set of rules in the Directory called the schema. The schema rules dictate the naming and relationship of the objects and properties that can exist in the Directory.

The NDS schema is automatically stored on every NetWare 4.1 server you install, and any updates to the schema are automatically synchronized to all NetWare 4.1 servers.

The schema consists of three major components: the object class, the property definitions, and the property syntaxes. These three components work together to establish the rules that control the creation of a particular object type in the Directory. For instance, each object is defined in terms of the attribute definitions and other class definitions.

The attribute definitions, in turn, are defined in terms of the attribute syntaxes. The attribute syntaxes are the data types of the attribute values. Figure 4.2 illustrates the relationship between the components of the schema.

FIGURE 4.2

Relationship between the three components of the schema: property syntaxes, property definitions, and objects

Object Class

Property Definitions

Property Syntaxes
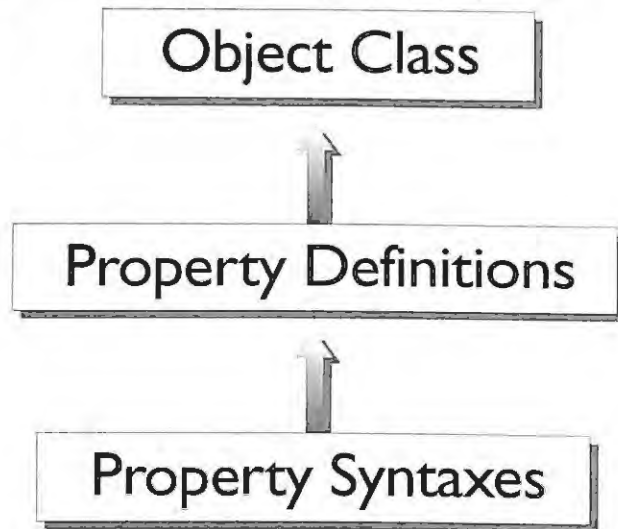
The object class defines the types of NDS objects that can be stored in the Directory. The object class is used as the rules for creating the individual objects and determining that object's set of characteristics. This means that every object in the tree belongs to an object class that specifies what properties are used.

The base schema that ships with NetWare 4.1 contains approximately 32 object classes. These object classes cannot be removed or deleted from the schema.

Each of the object classes in the schema is defined by the relationship of the object's containment in the Directory tree. This relationship is constructed on the basis of the containment class rules.

OBJECT NAME TYPES

Each object consists of a naming attribute and its value. The value is assigned by the administrator during the object creation in the Directory tree. Connecting the naming attribute with the value is an equal (=) sign. The naming attribute determines how the

object will be used in the Directory tree. Some objects are container objects while other objects are leaf objects (or non-container objects). Table 4.1 lists the object name attributes that are assigned to each of the NDS objects. Generally, they are abbreviated.

	NAME ATTRIBUTE	NAME DESCRIPTION
TABLE 4.1	C	Country Name
List of the object name types that are assigned to each of the NDS objects	L	Locality Name
	S	State or Province Name
	O	Organization Name
	OU	Organizational Unit Name
	CN	Common Name (includes all leaf objects)
	Bindery Type	Bindery Type

The name types C=Country Name, L=Locality Name, O=Organization Name, and OU=Organizational Unit Name are naming attributes of the container objects. The TOP class is also a container object (containing [ROOT]), although it is not visible by the Novell utilities. State or Province Name is not a base class.

CONSULTING EXPERIENCE

As you may have noticed from the Novell utilities, not all of the object class attributes of the container objects are available to you for use when building your NDS tree. For example, the class L=Locality Name and the S=State or Province Name do not show up in the current versions of the Novell utilities. These object classes are found in the NDS schema but have not been exposed by the utilities. The L=Locality class was placed in the schema for compatibility with the X.500 standard. When building the structure of your NDS tree, you can use the O=Organization and OU=Organizational Unit container classes, which provide you with the same functionality. The Locality container can be enabled through utilities such as the Preferred Systems' DS Standard utility.

The name type CN=Common Name includes all the leaf objects or non-container objects. Therefore, if the object is a user, printer, server, or any other leaf object, it is

named by the type CN. The following is a list of all the leaf objects you will use to represent the network resources in your NDS tree. These objects are all named with the "CN=".

- ▸ AFP Server

- ▸ Bindery

- ▸ Bindery Queue

- ▸ Computer

- ▸ Directory Map

- ▸ Group

- ▸ NetWare Server

- ▸ Organization Role

- ▸ Print Queue

- ▸ Print Server

- ▸ Printer

- ▸ Profile

- ▸ User

- ▸ Volume

Alias is not shown in this list because it is actually named by the object that it is referencing, such as an Organization, Organizational Unit, or Common Name.

Distinguished Name

Each object in the tree is given a distinguished name during its creation. An object's distinguished name consists of the name of the object plus the names of each of the container objects between itself and the [ROOT] object of the NDS tree. The naming of the object determines the location or position of the object in the tree. The object name attribute, mentioned in the previous section, can be used to help define the distinguished name by specifying the object type for the distinguished name. The distinguished name, therefore, provides a global, complete, and unique name in the Directory tree.

For example, in Figure 4.3 in the ACME tree, the distinguished name for the user Abraham Lincoln is the user object name plus the names of all the containers in which he is a member. It can be viewed as a path back to the top or [ROOT] of the tree. Thus, the distinguished name for the user Abraham Lincoln begins with his common name, CN=ALINCOLN, followed by each of the containers in which he is a member, OU=AUDIT.OU=ADMIN.OU=RIO.O=ACME, back to the [ROOT] object. Therefore, the distinguished name is read from left (lowest level in the tree Common Name) to right (highest level [ROOT]).

FIGURE 4.3

Each object in the ACME tree has a complete or distinguished name that is unique based on its object name plus the names of each of the container objects all the way back to [ROOT]. For example, CN=ALINCOLN.OU=AUDIT. OU=ADMIN.OU=RIO. O=ACME.
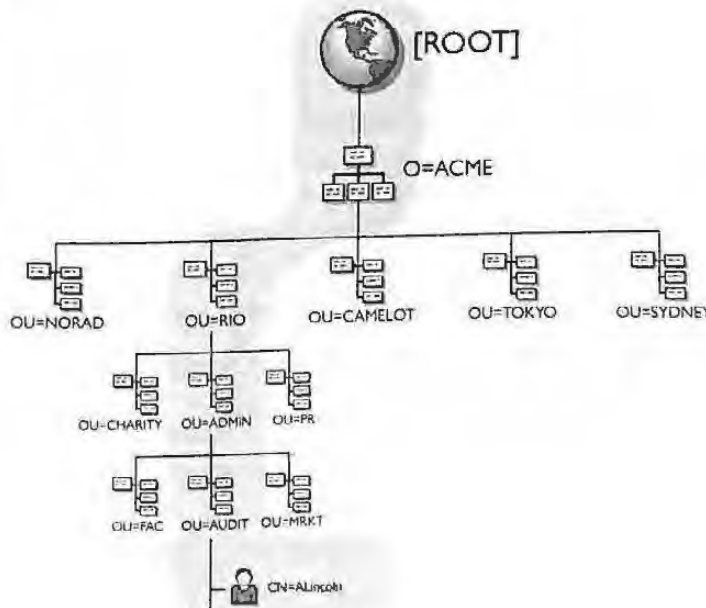
The sequence for writing or displaying the distinguished name of an object is as follows:

```
CN=ALINCOLN.OU=AUDIT.OU=ADMIN.OU=RIO.O=ACME
```

The distinguished name can be separated into segments. Each segment of the object's name is separated by periods. Each segment of an object's name consists of two parts: the object name type and the name value. The first segment in the example given above is CN=ALINCOLN where CN is the object name type (abbreviated for Common Name) and ALINCOLN is the object name value.

> **The proper order for writing an object's distinguished name is the least-significant (object deepest in the tree) to the most-significant (object closest to the [ROOT]). The distinguished name is always written left to right.**
>
> TIP

NDS will resolve the distinguished names regardless of their character case. For example, ALINCOLN's distinguished name could also be written as follows:

```
cn=alincoln.ou=audit.ou=admin.ou=rio.o=acme
```

as well as:

```
CN=ALincoln.OU=Audit.OU=Admin.OU=Rio.O=Acme
```

Each of the above distinguished names for the user ALINCOLN is equivalent.

Another example of a distinguished name in the ACME tree is for the print queue object known as HP4SI-PQ1. The HP4SI-PQ1 is found in the NDS tree in the container MEDICAL, which is in the container HR, which is in the container SYDNEY, which, in turn, is in container ACME. Figure 4.4 illustrates the distinguished name for this print queue object.

Figure 4.5 shows the distinguished name for two NDS server objects. These server objects are:

```
CN=LABS-SRV1.OU=LABS.OU=NORAD.O=ACME and

CN=WHITE-SRV1.OU=WHITE.OU=CRIME.OU=TOKYO.O=ACME.
```

FIGURE 4.4

The distinguished name for a print queue object in the ACME is: CN=HP4SI-PQI.OU=MEDICAL.OU=HR.OU=SYDNEY.O=ACME
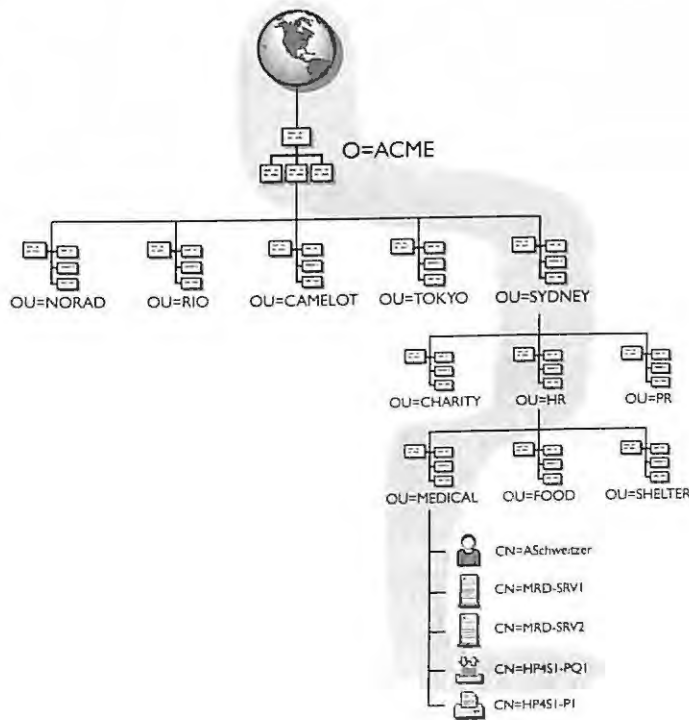
TIP Notice that the object [ROOT] is not used in any of the distinguished names. The [ROOT] object is always implied by the NDS system. Therefore, you are not required to supply a [ROOT] object when entering a distinguished name.

Relative Distinguished Name

A relative distinguished name (RDN) is the individual name assigned to an object otherwise known as the value plus the object naming attribute. The relative distinguished name must be unique only in relation to its parent object. This implies that there cannot be two objects in the same container using the same name.

The distinguished names for the server object CN=LABS-SRV1.OU=LABS. OU=NORAD.O=ACME and the server object CN=WHITE-SRV1.OU=WHITE. OU=CRIME.OU=TOKYO. O=ACME
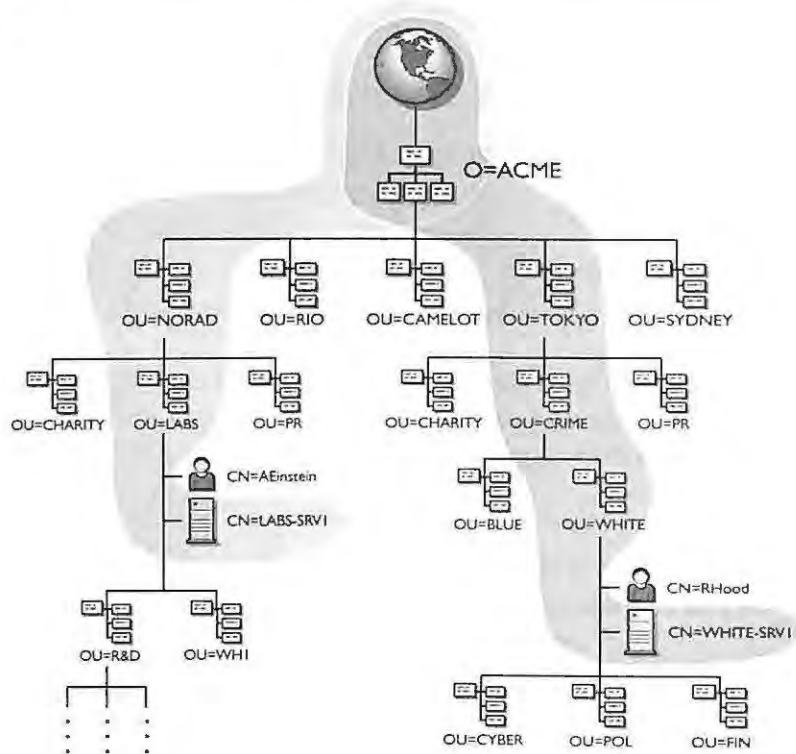
CONSULTING EXPERIENCE

It can be cumbersome to use the distinguished name when referring to the NDS objects. You should become familiar with the use of the relative distinguished name, which is defined as the leaf-most portion of the name. A partial name is another commonly used term and is defined as the name in relation to a container. For example, referring back to Figure 4.5, LABS.-SRV1.LABS is a partial name in relation to the container NORAD.ACME. The server object LABS.-SRV1 is the relative distinguished name in relation to the container LABS.

In Figure 4.6 objects in the R&D department in the NORAD location of the ACME tree have relative distinguished names that are unique in relation to their parent object
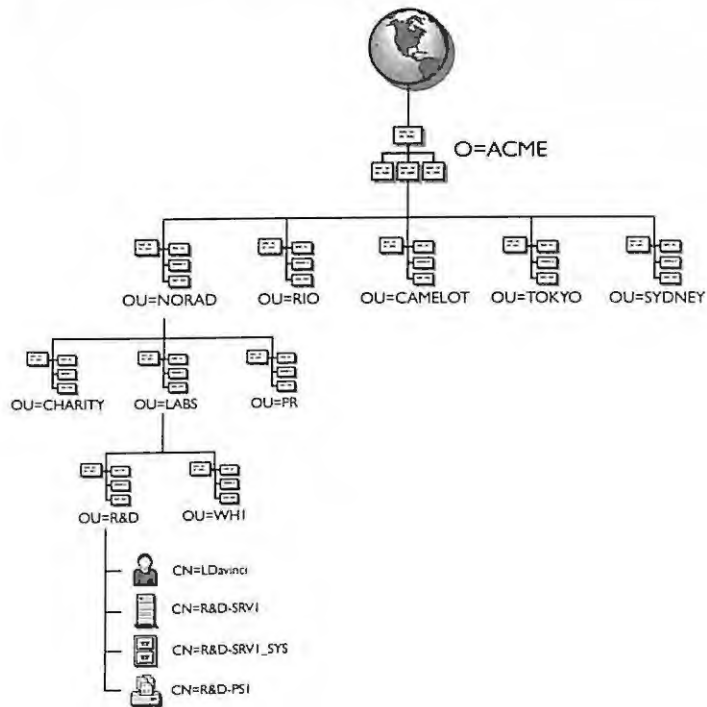
OU=R&D. The objects are CN=LDAVINCI (user), CN=R&D-SRV1 (server), and CN=R&D-SRV1_SYS (volume).

You can see in Figure 4.7 that you will not be able to create a second user object named CN=LDAVINCI in the same container R&D.LABS.NORAD.ACME. NDS naming does not permit duplicate object names in the same container regardless of the object type.

It may not always be obvious that objects of different types still need unique relative distinguished names in the container. For example, a print queue object and a group object cannot have the same relative distinguished name in the same container for the same reasons discussed above.

CONSULTING EXPERIENCE

If you are migrating from NetWare 3, be aware that you will need to have a unique relative distinguished name for each of the objects that you will be moving to NetWare

4.1 and NDS. The NetWare 3 bindery permits you to create objects with the same name as long as the object types are different. This rule does not apply to NetWare 4.1 and NDS.

For example, it is very common for an administrator in NetWare 3 to create a print queue object and the group that will use the print queue with the same name. This is an easy method to administer the printing setup as this form of naming is legal in the NetWare 3 bindery. However, in NetWare 4.1 each object in the container must have a unique relative distinguished name.

During the preparation of migrating from NetWare 3 to NetWare 4.1, you must resolve any of these duplicate names. If you do not resolve the naming conflicts, the NetWare 4.1 migration utility will migrate the first object and not the second object with the same name.

FIGURE 4.7

Two objects within the same container cannot have the same relative distinguished name. There cannot be two objects named CN=LDAVINCI in the container R&D.LABS.NORAD.ACME.
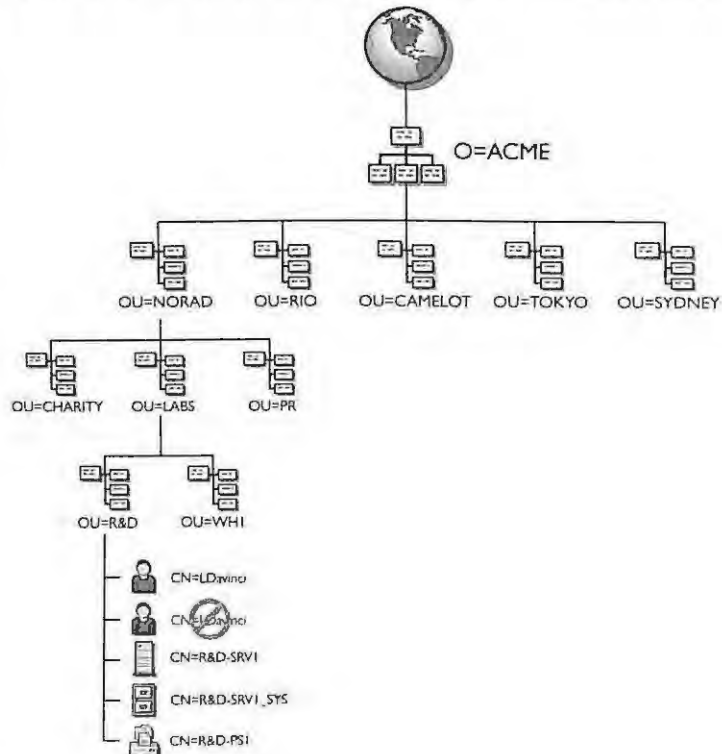
The relative distinguished name has to be unique in only the parent container. The same object name can exist in the tree if the objects are located in different containers because the parent object has a different name, which causes the object to have a unique distinguished name. Thus, the same named object in different containers has a relative distinguished name in relation to its parent.

Figure 4.8 illustrates how two different printer objects in the ACME tree can have the same relative distinguished name but still have a unique distinguished name in the Directory. The two printer objects are:

```
CN=HP4SI-P1 (in container OU=PR.OU=NORAD.O=ACME)

CN=HP4SI-P1 (in container OU=CHARITY.OU=TOKYO.O=ACME)
```

Both the printer objects named HP4SI-P1 are relative distinguished names in relation to their parent container OU=PR.OU=NORAD.O=ACME and OU=CHARITY.OU=TOKYO.O=ACME, respectively.

CONSULTING EXPERIENCE

Although the relative distinguished names for user objects must be unique within one container in the tree, we highly recommend that you create a unique naming standard across the entire NDS tree for users. Having a global naming standard for users reduces name conflicts when users are created and subsequently moved for any reason. Keep in mind that e-mail systems also require unique names.

Relative distinguished names are not just reserved for the leaf objects. All objects in the NDS tree have a relative distinguished name in relation to their parent container. For example, in the ACME tree, there cannot exist two OU=NORAD containers under the O=ACME. However, in Figure 4.9, you will notice two containers called CHARITY and PR that are under each of the locations. This is acceptable by NDS because each instance of the OU=CHARITY and OU=PR containers is unique in relation to its parent containers.

Remember, combining each of the relative distinguished names with its parent object back to [ROOT] creates a distinguished name for the Directory object. Both the relative distinguished name and the distinguished name can be entered as *typeful* or *typeless* names.

FIGURE 4.8

Two separate printer objects in the ACME tree can have the same relative distinguished name only if the printers are in different containers in the tree.
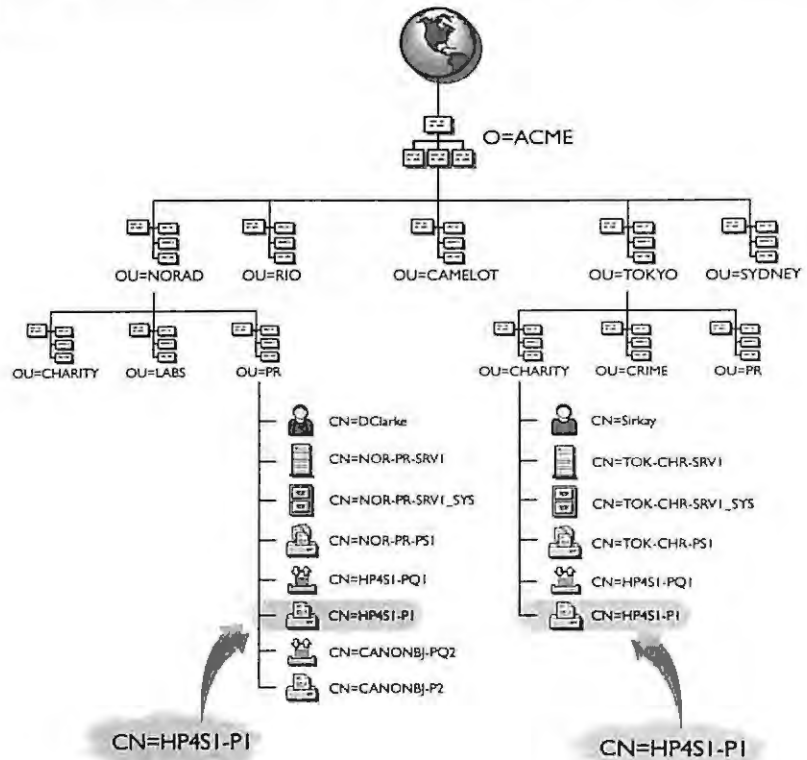
FIGURE 4.9

Each of the OU=CHARITY and OU=PR containers under each of the location OUs have relative distinguished names because they are unique under each of the locations.
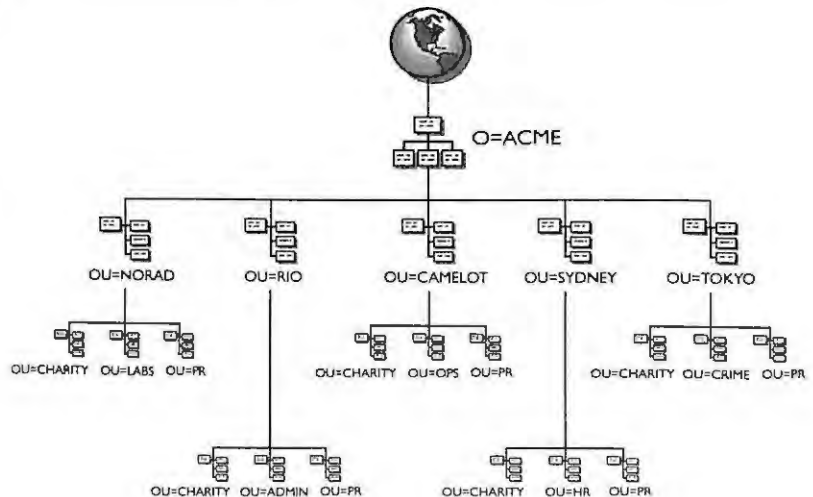
Typeful Names

Using or placing the object name types in an object's distinguished name is referred to as typeful naming. A couple of examples in the ACME tree of a typeful name are:

```
CN=SIRKAY.OU=CHARITY.OU=TOKYO.O=ACME
```

or

```
CN=TOK-CHR-PS1.OU=CHARITY.OU=TOKYO.O=ACME
```

where the user SIRKAY and print server TOK-CHR-PS1 are in the Charity OU in the Tokyo location.

You can also refer to each of these objects independently as CN=SIRKAY, CN=TOK-CHR-PS1, OU=CHARITY, OU=TOKYO, and O=ACME.

Each of these objects has its object name type plus the object name; hence, the definition of typeful naming. Typeful names, such as the CN designator, help define the object's location in the hierarchy. The CN designator always indicates a leaf object and is the lowest object in the hierarchy.

Typeful names are not required by the Novell utilities or NDS to indicate an object's complete name because you can also use typeless names to write the complete name without using the object types. The Novell client software allows you to use typeless names because it parses the entered name with the appropriate name type.

The typeful naming method is not case sensitive. Using the examples given above, you can see that the following names are equivalent:

```
cn=sirkay.ou=charity.ou=tokyo.o=acme
```

is equivalent to

```
CN=SIRKAY.OU=CHARITY.OU=TOKYO.O=ACME
```

and

```
cn=tok-chr-ps1.ou=charity.ou=tokyo.o=acme
```

is equivalent to

```
CN=TOK-CHR-PS1.OU=CHARITY.OU=TOKYO.O=ACME
```

CONSULTING EXPERIENCE

Using typeful names can obviously be a lengthy method for indicating an object's distinguished name. Thus, this form of naming is not required or recommended for use with the command line utilities, especially LOGIN.EXE. It is cumbersome for the user to learn and use typeful naming. The user should only need to use typeless naming for the NDS objects.

Typeless Names

Typeless names are object names that do not include the object name type in each segment. Using the same examples that we used in the typeful naming from the ACME tree, the typeless names would be:

```
SIRKAY.CHARITY.TOKYO.ACME
```

or

```
TOK-CHR-PS1.CHARITY.TOKYO.ACME
```

where the user SIRKAY and print server TOK-CHR-PS1 are in the Charity OU in the Tokyo location.

Notice that we have removed the object name type from each of the segments in the object. You can now refer to the objects as simply SIRKAY (where the CN= has been removed), TOK-CHR-PS1, CHARITY, TOKYO, and ACME. All the object name types have been removed.

Like the typeful naming method, typeless naming is not case sensitive. Again, using the examples given above, you can see that the following names are equivalent:

```
sirkay.charity.tokyo.acme
```

is equivalent to

```
SIRKAY.CHARITY.TOKYO.ACME
```

and

```
tok-chr-ps1.charity.tokyo.acme
```

is resolved in the same way as

```
TOK-CHR-PS1.CHARITY.TOKYO.ACME
```

> **TIP**
>
> **Typeless names are easier to use for both administrators and users because they are shorter and more intuitive. You will use the typeless naming more during your day-to-day operations. You should become very familiar with it.**
>
> **You can also use this method of typeless naming for indicating either a relative distinguished name or a distinguished name.**

Context

An object is identified in the Directory by its distinguished name, which provides the dual benefit of uniqueness and location within the tree. An object's context is its position or location in the NDS tree. If two objects are in the same container, they have the same context because their parent containers will be identical. In NDS tree terms, the context can be defined as the name of the parent object.

For simplicity, NetWare 4.1 allows the client to shorten or abbreviate a complete object name through the use of a user context.

Figure 4.10 shows CDARWIN's context or location in the ACME tree. CDARWIN is in part of the Pollution department or organizational unit (OU=POLL), in the R&D department (OU=R&D), in the LABS division (OU=LABS), in the NORAD office (OU=NORAD), under the ACME tree (O=ACME). The context for CDARWIN is OU=POLL.OU=R&D.OU=LABS.OU=NORAD.O=ACME.
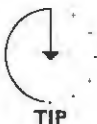
> **TIP**
>
> **The best analogy for the context in NDS is the "path" in the DOS file system. The path statement in DOS helps you quickly and automatically find files stored in specific subdirectories. In the same fashion the context in NDS will help you find resources in specific NDS containers.**

FIGURE 4.10

*The context in the ACME
tree for the user
CN=CDARWIN is
OU=POLL.OU=R&D.
OU=LABS.OU=NORAD.
O=ACME.*

Context is: OU=POLL.OU=R&D.OU=LABS.OU=NORAD.O=ACME
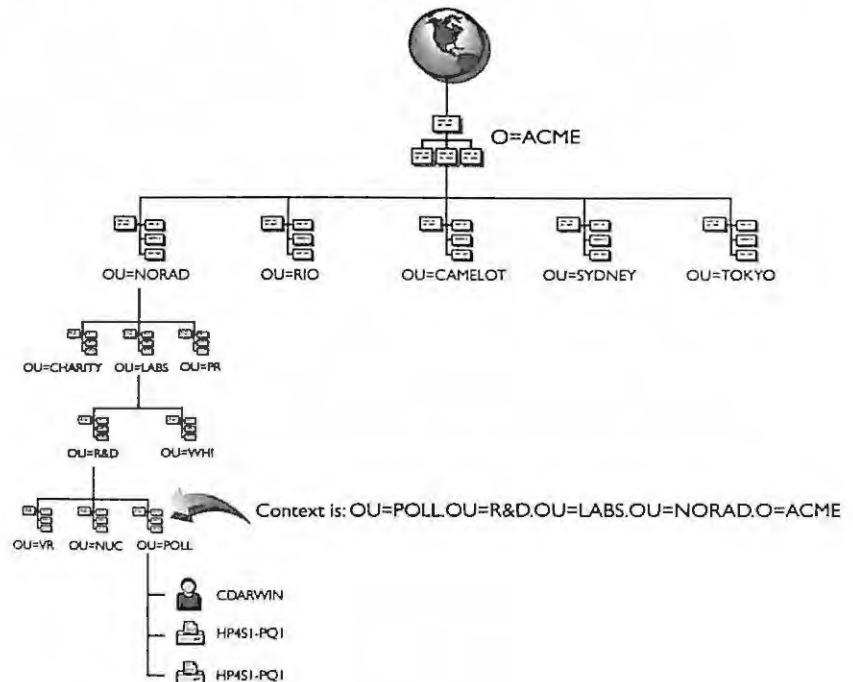
Setting a context helps the user and administrator access the objects in the tree more easily. If a context is set then the user and administrator simply have to enter the relative distinguished name for the resource in the container where the context has been set. Figure 4.11 shows several contexts that exist on the ACME tree:

```
OU=OPS.OU=CAMELOT.O=ACME

OU=FIN.OU=OPS.OU=CAMELOT.O=ACME

OU=DIST.OU=OPS.OU=CAMELOT.O=ACME
```

A context can be set only to a container object. This means that you can set a context to point to either an Organizational Unit (OU=), an Organization (O=), or the [ROOT] object. You cannot set a context to a leaf object.

FIGURE 4.11

Several contexts at different levels are shown in the ACME tree.
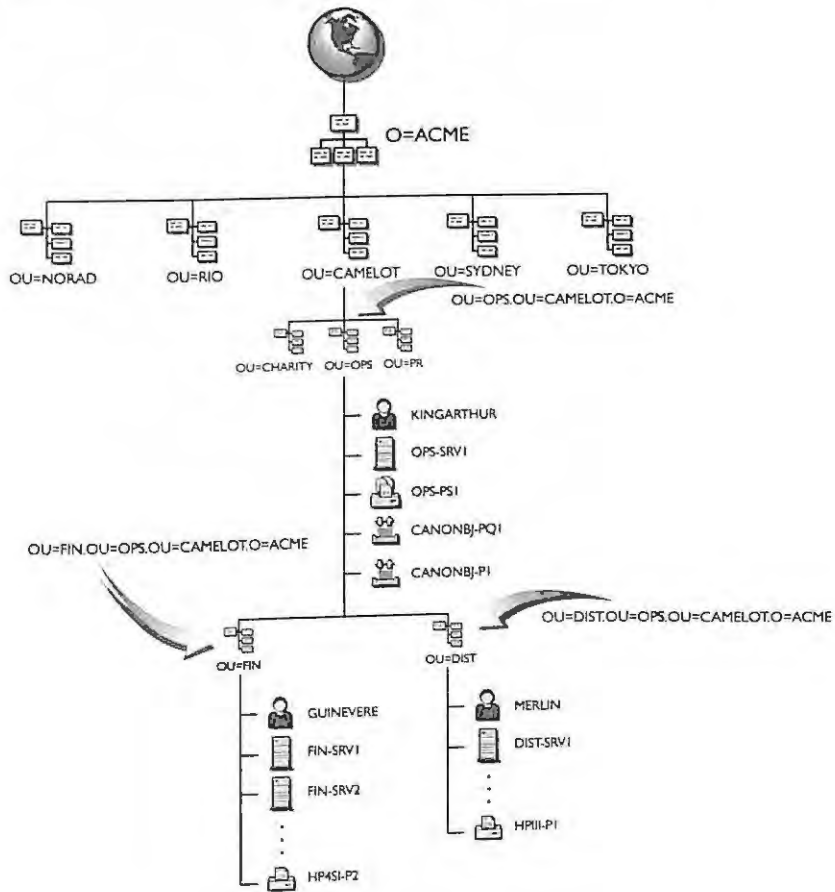
A user can more easily identify an object in the tree if his or her context is set to the appropriate container in the tree. For example, if the user MERLIN wants to search for all printers in the OU=DIST in the OPS division in CAMELOT, he can set his context to OU=DIST.OU=OPS.OU=CAMELOT.O=ACME and search only there for the desired printer using the NWUSER utility.

Using the information in Figure 4.11, in order for the user KINGARTHUR to log in using just the object name

```
LOGIN KINGARTHUR
```

he should set the name context variable in the workstation file
NET.CFG to

```
Name Context = "OU=OPS.OU=CAMELOT.O=ACME"
```

The user **KINGARTHUR** could also use the Novell **CX** (Change
ConteXt) utility to manually select the same context. Keep in mind
that the **CX** command is valid only during the current session. By
selecting the correct context he can simply enter his user name at
the **LOGIN.EXE** command line and **NDS** will look in the **OU=OPS**
container for his name.

The alternative is not having a context set. This means that the user
would have to know his distinguished name and be able to enter it in
order to log in. For example, if the user **KINGARTHUR** did not set a
context, he would have to enter the following during login:

```
LOGIN CN=KINGARTHUR.OU=OPS.OU=CAMELOT.O=ACME
```

or

```
LOGIN KINGARTHUR.OPS.CAMELOT.ACME
```

The name context in NDS gives you the functionality of navigating the NDS naming
service. The bindery context gives you the functionality to browse the same information
as a bindery naming service, which is an entirely different context.

BINDERY CONTEXT

NDS provides compatibility with NetWare 2 and NetWare 3 using a feature called
bindery services. This feature allows bindery versions of NetWare and other bindery-
based applications to access the NDS tree as if it were the bindery. Bindery services is
accomplished by setting a bindery context on the server, which specifies the name of the
NDS tree container(s) to search as the bindery.

In order to set the bindery context on a server, you can enter

```
SET BINDERY CONTEXT = OPS.CAMELOT.ACME
```

at the server console. The server bindery context can also be set
using the server-based **SERVMAN** utility.

Prior to NetWare 4.1, object information was not available from a distributed directory. A NetWare 3 server stored only information related to its own server in the bindery. Therefore, applications were written that accessed each NetWare 3 server's bindery for a user name. NDS provides bindery services that allow objects in a container to be accessed by bindery-based clients as well as by NDS objects. When a bindery-based application makes bindery calls to the server, the server sees the objects in the container where the server bindery context is set.

NOTE

In terms of bindery context on the server, NDS will see only the objects that previously existed in a NetWare 3 bindery as the bindery objects. These objects are limited to: user, group, print queue, print server. In addition, NetWare 4.1 has added the profile object to assist with migrations from NetWare Name Services (NNS).

The default server bindery context is set to the container where the server was installed. You can change the context to another container or OU if you want. Previous versions of NetWare 4, up to and including NetWare 4.02, allowed you to set only a single context for your server. NetWare 4.1 allows you to set up to 16 contexts for a single server.

You can set multiple contexts by using the SERVMAN utility as shown in Figure 4.12. This utility allows you to separate each context by using a semicolon or a space as shown in this example.

FIGURE 4.12

Use the SERVMAN utility to set multiple bindery contexts.

LEADING PERIODS

The period (.) separates the individual name segments in distinguished names, relative distinguished names, and user contexts. Through the use of leading and trailing periods in the name sequence, you can quickly navigate to any object in the tree. This is useful for referencing objects located in other portions of the tree.

A name with a leading period instructs the client software to ignore the current context and start the search at the [ROOT] object. This means that any object name with a leading period will be treated as a distinguished name starting at the [ROOT] object.

For example, using the information in Figure 4.13, the current context for the user is set to:

```
OU=R&D.OU=LABS.OU=NORAD.O=ACME
```

Using the Novell CX utility you type the following command (notice the leading period at the beginning of the distinguished name):

```
CX .DIST.OPS.CAMELOT.ACME
```

which will be resolved to the container OU=DIST.OU=OPS.OU=CAMELOT.O=ACME by the NDS client. The leading period causes the name to be treated as a whole name, not a partial name. In other words, start at the [ROOT] and work down.

Using the leading period is simply a shortcut for navigating the tree from the command line utility of your workstation. There can be only one leading period in the object name on the command line.

Trailing Periods

A period after an object name (known as a trailing period) can be used to select a new context. The use of a single trailing period will move the context up by one container in the tree. For instance, you can use the Novell CX (Change ConteXt) utility and type the following command at the DOS prompt (notice the single trailing period):

```
CX.
```

This command will automatically move you up to the previous layer or container in the tree. Unlike leading periods, you can have as many trailing periods up to the number of relative distinguished names (RDNs) in the current context. If you want to move the current context up two layers in the tree, then you simply supply two trailing periods. You change the context in the tree one layer for every trailing period you supply at the end of the name.

FIGURE 4.13

A view of the ACME tree and
changing the context
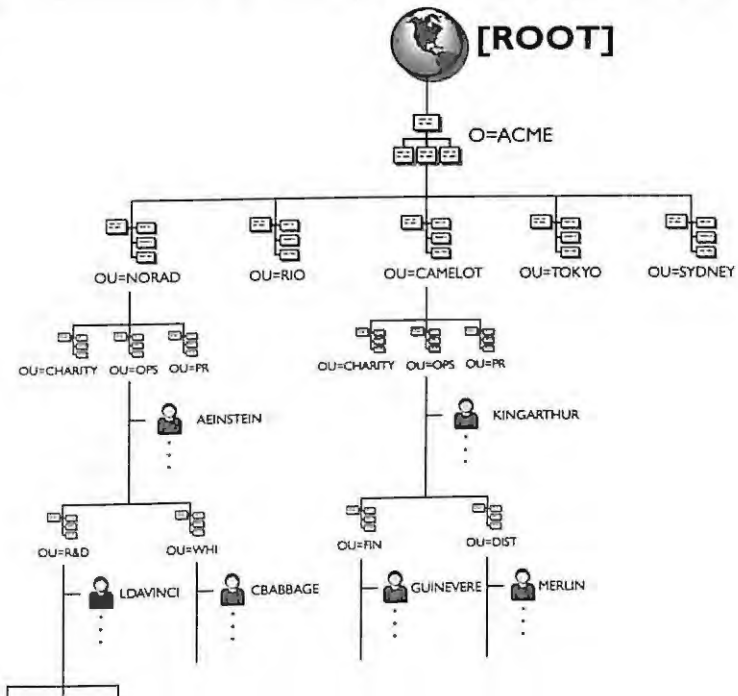
In Figure 4.14, let's assume that the current context is

```
OU=DIST.OU=OPS.OU=CAMELOT.O=ACME
```

and you want to change the context to O=ACME. You simply type the following command
(notice the three trailing periods):

```
CX...
```

Trailing periods can be used with relative distinguished names to instruct segments
of the context to be replaced. The replacement occurs during the resolution of the full
distinguished name.

In Figure 4.14, for example, let's assume that the current context is:

```
OU=DIST.OU=OPS.OU=CAMELOT.O=ACME
```

If we wanted to change our context to the OU=FIN container, which is under OU=OPS,
then we would simply type the following command (notice the trailing period):

```
CX OU=FIN.
```
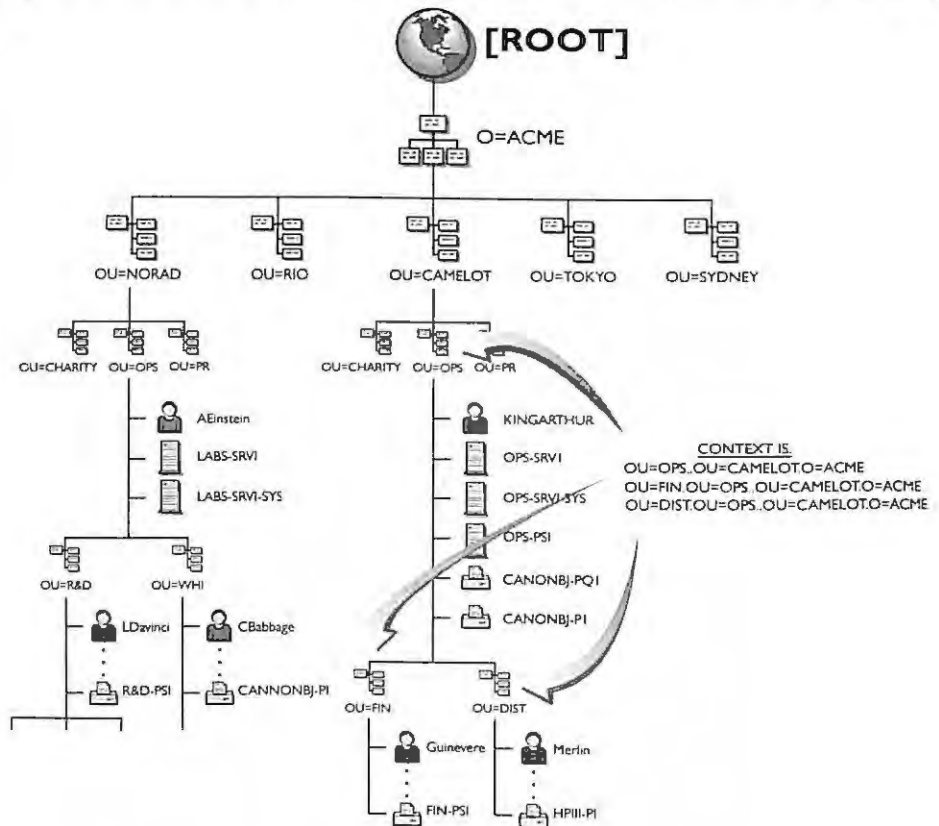
For each trailing period in the RDN, one segment of the context is ignored. This starts at the least-significant segment (deepest in the tree) and proceeds to the most-significant segment or until the trailing periods are exhausted.

Again in Figure 4.14, let's assume the current context is:

```
OU=DIST.OU=OPS.OU=CAMELOT.O=ACME
```

If we wanted to change our context to the OU=WHI container, which is under OU=LABS in the NORAD location, then we would need to type the following command:

```
CX OU=WHI.OU=LABS.OU=NORAD...
```

Naming Standards Document

Creating a naming standards document need not be difficult or time consuming. The first step is to determine why such a standard should be created. The preceding sections of this chapter have demonstrated that naming is the structure of the Directory tree. As a basis or purpose for why the standard is important, consider the following:

► How can you provide a Directory structure for your employees that is flexible, easy to use, and meets today's business needs?

► How can you ensure that you provide a consistent Directory structure within all divisions, locations, and perhaps, operating companies?

► How can you communicate and implement a consistent standard across an entire company that everyone will follow?

The answers to these questions lie in creating a naming guidelines document that will force you to devise clear and easy-to-remember names and thus make the NDS database a more useful resource.

Naming Guidelines

Create some naming guidelines before installing your first NetWare 4.1 server. Once an installation has begun it becomes more difficult to implement a standard. At a minimum, you should determine how Organizational Units, Servers, and User objects will be defined. These will be the most widely created objects along with your printing objects.

If you have already implemented some existing naming, that's the place to start your review. Many companies already have some guidelines in place for defining their users' names. These guidelines are usually based on an e-mail standard. Review the standard and make modifications where necessary to add or modify your current standard. If your company's network is small, you should still create naming guidelines because the network will most likely grow to include more users and resources in which naming

becomes more important. Supporting a network with consistent naming is made easier as well.

Your naming guidelines need to be global in nature. You should strive to implement your naming guidelines consistently across the entire network so that all users and departments can readily and easily access resources anywhere on the LAN. Global naming guidelines can also help your administrators to understand the purpose and use of all objects in the tree.

Most likely you will need the input from others in your company in order to create some naming guidelines. You should gather information from the organizations within your company that must participate to form the naming standard. This information may include input from e-mail administrators, network administrators, and mainframe personnel.

Usually an established host environment will have user naming standards in place. Review these standards as a basis for your NetWare 4.1 user naming guidelines. Some customers prefer to maintain their host userids on the LAN rather than create a new standard at the network level. Others can't wait to change their host userids!

Consider all other naming guidelines that you may already have in place from a NetWare 3 environment to determine how well they might fit into the NetWare 4.1 environment. Your NetWare 3 printers and servers may already have names defined that will work just fine after migration to a NetWare 4.1 tree. If your organization has sufficient or acceptable guidelines, then there is no need to change them.

As mentioned previously many sites already have e-mail addresses set up that require a user naming convention, such as the first initial and last name of the user (for example, TJEFFERSON for Thomas Jefferson).

Naming guidelines provide consistency across your network. If your naming standard is already consistent and firmly in place, you are in great shape and can move on to the next phase of designing your NDS tree. If you do not have an existing naming standard then you should consider establishing naming guidelines that meet the goals discussed in the following section.

GOALS FOR THE NAMING GUIDELINES

The goals or objectives for creating naming guidelines for NetWare Directory Services should be as follows:

- To make browsing and navigation of the NDS tree easier for the users

- To make maintenance of the NDS tree easier for the administration staff

- To make merging separate NDS trees easier

- To keep the NDS object names unique as required by certain services

- To avoid special characters reserved by the operating systems

NAMING GUIDELINES HELP NDS BROWSING AND NAVIGATION

Your primary objective for creating the naming guidelines is to provide the network user a simple view of the network resources contained in the tree. You may not see the immediate need for good naming guidelines, which enable the user to easily navigate the tree. But as more applications are written to take advantage of NDS and its searching capabilities, an uncomplicated naming standard will be needed.

Objects named with purpose and consistency will eventually provide you with a solid foundation that benefits your administrators as well as your users. Other benefits will be recognized as applications make more use of the Directory and its services. You will see that the system is much more efficient if users can quickly identify the network resources available to them and minimize the impact on searching NDS.

You should keep the names of all the objects short and simple, yet descriptive enough that the user knows by the name what the object is and the services it provides. For example, the object name HP4SI-P1 is short yet descriptive and lets the user know that this object is an HP 4si laser printer. The object name HP4SI-PQ1 is another example of a short name but is different from the previous example. The PQ1 suffix lets the user know that the object is the print queue that supports the printer (P1).

As you can see, there are many instances throughout the book where we have benefited by using the names of containers and network resources from the ACME tree. It has been helpful to have clear and simple names. Some examples of the naming guidelines for ACME are shown in Table 4.2.

T A B L E   4.2

Examples of the naming
guidelines for ACME

OBJECT	ENTITY	STANDARD
[ROOT]	Tree Name	ACME_TREE
O=Organization	A Cure for Mother Earth	ACME
OU=Organizational Unit	Rio de Janeiro, Brazil	RIO
	Tokyo, Japan	TOKYO
	Sydney, Australia	SYDNEY
	Research and Development	R&D
	World Health Index	WHI
	Facilities	FAC
	Public Relations	PR
Users	George Washington	GWASHINGTON
	Thomas Jefferson	TJEFFERSON
	Leonardo DaVinci	LDAVINCI
	Albert Einstein	AEINSTEIN
Servers	Server in Labs	LABS-SRV1
	Server in White	WHITE-SRV1
	Server in Norad/PR	NOR-PR-SRV1
Printers	HP 4si in Norad/PR	HP4SI-P1
Print Queues	Queue in Norad/PR	HP4SI-PQ1

NAMING GUIDELINES HELP MAINTAIN NETWORKS AND NDS

Consistent naming provides a framework for the network administrators to monitor
and maintain the network and NDS. The administrators will be installing file servers,
creating users and printers, modifying existing objects, and moving objects within the
tree. The administrators will also set up all the user configuration files at the workstation
to connect to NDS in the predetermined fashion. The administrator's job is going to be
much easier if there are naming guidelines in place.

NAMING STANDARDS HELP MERGE NDS TREES

The capability to merge multiple NDS trees is a feature of NetWare 4.1. However, two trees being merged within the company will be made much easier if the trees are based on the same naming standards for container and leaf objects. The tree merge will be seamless to users because their workstation configuration files will not have to be modified, even though the NDS trees are merged.

NAMING GUIDELINES HELP KEEP NDS OBJECT NAMES UNIQUE

Some of the NDS tree objects are required to have unique object names in order for the network to work properly. For instance, the file server and print server objects stored in NDS broadcast their services using the Service Advertising Protocol (SAP). While the file server and print server objects are not required to have unique names on the network, it is highly recommended in order to avoid confusion.

Establishing and dictating the naming guidelines to all the network administrators in the network will help keep the NDS objects unique. As stated previously, it is recommended that the user objects have unique names throughout the network. This way users can be moved or new users added without conflict.

Not all the objects in the NDS tree will need unique names throughout the network. You decide which NDS objects have unique names. The SAP requires that file server object names be unique on the network.

NOTE

All network devices on the network that communicate using the Service Advertising Protocol should have unique object names in the entire network. Therefore, all file servers and print servers should have unique names on the wire. The maximum length of the names of the objects that use SAP is 47 characters. The space character is illegal for server names.

NAMING GUIDELINES SHOULD AVOID SPECIAL OR RESERVED CHARACTERS

NDS requires the use of the escape character with some characters and therefore we recommend you avoid using them in your naming guidelines. The characters are:

▸ Period (.) — The period is used by NDS to separate the name segments of distinguished names. For example, the distinguished name for the user George Washington is:

```
CN=GWASHINGTON.OU=ADMIN.OU=RIO.O=ACME
```

▸ Commas (,) — The comma is allowed by NDS naming rules; however, it is very confusing and generally should be avoided.

▸ Plus (+) — The plus sign is used by NDS to represent objects with multiple naming attributes such as the bindery objects. A common use of the plus sign is for bindery objects placed in the NDS database. For example, a bindery object with the plus sign in the name is seen as CN=BinderyObject+Name. In the case of an Appletalk print server it might appear as:

```
CN=AtpsQuser+83
```

▸ Equals (=) — The equals sign is used by NDS to tie name types and object names together. For example, the object type O=Organization would appear as:

```
O=ACME
```

▸ Backslash (\) — The backslash precedes the special characters above if they are used as part of an object name. For example, you want to use the name ACME Inc., which has a period at the end. In order to make it a legal NDS name, you would need to enter:

```
ACME Inc\.
```

It is also recommended that you understand the following guidelines if you use spaces in any of the object names. You may want to use the underscore character instead. Spaces are used as delimiters for all the command line parameters. If you use spaces in the NDS object names and need to use the name with a command line utility, then you will need to enclose the name in quotes.

For example, the user George Washington is named "George Washington" in NDS with a space separating the first and last name. In order to use the relative distinguished name with the LOGIN.EXE program, the name would have to be entered with quotes around it:

```
LOGIN "George Washington"
```

and not

```
LOGIN George Washington
```

If you were required to provide the full distinguished name during login, the name would then appear as:

```
LOGIN "GEORGE WASHINGTON.ADMIN.RIO.ACME"
```

You can also use the underscore character as a replacement for the space character because NDS interprets the space and underscore in the same way. In the example of the user George Washington, his name would appear as:

```
George_Washington
```

When the name is used during login, it would be entered without the quotes:

```
LOGIN George_Washington
```

You should also avoid the use of the forward slash (/) in your naming of any container or leaf object. Novell's Windows-based NWUSER utility will fail if you map permanent drives or capture a print queue. This is an MS Windows issue that does not allow for the use of the forward slash in any name.

Producing a Naming Standards Document

Producing a naming standards document is not very glamorous, but it shouldn't require a great deal of time to complete either. Here are some suggested steps you can take to produce your naming standards document as quickly and efficiently as possible:

▸ Document the naming standard for each object used in the NDS tree.

▸ Provide an example for each object used.

▸ Specify properties for each of the objects selected.

DETERMINE THE NAMING STANDARD FOR EACH OBJECT USED IN THE TREE

In order to create the naming guidelines, review which NDS objects you will use in your NDS tree. Typically, these objects include organizational units, servers, users, printers, print queues, groups, organizational roles, directory maps, and profiles.

Other possible areas for naming consistency include the attributes associated with each object you define in the NDS tree. In short, any information you plan on using to search the NDS database needs a naming standard for consistency.

The naming standard for each of the objects used in the ACME tree is included in Table 4.3.

TABLE 4.3

NDS object naming standard for the ACME tree

NDS OBJECTS	STANDARD
Users	First character of the first name plus the entire last name. All titles are spelled out.
Organization	Abbreviation of the company name.
Organizational Unit	Location, division, or department name. Abbreviate the names if they are over eight characters long.
NetWare Server	Department-SRV#. Exception is the Location-Department-SRV#, which is for the CHARITY and PR departments. The difference is because of the duplication of container names for all locations of CHARITY and PR.
Volumes	ServerName_VolumeName.
Print Server	Department-PS#. Exception is the Location-Department-PS#, which is for the CHARITY and PR departments.
Printer	PrinterType-P#.
Print Queue	PrinterType-PQ#.
Computer	No standard because it is not used.

PROVIDE AN EXAMPLE FOR EACH OBJECT USED

You may have noticed when reading Table 4.3 that there were no examples of the object naming standards. Remember, a picture is worth a thousand words. An example can convey the meaning of an entire written page. Therefore, include some brief examples in your naming guidelines document. As shown below in Table 4.4, you can convey your meaning more quickly and easily by providing an example.

Table 4.4 is simply a reprint of Table 4.3, but examples for each of the object's naming standard have been added for clarity.

TABLE 4.4	NDS OBJECTS	STANDARD
NDS object naming standard for the ACME tree	Users	First character of the first name plus the entire last name. All titles are spelled out.
		Examples: GWASHINGTON, SIRGAWAIN
	Organization	Abbreviation of the company name, which is "A Cure for Mother Earth."
		Example: ACME
	Organizational Unit	Location, division, or department name. Abbreviate the names if they are over eight characters long.
		Example: NORAD, R&D, ADMIN
	NetWare Server	Department-SRV#. Exception is Location-Department-SRV#, which is for the CHARITY and PR departments.
		Examples: LABS-SRV1, NOR-CHR-SRV1
	Volumes	ServerName_VolumeName.
		Examples: LABS-SRV1_SYS, NOR-CHR-SRV1_SYS
	Print Server	Department-PS#. Exception for the CHARITY and PR departments, which is Location-Department-PS#.
		Examples: FAC-PS1, TOK-PR-PS1
	Printer	PrinterType-P#.
		Examples: HP4SI-P1, CANONBJ-P2

(continued)

NDS OBJECTS	STANDARD
Print Queue	PrinterType-PQ#.
	Examples: HP4SI-PQ1, CANONBJ-PQ2
Computer	No standard because it is not used.

SPECIFY PROPERTIES FOR EACH OBJECT CLASS SELECTED

For some installations it may be necessary to determine which properties will be required for the selected objects. Some properties are mandated by NDS when you create the object. This means that if the mandatory properties are not filled in then NDS will not create the object. For example, during the creation of a user object, NDS requires you to specify the Login Name and Last Name properties.

You may have additional attributes that you would like all your administrators to include when they create a new object. This type of information might include addresses, fax numbers, department names, and so on. Table 4.5 illustrates the naming standards for a user object with additional properties you want filled in when the users are created. Not all the properties for the user are represented in the table. Some properties are required and a value must be entered at creation, while other properties are optional and are left up to you to determine if they should be used. System properties are those properties that NDS automatically populates during the creation of the object.

PROPERTY	REQ/OPT/SYSTEM	STANDARDS
Login Name	Required	First character of the first name plus the entire last name. Add a middle initial to resolve name conflicts.
Given Name	Optional	First name of the user.
Last Name	Required	Last name of the user.
Full Name	Optional	First and last name of the user.
Generational Qual.	Optional	
Middle Initial	Optional	Middle initial of the user, if known.
Other Name	Optional	

TABLE 4.5

NDS property naming standards for a user object in the ACME tree
(continued)

PROPERTY	REQ/OPT/SYSTEM	STANDARDS
Title	Optional	Job title.
Description	Optional	
Location	Optional	City or site location (NORAD, RIO).
Department	Optional	
Telephone	Optional	Business phone number with area code.
Fax Number	Optional	Fax phone number with area code.
Language	Optional	Preferred language of the user.
Network Address	System	
Default Server	Optional	Enter the same server as the home directory.
Home Directory	Optional	Enter the volume/ subdirectory/user path.
Require Password	Optional	Force the user to have a password.
Account Balance	Optional	
Login Script	Optional	Determined by the site administrators.
Print Job Config.	Optional	Determined by site administrators.
Post Office Box	Optional	
Street	Optional	
City	Optional	
State or Province	Optional	
Zip Code	Optional	
See Also	Optional	

CONSULTING EXPERIENCE

Currently the Novell utilities cannot enforce additional required properties that you define as an administrator. However, we recommend a well-documented guideline in which you explain your reason for including a particular attribute. Your administrators will be more likely to adhere to the guideline if they understand the reasoning behind the inclusion of an object's attribute.

MORE IDEAS FOR NAMING STANDARDS

User Accounts

When you create a naming standard, one of the first steps is to decide how to standardize your usernames. NetWare 4.1 allows usernames to be up to 64 characters long. However, a 64-character username is a very long name and not easy to use.

In the example of ACME, the username was limited to the first character of the first name plus the entire last name. If there are duplicate names within a single container, then the middle initial is added to resolve the name conflict. This type of username may not be used globally because of the number of username conflicts.

Some companies have limited the length of the username to eight characters. This naming convention for the user accounts matches the naming of the user's DOS home directory. Since DOS is limited to just eight characters for subdirectory names, limiting the usernames to eight characters will automatically match the name of the DOS subdirectory assigned to the user. For example, if you use eight characters for the user, Thomas Jefferson, his name could be defined as:

```
TJEFFERS
```

which is the first character of his first name and the first seven characters of his last name.

Another way to represent a user is to take the first six characters of his last name and the first two characters of his first name. This name would appear as:

```
JEFFERTH
```

Duplicate names can be handled by specifying a middle initial in one of the two names such as:

TQJEFFER

(Thomas Q. Jefferson)

or

QJEFFETH

(Q. Jefferson Thomas)

> **Remember, object names within the same container must be unique. Duplicate names are allowed only if the objects are in different containers. However, it is still recommended that you keep object names unique across the entire tree.**
>
> TIP

Organization and Organizational Unit Names

Your organization name should reflect your company name. The company name "A Cure for Mother Earth" is abbreviated to ACME. Thus, the O=Organization is called O=ACME. For the Organizational Units, the locations and departments with short but descriptive names are used. For instance, OU=SYDNEY and OU=TOKYO are examples of the location OUs. They could have been abbreviated to SYD and TOK but their names are already short. Examples of the division and department Organizational Units are OU=R&D for the Research and Development department and OU=MKRT for the Marketing department.

> **The tree name can be named with the company name plus _TREE. We have chosen as our tree name for the ACME company ACME_TREE and O=ACME for the organization. Notice the difference between tree name and the organization name.**
>
> NOTE

Server Names

Server names must be unique on the entire network because each server name is broadcast through the Service Advertising Protocol. You may wish to consider a server name that signifies a location and department.

For ACME, a NetWare server in the CHARITY department of NORAD is called NOR-CHR-SRV1, and a print server is called NOR-CHR-PS1. A server in the R&D department in NORAD is called R&D-SRV1. You may wonder why the naming standards seem

different for different departments. Notice that the R&D department is a unique department within the ACME tree and that there is no need for the location name in the server name. However, the CHARITY and PR departments are not unique departments within the ACME tree. These two departments are placed as OUs in every location, and the servers in those containers should have the location name in order to be unique.

Printer and Print Queue Names

Printers and print queue names do not have to be unique on the network. The department and location information is obtained by where the printer and print queue are placed in the tree. The distinguished name of the printer or print queue object is used to discover the location and department in which the printer is placed. Thus, the individual object name for the printer and print queue should show the functionality of the printer.

For example, in the ACME tree an HP 4si printer is named HP4SI-P1, and the print queue that supports the printer is called HP4SI-PQ1. A Canon Bubble Jet printer is named CANONBJ-P2, and the print queue is named CANONBJ-PQ2. This type of naming provides users with fundamental information about the printer they will be using.

The user can obtain the location and department in which the printer has been placed from the distinguished name or location in the ACME tree. For example, a Canon Bubble Jet printer (CANONBJ-P1) in the Operations (OPS) department in Camelot (CAMELOT) has a distinguished name of CANONBJ-P1.OPS.CAMELOT.ACME.

The difference between the printer object and the print queue object is defined by the object suffix. For instance, the "P" is for the printer object and the "PQ" is for print queue object.

NOTE

A printer that connects directly to the network and services a print queue on the file server is called a queue server. A queue server is a special-purpose print server that needs to have a unique name to broadcast using SAP. All queue servers use SAP to advertise their services on the network.

Sample of a Basic Naming Standard

The following table is another example of how you can create a naming standard document that is simple but very useful. Table 4.6 shows the document with the object type, the naming syntax, and an example. Keep in mind that, although we have already demonstrated our naming standard in previous sections of this chapter, the following naming standard is another one for your review.

TABLE 4.6

Sample of a Basic Naming Standard

OBJECT TYPE	SYNTAX	EXAMPLE
[ROOT] or Tree Name	AAAA_TREE	ACME_TREE
Organization	AAAA = Company Name	O=ACME
Organizational Units	XXXXXX, YYYYYY	OU=RIO, OU=FIN
	XXXXX = Location	
	YYYYY = Department	
NetWare Servers	XXXYYZ##	RIO-ADM-F0I
Print Servers	XXXYYZ##	NOR-CHR-P04
Volume Names	Server_Volume	RIO-ADM-F0I_SYS
	XXX = Location (NOR, RIO, CAM, TOK, SYD)	
	YYY = Department (LAB, CHR, PR, ADM, OPS, FIN)	
	Z = Server type where:	
	F=File Server, T=Test Server, P=Print Server, C=Comm Server, etc.	
	## = Quantity (01, 02, 03, . . . , 99)	
	Volumes are: SYS, APPS, DATA, USERS, SHARE	
Printers	YYYTTLL##P	FIN-LJPS0I-P
Queues	YYYTTLL##Q	FIN-LJPS0I-Q
Queue Server	YYYTTLL##QS	FIN-LJPS0I-QS

Sample of a Basic Naming
Standard
(continued)

OBJECT TYPE	SYNTAX	EXAMPLE
	YY = Department (LAB, CHR, PR, ADM, OPS, FIN, HR)	
	TT = Type of printer (LJ for LaserJet, BJ for BubbleJet, PL for plotter)	
	LL = Language (PS for PostScript, PC for PCL, etc.)	
	## = Number of printer, queue, or queue server (01, 02, 03, ... , and so on)	
	P = Printer	
	Q = Queue	
	QS = Queue Server	
Group HelpDesk	Function Function (the activity that the group will perform)	WP_Group
Organizational Roles	Function_OR Function (the administrative activity or role that the Organizational Role performs)	RioAdmin_OR
Profiles	Name_P Name (the purpose of the Profile object)	Mobile_P
Directory Maps	Directory Name Directory Name (the DOS directory name in which the application has been installed)	WP61

Network Addressing Standards

SERVER INTERNAL NETWORK ADDRESS

The IPX cable address defines the physical cable segment for your file servers. The cable segment may be your entire network or you may have many segments based on locations in your network infrastructure. If you do not already have a standard for cable segments, you should implement one. A brief example of a cable segment standard is:

XXXSSSSN

For example: 20025451 represents the Norad (200) facility building 2545, segment number 1, where:

▶ XXX = Numeric value that represents a location such as NOR, CAM, TOK

▶ SSSS = Numeric identifier for a wiring closet or building

▶ N = Numeric value that represents a number for the specific segment. This value is useful if you have more than one segment in that location.

IPX NETWORK ADDRESS

Most sites using NetWare 3 will have an IPX network address in place for the existing NetWare 3 servers. If you have an IPX network address, you can continue to use the same standard in NetWare 4.1. The important point is to maintain unique network addresses for all your NetWare servers because duplicate addresses will cause great difficulties for your network. The IPX network address must be unique for the cable segment address as well.

A sample network address may appear as the following convention:

ARRSSSSN

where:

▶ A = Indicates that this is an internal IPX number

- ► RR = Indicates a numeric value designating a state or other region

- ► SSSS = Indicates a location

- ► N = Indicates the server number such as 1 to 10

With a naming standard in place, you are ready to begin the design of your NDS tree. Remember that your naming standard will make administration of your tree easier and will provide you with consistent naming across your entire organization.

Designing NetWare Directory Services

NetWare Directory Services Tree

"He that plants trees loves others beside himself." Thomas Fuller (Gnomolgia 1732)

After you are familiar with NDS objects and have devised a naming standard, you are ready to begin designing your NetWare 4.1 tree. This section will outline the steps involved in the design of your NDS tree and explore different design options. Once you have a design that meets your needs you can begin the installation or migration to NetWare 4.1. Keep in mind that you can always make changes to your tree later, but a properly designed tree will almost never need significant changes. Modifications may be necessary at some point to meet the needs of your changing organization.

When designing your NDS tree, three goals should help drive the design decisions you make. The three goals are:

▸ Organize the network resources in your company for ease of access.

▸ Provide a blueprint for a consistent rollout of NetWare 4.1.

▸ Provide flexibility to the design to reflect corporate changes.

Organize the Network Resources in Your Company

One of the primary objectives for designing the NDS tree is to organize the network resources in your company. The NDS tree should be arranged to reflect the location and placement of the network resources in the network. The network resources should also be placed in the tree to provide easy access to both users and administrators. The goal of the NDS tree design then is not to reproduce the organization chart of the company but to represent all users and resources in such a way that allows them to work efficiently and easily. Notice in Figure 5.1 how users and resources are grouped together for ease of access. Some users need access to a particular printer or server while a larger group may need access to an e-mail server. The e-mail server is placed higher in the tree so that all users needing that resource have easier access to it.

FIGURE 5.1

Resources and users are grouped near each other in the ACME tree. The OPS division has a NetWare 4.1 server dedicated for e-mail.

[ROOT]

O=ACME

OU=NORAD   OU=RIO   OU=CAMELOT   OU=TOKYO   OU=SYDNEY

OU=OPS

E-MAIL SERVER
FOR THE UPS DIVISION

OU=FIN   OU=DIST

SERVER FOR THE
FINANCE DIVISION — FIN-SRV1

MMorgan

Gunevere
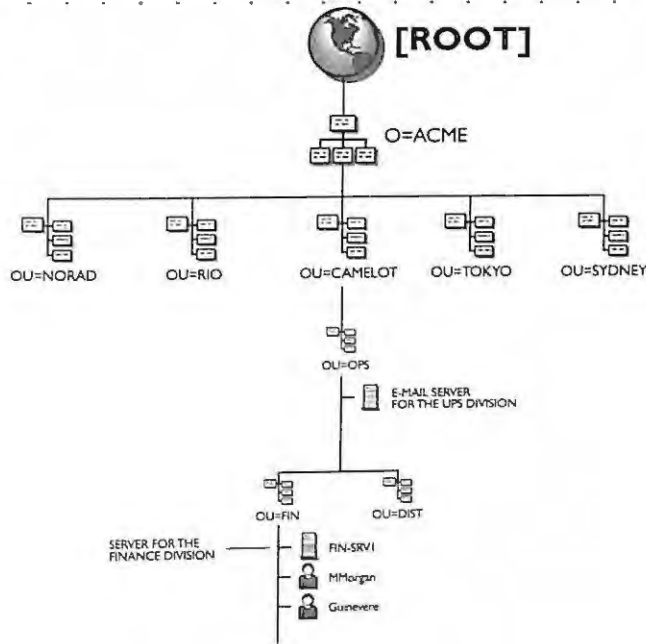
Provide a Blueprint for the Consistent Rollout of NetWare 4.1

Your blueprint for NDS installation includes your naming standard, tree design, and migration strategies. With this information in hand, you will have a guide for the installation of all your NetWare 4.1 servers. For example, the tree design will give you the location where an NDS server should be installed. An administrator can review the naming standards and know exactly how a server should be named. During the rollout of NetWare 4.1, some companies may be forced to implement more than one NDS tree on the network, with the intent to merge them together in the future. Creating more than one NDS tree is made much easier if the multiple trees are based on the same NDS tree design.

In addition, your blueprint gives you a clear map of how to migrate an entire corporation to NetWare 4.1. Initially, your migration will begin with a few servers and may eventually involve hundreds. Your design can be the master plan for migration of all servers to NetWare 4.1.

Provide Flexibility to the Design to Reflect Corporate Changes

Another objective of the NDS tree design is to provide flexibility for both users and administrators. This simply means that as the corporation changes, both organizationally and physically, you will be able to facilitate those changes easily without making large modifications to your tree structure. For example, you may want a design that allows for easy moves of users and containers. This objective can be met by designing the NDS tree as described in the following sections.

These goals seem trivial but they become very important when you are considering the NDS design issues. One classic example that seems to always come up is that different groups or individuals in your organization want the NDS tree designed to meet their specific situation. These political issues can best be handled by referring back to your design goals. For example, how does a particular design approach help you organize your network resources? In other words, if you have specific goals to fall back on, you can deal with the political issues by applying each issue against your design goals. Keep in mind that the network is installed to serve a business purpose and your job as an administrator is to see that that purpose is being met. If you design your NDS tree with this goal in mind, many of the political issues can be removed from this process.

CONSULTING EXPERIENCE

Politics should not get in the way of your tree design, but sometimes it does. Not long ago an agency was in the process of designing its NetWare 4.1 tree. During the design process, some individuals insisted that they be represented as their own organizational units in the tree because of where they appeared organizationally in their agency. They incorrectly assumed that managerial positions or positions of authority should receive special attention in the design of the tree. As important as these individuals may be, they do not function as their own organizational units in the tree. Regardless of their titles, they are users and, thus, share resources with other individuals on the same floor or building.

NDS Tree Design Tasks

This section introduces the tasks associated with the NDS tree design. The tasks or steps provide an easy-to-understand and effective method for designing your NDS tree. The tree design consists of five steps:

1. • Gather your network and company documents.

2. • Design your tree in the shape of a pyramid.

3. • Design the top level of your tree.

4. • Design the bottom level of your tree.

5. • Make modifications to the tree based on your needs.

While there is no absolute or exact way to design an NDS tree, some methods are definitely more efficient than others. Efficiency is achieved when the tree is stable, and the design provides for the least amount of NDS traffic possible.

These methods are presented in the following sections to help you design an efficient NDS tree. The methods presented here have been developed by Novell Consulting Services and have been successfully implemented by hundreds of their large and small customers. The design guidelines presented here work for small and large companies regardless of any unique requirements.

It is important to define what we mean by a small and large company. For the purposes of the examples given in this chapter, a small company consists of 5 servers or fewer (with no wide area network (WAN) connections) and fewer than 500 users. Certainly, there is no written law that this definition is always true; however, a network of this size has fewer design ramifications than a network with thousands of users and a very complicated network infrastructure.

A large network then is anything greater than 5 servers with a WAN connection and 500 users. As you read through this chapter, you will be able to determine which category (small or large) you fall into. So, let's begin the design.

GATHER THE CORPORATE DOCUMENTS

Some company information or documentation is necessary to help you initiate an NDS tree design for your corporation. The following documents are listed and discussed below in order of importance.

The wide area network (WAN) diagrams for your company

Figure 5.2 shows the ACME WAN layout map. The WAN layout or physical layout usually consists of all your major hub locations that are interconnected with routers and bridges. Notice in ACME's WAN layout map that all five main sites are shown with their router connections and the speed of these links in kilobits per second. Your WAN layout map may look similar or it may include the link speeds of your satellite offices.

F I G U R E   5.2

The ACME Physical WAN layout map includes important information such as the major hub locations of its network as well as router connections and link speeds.
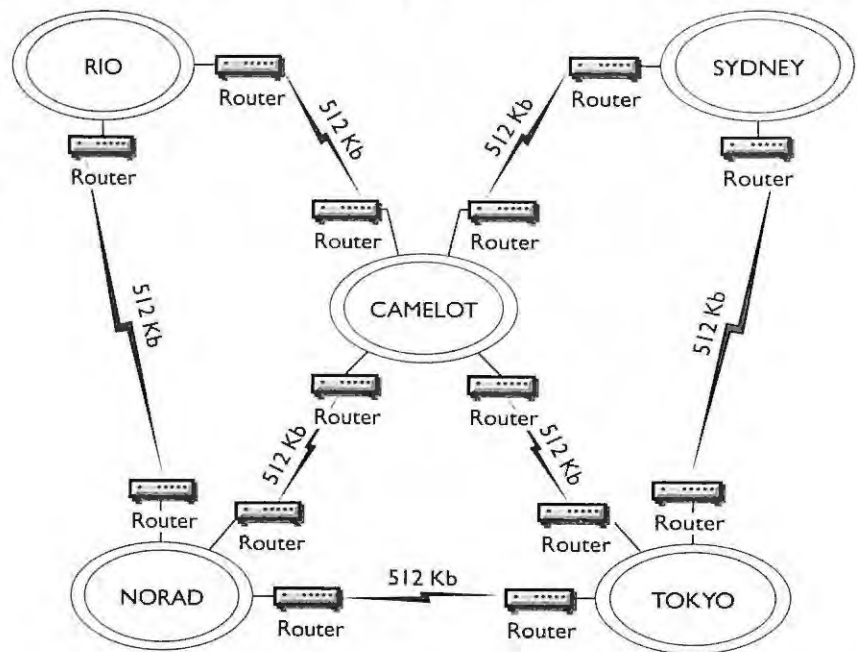
Your WAN documentation may consist of more than one compact diagram. Don't worry if your documentation consists of many pages of information. The key here is to understand how your infrastructure is organized and where the major hub locations exist. These documents are necessary for the upper layer design of your tree as we will explain. Most companies have some sort of wide area network diagrams available. We

recognize that a WAN network diagram is a document always in flux at most companies. However, try to obtain the latest WAN map from your staff that manages your network infrastructure.

The campus network maps for all your individual locations

Along with your WAN layout maps, campus diagrams will provide a further breakout of locations of your hub sites. This type of documentation varies from company to company. Some companies show the entire hub and campus diagrams together, while others separate the information because of the size of their network. A campus network map may show you a campus, such as the NORAD campus shown in Figure 5.3.

The ACME campus map for NORAD shows information such as an FDDI ring and routers connecting this site's buildings. The campus diagram may also show buildings and their interconnections. Again, this information, along with the physical WAN map, describes your WAN/LAN infrastructure. This preliminary information is necessary to build the foundation for your NetWare 4.1 environment.

FIGURE 5.3

The campus diagrams show a continuation of your physical WAN diagrams. Information can include connection types between buildings such as FDDI, bridged token rings, 10baseT, fiber, and so on.
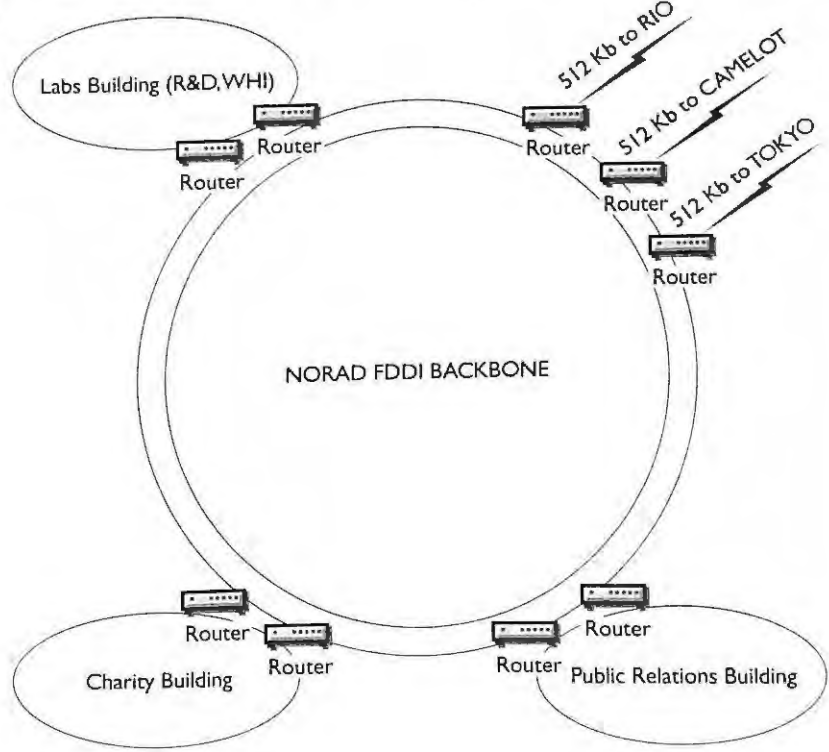
A list of sites or locations within your company

The next item to obtain for your tree design is a list of sites or locations of your network. Sometimes this information is included with your other documentation discussed previously. In large companies, there is usually a list of sites with their locations. For example, the ACME network has a site known as CAMELOT, which is shown as a single site on the WAN map for ACME. Further documentation, however, shows sites within the CAMELOT area, such as an operations center (OPS) located at the south end of the city and a public relations office (PR) center located in the downtown district. An office for charitable contributions (CHARITY) is also found at the downtown location. Figure 5.4 shows an example of the sites for the CAMELOT location.

F I G U R E   5.4

The CAMELOT site includes three buildings under its designation: Operations (OPS), Public Relations (PR), and CHARITY.
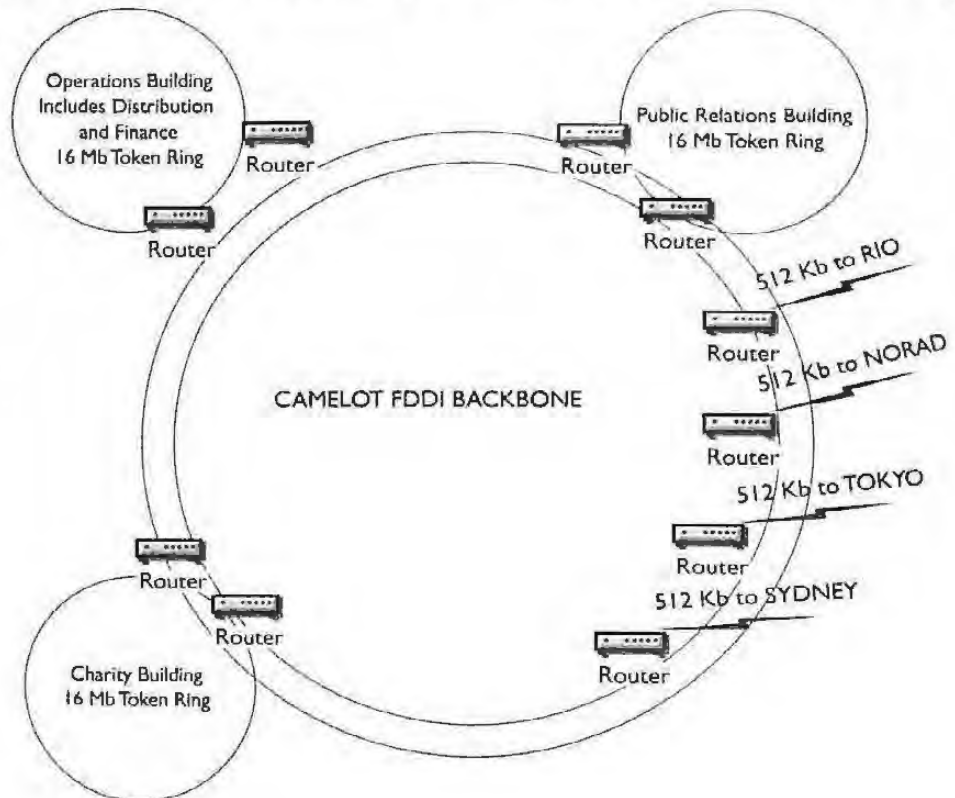
Resource list that includes your file servers, printers, and other major network resources

A resource list may be included with your LAN maps for each site. This list gives important information about the servers and printers found in each region, site, building, or department. Table 5.1 shows the resource list for ACME with its servers, printers, and print queues. This information is later used in your tree design for the placement of these resources.

> **NOTE**
>
> Keep in mind that the information we are referring to in this resource list will be either NetWare 3 server information or other operating systems that will be migrated to NetWare 4.1. This list enables you to know beforehand where objects need to be created in the NDS tree. Notice also that some of the naming standards shown in the list can be used in your NetWare 4.1 environment while others may need some slight changes.

TABLE 5.1

ACME Resource List

LOC/DIV	SERVER NAME	PRINTERS/QUEUES
NORAD	NOR-SRV1	
CHARITY	NOR-CHR-SRV1	HP4SI-PQ1
		HP4SI-P1
LABS	LABS-SRV1	
R&D.LABS	R&D-SRV1	HP4S1-PQ1
	R&D-PS1	HP4S1-P1
POLL.R&D.LABS		HP4S1-PQ2
		HP4S1-P2
NUC.R&D.LABS		HP4S1-PQ3
		HP4S1-P3
VR.R&D.LABS		HP4S1-PQ2
		HP4S1-P2
WHI.LABS	WHI-SRV1	CANONBJ-PQ1
	WHI-SRV2	CANONBJ-P1

(continued)

ACME Resource List
(continued)

LOC/DIV	SERVER NAME	PRINTERS/QUEUES
WHI.LABS	WHI-SRV3	
	WHI-PS1	
PR	NOR-PR-SRV1	HP4S1-PQ1
	NOR-PR-PS1	HP4S1-P1
		CANONBJ-PQ2
		CANONBJ-P2
RIO	RIO-SRV1	
CHARITY	RIO-CHR-SRV1	HP4S1-PQ1
	RIO-CHR-PS1	HP4S1-P1
ADMIN	ADMIN-SRV1	CANONBJ-PQ1
	ADMIN-SRV2	CANONBJ-P1
	ADMIN-ADM-PS1	
FAC.ADMIN	FAC-SRV1	HP4S1-PQ1
	FAC-PS1	HP4S1-P1
AUDIT.ADMIN	AUDIT-SRV1	HP4S1-PQ1
		HP4S1-P2
MRKT.ADMIN	MRKT-SRV1	HPIII-PQ1
		HPIII-P2
PR	RIO-PR-SRV1	HP4S1-PQ1
	RIO-PR-PS1	HP4S1-P1
		CANONBJ-PQ2
		CANONBJ-P2
CAMELOT	CAM-SRV1	
CHARITY	CAM-CHR-SRV1	HP4S1-PQ1
	CAM-CHR-PS1	HP4S1-P1
OPS	OPS-SRV1	CANONBJ-PQ1
	OPS-PS1	CANONBJ-P1

	LOC/DIV	SERVER NAME	PRINTERS/QUEUES
TABLE 5.1 *ACME Resource List (continued)*	**FIN.OPS**	FIN-SRV1	HP4SI-PQ1
		FIN-SRV2	HP4SI-PQ2
		FIN-SRV3	HP4SI-P1
		FIN-PS1	HP4SI-P2
		FIN-PS2	
	DIST.OPS	DIST-SRV1	HPIII-PS1
		DIST-SRV2	HPIII-PQ1
			HPIII-P1
	PR	CAM-PR-SRV1	HP4SI-PQ1
		CAM-PR-PS1	HP4SI-P1
	SYDNEY	SYD-SRV1	
	CHARITY	SYD-CHR-SRV1	HP4SI-PQ1
		SYD-CHR-SRV1	HP4SI-P1
		SYD-CHR-PS1	
	HR	HR-SRV1	HP4SI-PQ1
		HR-SRV2	HP4SI-P1
		HR-PS1	
	MEDICAL.HR	MED-SRV1	HP4SI-PQ1
		MED-SRV2	HP4SI-P1
	FOOD.HR	FOOD-SRV1	HP4SI-PQ2
		FOOD-SRV2	HP4SI-P2
	SHELTER.HR	SHELT-SRV1	HP4SI-PQ3
		SHELT-SRV2	HP4SI-P3
	PEACE.HR	PEACE-SRV1	HPIII-PQ4
			HPIII-P4
	PR	SYD-PR-SRV1	HP4SI-PQ1
		SYD-PR-PS1	HP4SI-P1

(continued)

ACME Resource List
(continued)

LOC/DIV	SERVER NAME	PRINTERS/QUEUES
TOKYO	*TOK-SRV1*	
CHARITY	*TOK-CHR-SRV1*	*HP4S1-PQ1*
	TOK-CHR-SRV1	*HP4S1-P1*
	TOK-CHR-PS1	
CRIME	*CRIME1-SRV1*	*HP5-PQ1*
	CRIME1-SRV2	*HP5-PQ1*
	CRIME1-PS1	*HP4F1-P2*
BLUE.CRIME	*BLUE-SRV1*	*HPIII-PQ1*
	BLUE-PS1	*HPIII-P1*
VIO.BLUE.CRIME	*VIO-SRV1*	*HP4SI-PQ2*
	VIO-SRV2	*HP4S1-P2*
ENV.BLUE.CRIME	*ENV-SRV1*	*HP4S1-PQ3*
		HP4S1-P3
THEFT.BLUE.CRIME	*THEFT-SRV1*	*HP4SI-PQ4*
		HP4SI-P4
WHITE.CRIME	*WHITE-SRV1*	*CANONBJ-PQ1*
	WHITE-SRV2	*CANONBJ-P1*
	WHITE-PS1	
CYBER.WHITE.CRIME	*CYBER-SRV1*	*HPIII-PQ2*
		HPIII-P1
POL.WHITE.CRIME	*POL-SRV1*	*HP4SI-PQ3*
		HP4SI-P3
FIN.WHITE.CRIME	*CRIME1-FIN-SRV1*	*HP4SI-PQ4*
		HP4SI-P4
		HP4SI-P5
PR	*TOK-PR-SRV1*	*HP4S1-PQ1*
	TOK-PR-PS1	*HP4S1-P1*

The organizational chart or similar document for your company

The last piece of information that is helpful in designing your tree is your company's organization charts. Your company may have many pages of organizational charts or a single chart. Your main purpose in obtaining the organization chart is to determine what divisions, departments, or other groups need to be created at the bottom layer of your tree. Figure 5.5 shows ACME's organizational structure. It is the input from this structure that will determine the lower layers of the NDS trees.

FIGURE   5.5

The organizational structure for ACME gives us information about departments, divisions, and workgroups that may need to be represented in your NDS tree structure.

Gandhi
Albert Einstein
King Arthur
Sherlock Holmes
George Washington
Board of Directors

| Gandhi
Human Rights | Albert Einstein
Labs | King Arthur
Operations | Sherlock Holmes
Dr Watson
Crime Fighting | George Washington
Admin |

REST OF THE ACME ORGANIZATIONAL CHART
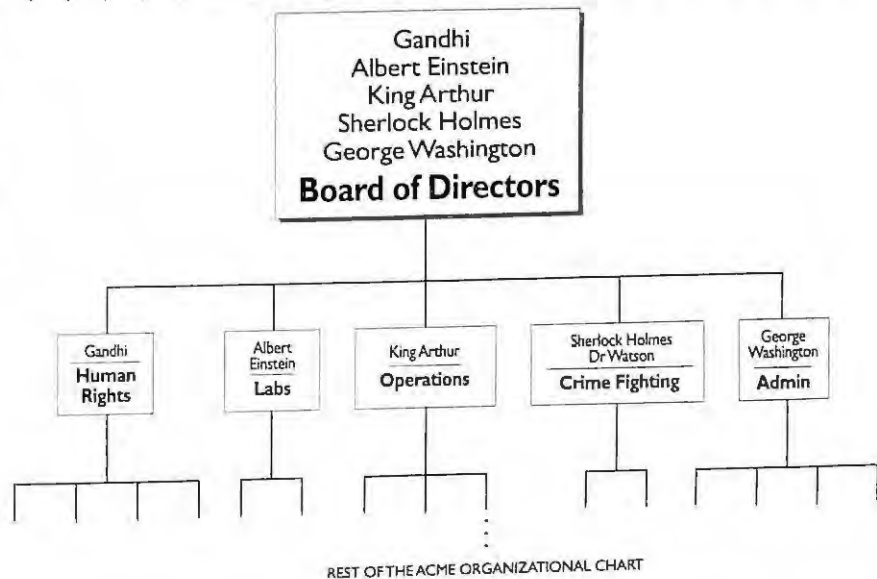
Typically, the most difficult task in designing the NDS tree is gathering all these corporate documents. Yet, an even more difficult task is trying to design the tree without them. These documents are the inputs into the NDS tree design process. Try to obtain as many of them as you can.

Each of the documents is used during different phases of the design process, and some documents are more important than others. The most important company document is the WAN layout or map. The WAN map is required before starting the NDS design for your company because it is used to design the top of the NDS tree. If your company has campus networks at one or more locations, then documentation of the physical campus network is also needed. If you have a small network (based on our previous definition of a small company), then you may not have an WAN map, but you will probably have an organizational chart.

The organizational chart is also an important document. But, as you will see, the organizational chart is used only during the design of the bottom of the tree. After you have gathered the corporate documents, you are ready to start the planning of the NDS tree.

DESIGN THE TREE IN THE SHAPE OF A PYRAMID

The design of the NDS tree should take the shape of a pyramid or inverted tree. The pyramid design implies that you place most of the containers and objects at the bottom of the structure with fewer containers at the top. The pyramid design shape of the tree is logically split into two sections. First, you will design the top of the tree, with its appropriate containers, and second, design the bottom of the tree. Figure 5.6 illustrates how the pyramid design is split into the top and bottom sections of the tree.

FIGURE 5.6

The NDS tree should be designed like a pyramid with a top and bottom portion of the tree.
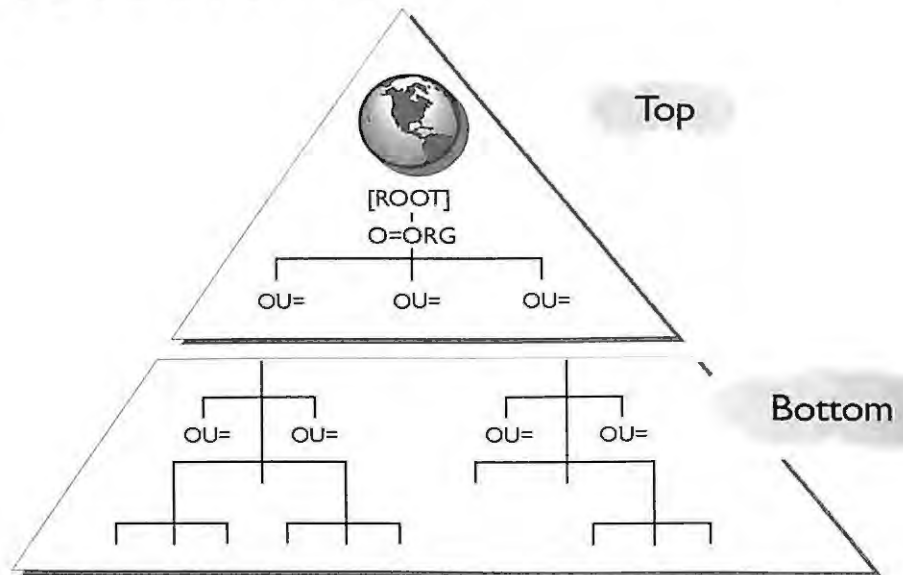
The advantage of a pyramid-shaped tree layout is that the top layers become the foundation (or static layers) upon which the bottom layers can be established. The bottom layers of the tree will be more dynamic, allowing for greater flexibility to change when your company changes. This design approach enables you to more easily make changes, such as moving users or subtrees.

Another advantage of the pyramid tree design is that the partitioning of the NDS database is more natural. The alternative to the pyramid design is to create a flat tree layout or structure that places all the objects in the top layers of the tree. As shown in Figure 5.7, a very flat and wide tree is not an efficient design approach because of how NDS communicates with its subordinate levels.

An efficient NDS tree will have more organizational units at the bottom and less at the top. Our recommendation for each level is not to exceed 50 organizational units per level in the tree. Even the largest companies with many branch offices will usually meet this requirement as explained later in this chapter.

TIP

FIGURE 5.7

A very flat and wide tree is not as efficient as designing an NDS tree with the pyramid shape. The design in this figure is not recommended.
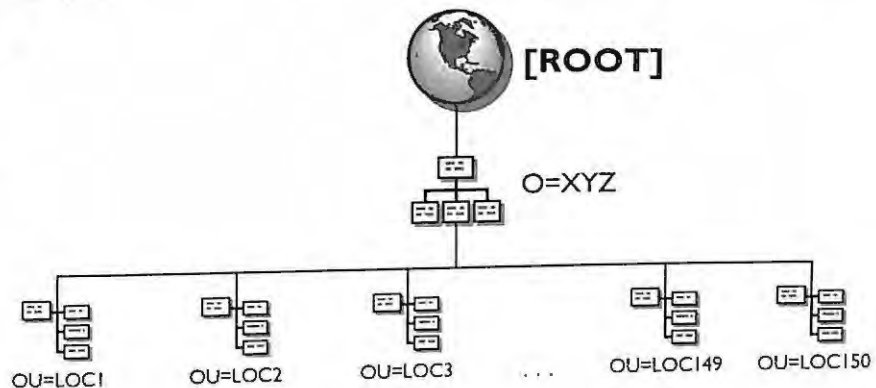
Having all the objects in the top of the tree also makes the tree rigid and inefficient for most large companies. A flat tree is not recommended primarily because of the way it has to be partitioned and replicated. Synchronization traffic of NDS on all servers in the tree is increased considerably with this type of approach. The tree partition and replica layout becomes extremely flat, which could cause many subordinate reference replicas to be created. Subordinate references are pointers between partitions and subordinate partitions and are automatically created by NDS. For more information about subordinate reference replicas, refer to the "NDS Partitions and Replicas" section later in this chapter.

As seen in Figure 5.8, the tree begins with the [ROOT] object at the top followed by the O=Organization object. After the O=Organization object(s), the next level is a layer of OU=Organization Unit(s). Typically, only a selected set of users and network resources is located in the top layers of the tree. For example, the ADMIN user object is located in the O=Organization object because the installation program automatically creates it.
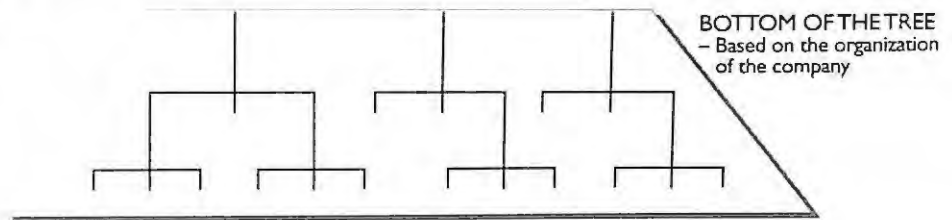
TOP OF THE TREE
– Based on the WAN Infrastructure

[ROOT]
O=ORG
OU=   OU=   OU=
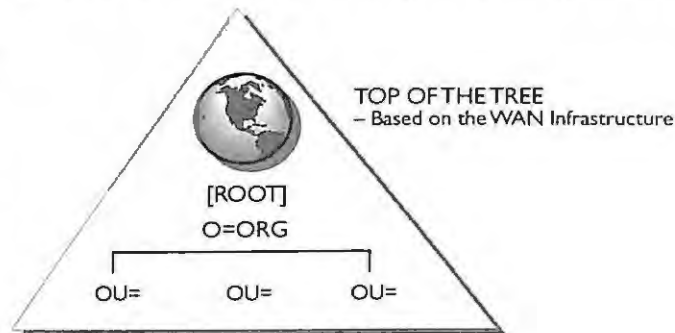
The bottom of the tree is defined by the local area network (LAN) and is based on the actual organization of your company from a departmental or divisional standpoint. In Figure 5.9, the bottom layers in the tree are most flexible if they represent the organizational structure of your company. This can be accomplished by using the divisions, departments, and workgroups in the corporation. The bottom layer OUs will hold the majority of your leaf objects such as the users, file servers, printers, queues, and other network resources. This approach provides the greatest flexibility because when you make changes to a department you will only affect the container(s) of that department. The rest of the tree is left unchanged. In contrast, if you have organizations represented at the top of your tree and you make changes to an organization, these changes can possibly affect your containers that are subordinate to the container you initially changed.

FIGURE 5.9

The bottom of the tree is based on the organization of the company. These lower layers offer flexibility to the tree design for moves and other changes.

BOTTOM OF THE TREE
– Based on the organization of the company

Both the top and the bottom of the tree are illustrated in Figure 5.10 and will be discussed in the following sections in greater detail.

FIGURE 5.10

The top of the NDS tree is based on the WAN or locations of your company and becomes the static portion of the tree. The bottom of the tree is based on the LAN and organization of your company and gives you the flexibility to make moves and other changes.

[ROOT]
O=ORG
OU=   OU=   OU=

Top
– WAN
– Geography
– Static

Bottom
– LAN
– Organization
– Flexible
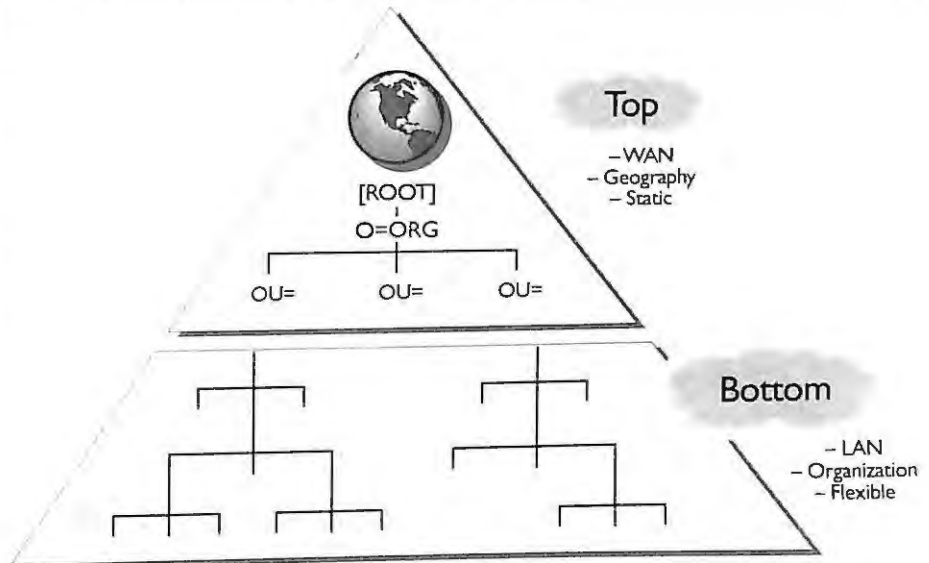
DESIGN THE TOP LEVEL OF THE TREE

The design of the top of the NDS tree is the most important because it is the foundation of the NDS tree. The rest of the tree will branch downward from the top. As we mentioned earlier in this chapter, the top level of the tree includes the tree name, which is the [ROOT] object, the O=Organization object, and the first layer of OU=Organization Units which, for most companies, will be based on your WAN infrastructure and locations. Figure 5.11 illustrates how we are taking the information from the WAN diagrams and designing the upper layer of the tree.

FIGURE 5.11

For the upper layer of the tree, match the major hub locations of your network to the first level of organizational units in your tree.
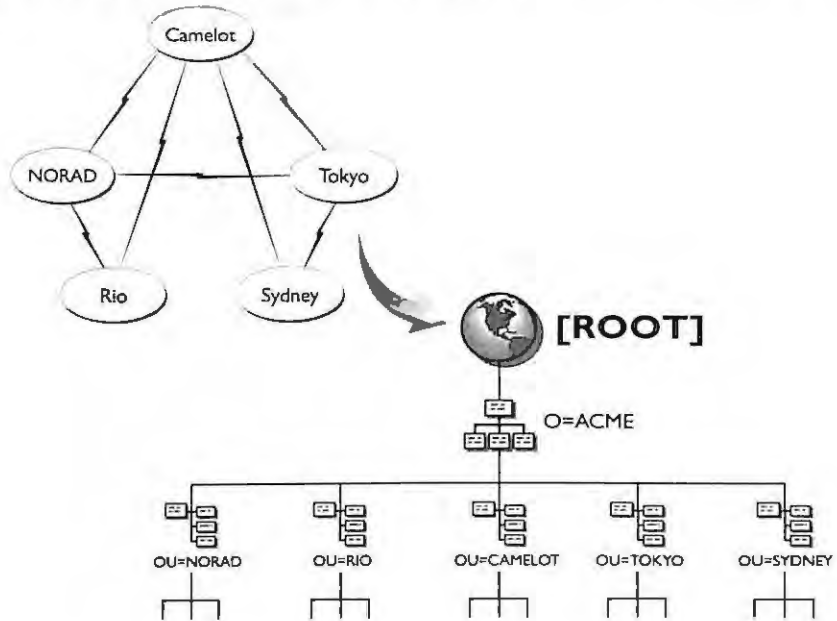
As shown in Figure 5.12, a guideline for the maximum number of containers at each level is 50. Although this number is not a strict rule, we have found that most companies can maintain greater efficiency for NDS synchronization if they stay at or below this range. For companies with many branch offices exceeding this value, try to insert a level of regional organizational units one level above to distribute the load of branch offices under these regional containers.

Most companies will be able to work within this guideline. However, for exceptionally large companies with thousands of branch offices, you may have no choice but to exceed this recommendation. If you must, pay special attention to partitioning and replication guidelines as presented in Chapter 6.

Tree Name or [ROOT] Object

The first entity that needs to be named in NDS is the tree itself. The tree name is also the name of the [ROOT] object, which is placed as the top-most object in your tree. The NDS tree name is displayed in many of the current NetWare utilities such as MONITOR, SERVMAN, and INSTALL. The tree name you choose can represent the company name plus _TREE. For example, our company is called ACME, so we chose to name the tree ACME_TREE, as shown in Figure 5.13.