If Microsoft Exchange 2000 Server can do it, you can do it too.

Packed with real-world tips and examples, this comprehensive guide delivers all the know-how you need to deploy, manage, and troubleshoot Exchange 2000 Server and Exchange applications. From step-by-step guidance on installation and configuration to incisive coverage of database management, Web storage, e-mail client configurations, and other advanced features, a crack team of Exchange experts shows you how to handle everyday administration issues — and harness the full power of Exchange Server communication and collaboration tools.

Inside, you'll find complete coverage of Exchange 2000 Server administration:

- Get a handle on Exchange 2000 Server components and protocols
- Work with Outlook, Outlook Express, POP3, and IMAP4 clients
- Develop a deployment plan and deal with topology, migration, and security issues
- Schedule and perform routine administration tasks using the book’s handy checklists
- Take advantage of Windows and Exchange utilities and troubleshoot problems
- Understand how to work with the Exchange application platform
- Harness tools such as event sinks and Digital Dashboard to administer Exchange applications

Robert Guaraldi, Mathew Often, Dr. Sam Gill, and Paul Guaraldi

“...and this book should be a key part of your library”
— Bob Jones, Microsoft Corporation

BONUS CD-ROM includes:
- PDF version of the book
- All sample code from the book
- Freeware versions of HyperSnap-DX and Acrobat Reader

System Requirements:
Pentium PC running Windows 95 or later, Windows NT 4 or later. See the CD Appendix for details and complete system requirements.

ISBN 0-7645-4782-8

$44.99 USA
$67.99 Canada
£36.99 UK incl. VAT

www.hungryminds.com
Exchange 2000 Server Administrator’s Bible

Robert Guaraldi, Mathew Often, Dr. Sam Gill, Paul Guaraldi
About the Authors

Robert L. Guaraldi is currently president and co-founder of @Ilmarin, LLC, a business consulting and investment company. Previously, he was the founder and president of Valinor, Inc. (one of the first Microsoft Solution Providers and one of the first ATECs) from 1982 to May 2000. Throughout his career, Mr. Guaraldi has been an early adopter of the Microsoft products and technologies that led to Microsoft Office and BackOffice, especially Exchange (he holds a low three-digit MCP number). In the early days of Exchange, Mr. Guaraldi was one of the few instructors authorized by Microsoft to deliver the ten-day Exchange server PSS class to Microsoft employees and large customers. He has been a speaker at numerous industry conferences and technology shows, including Microsoft TechEd, NetWorld, DevDays, Microsoft Explorer, and PC Expo. Over the years, he has participated in many advisory boards, including CRN, 3Com, Microsoft Solution Provider and ATEC, Connect magazine and Networld Boston. He has been a regular writer of articles and was lead author of the Exchange Server 5.5 Secrets book. He has been married for over 25 years, has four children, and continues to work on balancing the needs of his career in our exciting industry with the care and nurture of his family and friends.

Matthew Often has worked as a systems integration consultant for seven years and has worked on several enterprise-scale architecture and migration-related engagements for Fortune 500 companies in diverse industries. Notable clients include Pfizer, Merrill Lynch, ADP, and Starwood Hotels and Resorts. In addition to consulting, Mr. Often has served as a trainer of various Microsoft-related technologies, including Exchange, SQL Server, TCP/IP, and various Microsoft Operating Systems. Currently based out of Lucent Technologies’ Burlington, Massachusetts, office, Mr. Often works for the company’s Worldwide Services division in the Microsoft technical practice. His most recent engagement involved the design and rollout of Windows 2000 Active Directory at CVS Pharmacy in preparation for the installation of Exchange 2000. Mr. Often is Microsoft-certified as an MCSE+I and MCT. He is also a Compaq Accredited Systems Engineer with a specialization in Windows NT.

Dr. Sam Gill is a professor of Information Systems at San Francisco State University and Executive Vice President of @TecnoWiz LLC. Dr. Gill was a co-founder of DataWiz Centers, one of the first Microsoft Solution provider partners and the first Microsoft certified training partner. Dr. Gill publishes (articles, books, and courseware) and teaches in many areas of Microsoft technologies and development tools, including the Windows DNA architecture components, all the Visual Studio languages, and the Microsoft Solution Framework development methodologies. Dr. Gill’s career spans four decades of IT development, beginning with his work in the Israeli Ministry of Defense, and most recently as a consultant to Microsoft.

Paul E. Guaraldi is a graduate of the Massachusetts Institute of Technology. He is vice president and co-founder of @Ilmarin, LLC, a business consulting and investment company, and vice president and co-founder of DayMark Resources LLC, a technical and executive recruiting company. Mr. Guaraldi was formerly Executive Vice President of Valinor Inc., one of the first Microsoft Solution provider partners. Mr. Guaraldi has consulted on Microsoft technologies for over ten years and has spoken at various industry events, including NetWorld and Microsoft Explorer.

Contributors

Will Guaraldi spends much of his time steeped in the development community. He has worked with a vast array of technologies and languages since he started programming in 1986. Since 1995, he’s been reading RFCs and implementing protocols, giving him a deeply nuanced understanding of Internet application level protocols and interfaces. Mr. Guaraldi likes to spend his time in coffee lounges, where he makes token efforts at writing poetry, peruses technical books, and hums bars of POP3 and Circumstance.

Henry M. Allsworth is a Senior Network Systems Consultant with Lucent Technologies. He is currently a Microsoft Certified Systems Engineer and a member of Lucent’s Level 3 Microsoft Technologies Practice. During his time with Lucent, he has enjoyed many projects involving Windows NT, Windows 2000, BackOffice, and Exchange 2000 in a diverse set of enterprise scenarios.

Mal Swaim is the founder of AZ Computer Services, a system integrator focusing on Microsoft product solutions in Atlanta, Georgia. Prior to AZ, he was the SouthEast District Manager for Valinor, Inc., and worked at DCA. Mr. Swaim has broad and deep experience with small and mid-size customer systems, as well as large enterprise systems.
Credits

Acquisitions Editor
Nancy Maragioglio

Project Editors
Amanda Munz
Marcia Brochin

Technical Editors
Jim Kelly
David Gore

Copy Editors
Gabrielle Chosney
Erik Schwab

Project Coordinators
Joe Shines
Marcos Vergara
Emily Wichlinski

Graphics and Production Specialists
Sean Decker
Stephanie Jumper
Erin Zeltner

Quality Control Technicians
Laura Albert
Susan Moritz
Carl Pierce
Nancy Price
Linda Quigley
Marianne Santy
Charles Spencer

Permissions Editor
Carmen Krikorian

Media Development Specialist
Angie Denny

Media Development Coordinator
Marisa Pearman

Illustrator
Gabriele McCann

Proofreading and Indexing
York Production Services, Inc.
We dedicate this book to our parents. They taught us well. Their love persists. We remember. Always. George L. and Mary J. Guaraldi, Mary Often and John R. Often, and Henry and Carol Allsworth
I was quite surprised when Bob Guaraldi tracked me down by e-mail and asked if I would be interested in doing this Foreword, but the more I thought about it, the more sense it made to me. What I bring to the party is a real history with the product and customer issues and a field perspective. I have been in this industry for 27 years, first at IBM and then at Microsoft for the last 12 years. I am currently the Director of State and Local Government Sales and Consulting for Microsoft’s Eastern Region and previously ran the New York district for Microsoft. Now and during my five years as the General Manager of the New York District, I find myself in direct contact with the people who use messaging to build their businesses and government infrastructures. I also worked closely with the Microsoft Solution Providers and Training Centers that partner with my staff and customers to deliver on the promise of electronic messaging and collaboration. During that time I got to know some of the authors. Because the Guaraldis were principles in one of our Solution Provider Partners in New York and New Jersey, I got to know them and their team well. We collaborated with our teams to deliver quality Exchange solutions to metro-based companies with projects all over the world. We even used Bob to train some of the MCS staff in New York and New Jersey. It’s been a wild ride.

Today, Exchange 2000 and Windows 2000 deliver on the promise of integrating client-server, Windows and Internet architecture all at once. Coupled with other Microsoft Office, BackOffice, and development tools, they provide the development platform for the new millennium. Those companies that have taken advantage of the platform have already found it to be reliable, scalable, economic, and built to help tie and migrate legacy systems. They have found it has helped them gain control and drive down their total cost of ownership. This book does an excellent job of packaging the Guaraldi’s 18 years of experience working with Microsoft’s field and development staff to solve customer problems. Their experience with messaging, collaboration, Windows, development, and all the other knowledge that helps customers take advantage of these great products in the box with Exchange is hard to duplicate.

I love to see the proliferation of knowledge on Exchange. I appreciate the effort of these authors to share their experience and knowledge and give something back. For those companies that already have Exchange, this book will help ensure they are administrating it with insight and dexterity. For those companies that do not have Exchange yet, this book will illuminate the many reasons they should be considering it right now. For those individuals who are building a career in this industry, Exchange is a sure bet, and this book is a good start!

Bob Jones
Director East Region State and Local Government Sales and Consulting
Microsoft Corporation
Preface

Our work product, this book, was started when we received a pretty solid beta of Exchange 2000, but the preparation for it started many years ago, as each of us authors steered our careers toward a connected universe with messaging and collaboration as the unifying force. All those years of experience with design, administration, troubleshooting, product predecessors, underlying technologies, evolving standards, and just plain forbearance prepared us to write this book.

About This Book

Welcome to Exchange 2000 Server Administrator’s Bible!

Unlike other technology books, Exchange 2000 Server Administrator’s Bible was produced by a team of writers, an approach that let us better cover the spectrum of experience and insight appropriate for a product with such width and breadth. While many chapters had a single primary author, all were reviewed by — and received contributions from — multiple authors. We believe this method introduces you to the broadest set of insights and vision possible. We believe that is what you deserve.

We also believe you deserve a book that covers all the administrative aspects of Exchange and some of the consulting considerations. We know that some aspects of administration require knowledge of Windows 2000 administration. We tried to cover enough information about Windows 2000 to be explanatory, while referring you to other books for more thorough coverage. We also knew that some of you would be doing development or working with development teams. So we tried to cover the underlying concepts of development, how to administrate development done on your Exchange system, and referred you to books on development for more information. Lastly, we pushed the Exchange 2000 envelope a bit to include important underlying concepts.

How This Book Is Organized

We organized Exchange 2000 Server Administrator’s Bible into six different parts. To gain thorough knowledge of a topic, you may wish to read about it in more than one part of the book.
Part I — Introduction to Exchange Server

Part I covers the messaging and collaboration evolution that brought us to Exchange 2000, a glimpse of where Exchange is going, underlying standards, important concepts, and the lexicon of messaging. When we talk about administration of connectors later in the book, we will assume that you have some familiarity with the underlying concepts of messaging and collaboration. Read this chapter with no trepidation, even if you are a novice. No one memorizes all the RFCs and acronyms. After you’ve worked with them for years, they’ll become so familiar to you that they will stick in your memory. We recommend that you simply get acquainted with the concepts presented here; if you need to, you can refer back to this chapter as you work through subsequent chapters.

Part II — Exchange 2000 Server Components

In Part II, we take you on a tour of Exchange Server components. Because Exchange is so reliant on Windows 2000, we include Chapter 6 on Windows 2000 and Active Directory from an Exchange Administrator’s perspective. The rest of Part II is all Exchange. We want you to know about all of the components that make up an Exchange server and how they fit into the world.

Part III — Exchange 2000 Client Access

An Exchange messaging and collaboration system is not just a gaggle of servers: it is a client/server system. In Part III, we focus on the clients. Exchange is meant to work with a plethora of Microsoft and non-Microsoft clients, and this part will expose you to installing and configuring a number of the most popular ones.

Part IV — Planning, Deployment, and Installation

If you read this book sequentially, by the time you get to Part IV you know about all the components that make Exchange. In Part IV, we discuss how to put all the components and underlying standards together. Important concepts not often covered in Exchange books — such as mapping software to your hardware platform — are covered here. Doing a quick installation and doing a more thoughtful production installation of Exchange server are also covered. If you jump to Chapter 16 and follow the steps, you should be able to get a basic Exchange server installed pretty readily. If you take the time to read and follow the chapter on planning, an able team should be able to build a responsible plan for even the largest of environments.

Part V — Administration and Support

Part V covers the how-to of administration and support. We provide checklists, troubleshooting techniques, and a drill down on the utilities provided for Exchange 2000 administration.
Part VI – Administering Applications on Exchange

Many organizations use Exchange as a cornerstone of their messaging and collaboration development platform. This chapter introduces the components and concepts that make up that platform. It covers many concepts used to administer development on Exchange. It explains how to use some of the tools—such as event sinks and Digital Dashboard—to help you administrate Exchange. And it references you to whole books on Exchange development that carry the treatment further than we had room to go into here.

Appendix

The Appendix includes a document on the details of what is in the CD-ROM and an index.

Glossary

We included a glossary for quick lookup of the technobabble and acronyms you encounter in your day-to-day efforts in the world of Exchange.

CD-ROM

The CD-ROM that accompanies this book includes a copy of the entire book in PDF format. It also includes information on some third-party products, sample applications, shareware and word format copies of all the development code in the book. That way you can cut and paste the code you want to play with without retyping and running the risk of a typo.

How to Use This Book

Our experience is that books of this type are used differently by different readers. Some readers will try to learn about Exchange by starting at the beginning and reading right through. Most, however, will turn to specific chapters to get help in addressing a specific problem or doing something specific (such as installing an Outlook client). We wrote and organized the book with both approaches in mind. Even if all you want to do is find information on a specific topic, you may find it helpful to read some of the other associated information to ensure your Exchange project works as expected when you are done. The index, table of contents, and glossary will help you find the information you're looking for.

We encourage you to work with the product while you read this book so that you gain hands-on experience before you go into production with business critical systems. Many of us have been forced to figure things out under fire, but that's the hard way. For relatively small dollars, you can have a lab that enables you to build multiple servers and clients, and touch, experience, and learn about Exchange 2000 using our book as a guide.
Icons Used in This Book

The Caution icon flags items you should look out for and give special attention to when working with Exchange.

Cross-Reference points you to another section of the book, or some other book that might help you with the concept being discussed.

The Note icon calls your attention to an extraneous, but important piece of information.

This icon refers you to the CD-ROM, where you will find the item referenced.

We want you to learn from our investments in mistakes and successes. The Tip icon highlights our insights for your edification.

Adding to the Knowledge Bank

The information industry can make a difference for humans everywhere. Each of us should feel blessed to be part of it. It is one of the fastest growing industries in the world, and for many of us, it has provided an exciting and prosperous livelihood. One of our reasons for writing this book is to give something back. We thank all those who contributed to our knowledge base. We encourage all of you, as you grow in your knowledge and experience, to give something back. Share your knowledge in books, white papers, speaking engagements. Together, we will change the world for the better.
For those of us involved in the arduous process of creating our careers and writing this book, there are many to thank for their support and understanding. There are our spouses, siblings, parents, friends, children, and significant others. There are our spouses-to-be — especially Mat Often’s future wife, Laura Ort. There are the people at Hungry Minds who helped us and worked this process with us through to completion — Nancy Maragioglio and Amanda Munz. We acknowledge our never-ending gratitude to the many people who have contributed so much, or had so little of our time, while we were creating this work: Virginia Guaraldi, Carmela Gill, Catharine Guaraldi, Lydia Guaraldi, Ben Guaraldi, Mary J. Guaraldi, Gilda Devine, Mary G. Guaraldi, Joanne Wiegele, Barbara Miniaks, George L. Guaraldi, Jr., Michael Eaves, Katrina and EJ Quinonez, John Often, David and Libby, Hunter Often, Todd Finnemore, and Stacy.
# Contents at a Glance

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Part I: Introduction to Exchange Server</strong></td>
<td>1-33</td>
</tr>
<tr>
<td>Chapter 1: Introducing Exchange: Mainframes, Minis, Terminals, and the Internet</td>
<td>3</td>
</tr>
<tr>
<td>Chapter 2: The Past and Future of Exchange Server</td>
<td>11</td>
</tr>
<tr>
<td>Chapter 3: Review of Underlying Technological Concepts</td>
<td>33</td>
</tr>
<tr>
<td><strong>Part II: Exchange 2000 Server Components</strong></td>
<td>79-207</td>
</tr>
<tr>
<td>Chapter 4: Server Overview and Architecture</td>
<td>81</td>
</tr>
<tr>
<td>Chapter 5: Exchange System Manager</td>
<td>97</td>
</tr>
<tr>
<td>Chapter 6: Windows 2000 Dependencies</td>
<td>121</td>
</tr>
<tr>
<td>Chapter 7: SMTP Routing and the Message Transfer Agent</td>
<td>139</td>
</tr>
<tr>
<td>Chapter 8: Data Repositories</td>
<td>161</td>
</tr>
<tr>
<td>Chapter 9: System Attendant, Monitoring, and Message Tracking</td>
<td>175</td>
</tr>
<tr>
<td>Chapter 10: Enterprise Connectors</td>
<td>189</td>
</tr>
<tr>
<td>Chapter 11: Administrating Protocols</td>
<td>207</td>
</tr>
<tr>
<td><strong>Part III: Exchange 2000 Client Access</strong></td>
<td>227-533</td>
</tr>
<tr>
<td>Chapter 12: Outlook</td>
<td>229</td>
</tr>
<tr>
<td>Chapter 13: Outlook Express</td>
<td>261</td>
</tr>
<tr>
<td>Chapter 14: Using POP3 and IMAP4 with Exchange 2000</td>
<td>281</td>
</tr>
<tr>
<td>Chapter 15: Outlook Web Access</td>
<td>297</td>
</tr>
<tr>
<td><strong>Part IV: Planning, Deployment, and Installation</strong></td>
<td>315-533</td>
</tr>
<tr>
<td>Chapter 16: Performing a Quick Install</td>
<td>317</td>
</tr>
<tr>
<td>Chapter 17: Planning an Exchange Implementation</td>
<td>339</td>
</tr>
<tr>
<td>Chapter 18: Migration and Coexistence</td>
<td>367</td>
</tr>
<tr>
<td>Chapter 19: Mapping, Hardware, and Software</td>
<td>417</td>
</tr>
<tr>
<td>Chapter 20: Performing a Smart installation</td>
<td>431</td>
</tr>
<tr>
<td>Chapter 21: Conferencing Server</td>
<td>457</td>
</tr>
<tr>
<td>Chapter 22: Securing Your System</td>
<td>483</td>
</tr>
<tr>
<td>Chapter 23: Topology and Design Considerations</td>
<td>513</td>
</tr>
<tr>
<td>Chapter 24: Exchange on the Internet</td>
<td>533</td>
</tr>
</tbody>
</table>
Part V: Administration and Support .......................... 547
Chapter 25: Easing the Administration and Support Burden ............... 549
Chapter 26: Daily, Weekly, Monthly, and Yearly Administrative Tasks .......... 561
Chapter 27: Windows 2000 and Exchange Utilities .......................... 571
Chapter 28: Areas of Special Attention for Support ............................ 587
Chapter 29: Troubleshooting ................................................. 613

Part VI: Administrating Applications on Exchange ............. 633
Chapter 30: An Introduction to Exchange Applications ......................... 635
Chapter 31: Administrating Directory Integration with ADSI ................. 651
Chapter 32: Administrating Exchange Applications .......................... 679

Appendix A: What’s on the CD-ROM ..................................... 699
Glossary .............................................................................. 705

Index .................................................................................. 715
End-User License Agreement .................................................. 755
CD-ROM Installation Instructions ........................................... 759
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreword</td>
<td>vii</td>
</tr>
<tr>
<td>Preface</td>
<td>ix</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>xiii</td>
</tr>
</tbody>
</table>

### Part I: Introduction to Exchange Server

#### Chapter 1: Introducing Exchange: Mainframes, Minis, Terminals, and the Internet

- Early E-Mail Systems ................................................................. 3
- No LAN Is an Island ............................................................... 5
- Information, Not Data ............................................................ 8

#### Chapter 2: The Past and Future of Exchange Server

- Studying the Precursor to Exchange — MS PC Mail ............................. 11
  - E-mail capabilities .................................................................... 12
  - Groupware capabilities ............................................................ 12
  - Windows NT ................................................................................. 13
  - Microsoft Office and other BackOffice applications ......................... 13
  - Earlier versions of Exchange .................................................... 14
  - Upgrading from Exchange 5.5 to Exchange 2000 .................................. 15
- Overview of Exchange 2000 ................................................................ 20
- Active Directory Service .................................................................... 21
- Administration of Exchange ................................................................ 22
  - Microsoft Management Console .................................................. 22
  - Managing the rest ......................................................................... 23
  - New administrative model ............................................................ 23
  - Users and mailboxes ..................................................................... 23
- Connectors and Migration Tools ......................................................... 24
- Information Store ............................................................................ 24
  - Web Store ..................................................................................... 24
  - Multiple databases per server (Enterprise Edition only) ...................... 25
- Workflow Capabilities ....................................................................... 25
- Real-time Communication ................................................................... 26
  - Instant Messaging ......................................................................... 26
  - Chat (Enterprise Edition only) ...................................................... 26
- Evolving Key Standards ..................................................................... 26
  - Internet standards ........................................................................ 27
  - Data access standards ................................................................... 27
  - VPIM ............................................................................................ 27
  - SMTP Virtual Server ...................................................................... 27
  - Clustering (Enterprise Edition only) .............................................. 28
Microsoft Exchange 2000 Enterprise server ........................... 85
Microsoft Exchange 2000 Conferencing server ....................... 85
Overview of Exchange 2000 Features and Components ............... 85
Fault tolerance ....................................................... 86
Mailbox store and public folder store .................................. 87
System Attendant ...................................................... 89
Routing Engine service ................................................ 89
Windows 2000 and Active Directory .................................... 89
Programmatic platform components ...................................... 94
Utilities and monitoring ................................................ 95
Other components ..................................................... 95

Chapter 5: Exchange System Manager .................................... 97
About the Exchange System Manager .................................... 97
Starting Exchange System Manager ..................................... 97
Exploring the Organization Object ..................................... 99
The Top-Level Containers ............................................... 100
Focusing on Administrative Groups .................................... 101
  Design considerations ............................................... 101
  Creating additional administrative groups ......................... 102
  Using administrative groups ....................................... 102
Creating System Policies ............................................... 103
Applying a Policy ..................................................... 104
Managing Address Lists ............................................... 104
  Default address lists ............................................... 105
  Custom address lists ............................................... 106
  Offline address lists ............................................... 106
  Creating an address list ........................................... 107
  Defining a filter rule .............................................. 108
  Using a custom search ............................................. 109
  Securing address lists ............................................. 109
  Updating an address list .......................................... 109
Recipients ............................................................... 110
  Creating a Recipient object ....................................... 110
  Configuring a Recipient object ................................... 112
Administering Public Folders .......................................... 112
  Creating a public folder ........................................... 113
  Configuring a public folder tree .................................. 113
  Public folder permissions ......................................... 114
  Public folder replication ......................................... 116
  Public folder referrals ............................................ 117
Using Routing Groups and Connectors ................................ 117
  Creating a routing group .......................................... 117
  Creating a connector ............................................... 117
Chat Communities and Conferencing ................................ 118
Managing Administrative Security .................................... 119
  Permission inheritance ............................................ 120
  Exchange Administration Delegation Wizard ....................... 120
# Chapter 6: Windows 2000 Dependencies

- Windows 2000
- Active Directory
- Windows 2000 and Active Directory Components
- DNS
- Forests, trees, and domains
- Organizational units and groups
- Windows 2000 server roles
- Examining the concept of sites
- Administrative groups

# Chapter 7: SMTP Routing and the Message Transfer Agent

- Transport System Changes Since Exchange Version 5.5
- Protocols and Exchange Communication
- The enhanced SMTP protocol
- The X.400 protocol
- Routing Groups
  - Creating and administering a routing group
  - Linking routing groups
  - Routing group topologies
- Link State Information
- Putting It All Together
  - Routing within a server
  - Routing within a routing group
  - Routing to servers in other routing groups
- Managing Message Transport
  - Viewing the queues
  - Message tracking

# Chapter 8: Data Repositories

- Storage Groups and Multiple Databases
  - Creating storage groups
  - Creating databases
  - Building storage groups and database
  - Database content
  - Replication
  - Installable file system
  - Clustering
- Web Store
  - Indexing and searching
  - HTTP, WebDAV, HTML, and XML
  - Web Store administration
- Replication
- Installable file system
- Clustering

# Chapter 9: System Attendant, Monitoring, and Message Tracking

- Monitoring
- Creating a New Monitor
- Creating Notifications
- Script Notifications
- Monitoring the State of Links
- Message Tracking
- Other SA Activities
## Chapter 10: Enterprise Connectors

- Basic Connector Concepts ........................................ 189
- Cost .................................................................. 189
- Addressing and routing ......................................... 190
- Bridgehead servers ............................................ 191
- Making connectors work ....................................... 191
- Building a Routing Group Connector ................. 192
- Creating an SMTP Connector ................................ 195
- Building X.400 Connectors .................................. 198
- MTA transport protocols ..................................... 199
- X.400 TCP/IP Connector ..................................... 201
- RAS X.400 ...................................................... 205
- Other Connectors ............................................... 206

## Chapter 11: Administering Protocols

- Using NNTP ..................................................... 207
- Administering SMTP ........................................... 214
  - SMTP message format .................................... 214
  - SMTP virtual servers .................................... 215
  - Other SMTP implementation concerns .......... 220
- Administering IMAP4 .......................................... 220
- Administering POP3 .......................................... 222
- Controlling HTTP Access .................................. 222
- Managing RPC .................................................. 225

## Part III: Exchange 2000 Client Access

## Chapter 12: Outlook

- Introduction to Outlook 2000 .............................. 229
- Components of Outlook 2000 ............................. 230
  - Message store providers (POP3/IMAP/Exchange/MAPI) ............ 230
  - Address providers ........................................ 233
  - Personal folders .......................................... 235
  - FAX and other services ................................. 235
- Using Outlook or Common Outlook Configurations .... 235
  - Internet client ............................................ 236
  - Standalone application ................................ 236
  - Exchange client ........................................ 236
- Configuration Options ....................................... 236
  - Profiles .................................................... 237
  - Data storage ............................................. 239
  - Additional Tools menu options ....................... 242
  - Auto signature .......................................... 243
  - Views ....................................................... 244
  - Working remotely — What are my configuration options? .... 245
- Outlook Utilities .............................................. 251
  - Assistants and wizards ................................ 251
  - Find capability .......................................... 253
Part IV: Planning, Deployment, and Installation

Chapter 16: Performing a Quick Install

Installing a Server
Installing Exchange 2000
Installing a Second Server
Connecting with a Client
What’s Missing?

Chapter 17: Planning an Exchange Implementation

Plan, Test, Implement
How to Plan
Anatomy of an Implementation Plan
The Plan
Roles
Responsibilities
Planning team
Evaluation and testing team
Deployment team
Training team
Support team
Elements of a Successful Plan
Executive sponsorship and buy-in
Scope
Boundaries
Fiscal resources
Human resources
Schedule
Almost immutable laws
Politics
Risk management
Time lines
Building a Deployment Plan ........................................... 354
   Concept phase ................................................. 354
   Development ................................................... 356
   Pilot .............................................................. 357
Rollout ............................................................... 359
   Adequate on-site support ................................. 360
   Off-site support .............................................. 360
   Schedule ......................................................... 360
   Assigning resources ...................................... 361
   The moment of truth ....................................... 361
   Dealing with unexpected problems .................... 361
   Documentation and checklists ....................... 361
Post-Installation ................................................. 362
   Monitoring ....................................................... 362
   Morning-after user surveys .......................... 363
   Modification .................................................... 363
   Transition to a steady state ......................... 363
   Using milestones ......................................... 364
Areas of Special Attention ................................. 364
   Topology and design .................................. 365
   Exchange on the Internet ............................ 365
   Active Directory Service and Exchange 2000 .... 365
   Exchange 5.x integration and migration .......... 365
   Migration wizards ....................................... 366

Chapter 18: Migration and Coexistence ..................... 367
   Earlier Versions of Exchange ....................... 367
      Checking your existing environment .......... 368
      Introducing Exchange 2000 into your existing environment .................. 370
      A component-level look at upgrading ........ 371
      Directory coexistence and migration with Exchange 5.5 systems .... 371
   Mixed Mode versus Native Mode ................. 378
   Coexistence and Migration from Other Systems .. 379
      Installing a connector ......................... 380
      Using the Migration Wizard ................. 381
      Active Directory Account Cleanup Wizard ... 383
      cc: Mail coexistence and migration ........ 383
      Microsoft Mail coexistence and migration .... 392
      Lotus Notes coexistence and migration ...... 404
   Coexisting and Migrating Novell GroupWise .... 410
      Preparing the GroupWise environment ........ 412
      Configuring Exchange to connect to GroupWise ........... 413
      GroupWise migration to Exchange 2000 .... 413
      Other options for coexistence ............ 414

Chapter 19: Mapping, Hardware, and Software .................. 417
   Choosing Hardware ..................................... 417
      Microsoft hardware requirements .......... 418
      Minimum practical requirements ......... 418
      Choosing a physical drive .................. 420
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup: tape drives and WAN/LAN backup</td>
<td>426</td>
</tr>
<tr>
<td>Disk I/O channels</td>
<td>426</td>
</tr>
<tr>
<td>Network interface cards</td>
<td>426</td>
</tr>
<tr>
<td>Redundant power supplies</td>
<td>427</td>
</tr>
<tr>
<td>Uninterruptible power supply</td>
<td>427</td>
</tr>
<tr>
<td>Building Large Systems</td>
<td>427</td>
</tr>
<tr>
<td>Four-way active clustering</td>
<td>427</td>
</tr>
<tr>
<td>Scalability using front-end/back-end architecture</td>
<td>428</td>
</tr>
<tr>
<td>Microsoft BackOffice Resource Kit LOADSIM.exe</td>
<td>429</td>
</tr>
<tr>
<td>Chapter 20: Performing a Smart installation</td>
<td>431</td>
</tr>
<tr>
<td>Focusing on Installation Scenarios</td>
<td>432</td>
</tr>
<tr>
<td>Installing Exchange Server in a new messaging environment</td>
<td>432</td>
</tr>
<tr>
<td>Installing an Exchange 2000 server in an existing Exchange 5.5 environment</td>
<td>432</td>
</tr>
<tr>
<td>Installing a new Exchange organization intending to coexist</td>
<td>433</td>
</tr>
<tr>
<td>Creating a test or pilot environment for one</td>
<td>434</td>
</tr>
<tr>
<td>Preparing for the Installation</td>
<td>434</td>
</tr>
<tr>
<td>Hardware preparation and planning storage</td>
<td>435</td>
</tr>
<tr>
<td>Physical installation media</td>
<td>436</td>
</tr>
<tr>
<td>Windows 2000 preparations — local machine</td>
<td>437</td>
</tr>
<tr>
<td>Special accounts, installation rights, and permissions</td>
<td>438</td>
</tr>
<tr>
<td>Active Directory considerations</td>
<td>439</td>
</tr>
<tr>
<td>Naming conventions</td>
<td>440</td>
</tr>
<tr>
<td>Clustering considerations</td>
<td>441</td>
</tr>
<tr>
<td>Coexistence with other mail systems</td>
<td>441</td>
</tr>
<tr>
<td>Licensing</td>
<td>441</td>
</tr>
<tr>
<td>Performing an Installation</td>
<td>442</td>
</tr>
<tr>
<td>Surveying Server Changes and Verifying the Success of the Install</td>
<td>449</td>
</tr>
<tr>
<td>Ways to Automate</td>
<td>452</td>
</tr>
<tr>
<td>Unattended setup of Windows 2000</td>
<td>453</td>
</tr>
<tr>
<td>Integrating Windows 2000 service packs</td>
<td>453</td>
</tr>
<tr>
<td>Unattended setup of Exchange Server</td>
<td>454</td>
</tr>
<tr>
<td>Using Sysprep and imaging</td>
<td>454</td>
</tr>
<tr>
<td>Client automation</td>
<td>455</td>
</tr>
<tr>
<td>Chapter 21: Conferencing Server</td>
<td>457</td>
</tr>
<tr>
<td>What Is the Microsoft Conferencing Server?</td>
<td>457</td>
</tr>
<tr>
<td>Conferencing server features</td>
<td>458</td>
</tr>
<tr>
<td>Using Conferencing Server in your organization</td>
<td>460</td>
</tr>
<tr>
<td>Conferencing Server key components</td>
<td>461</td>
</tr>
<tr>
<td>Configuring, Planning, and Installing a Conferencing Server Site</td>
<td>468</td>
</tr>
<tr>
<td>System requirements</td>
<td>469</td>
</tr>
<tr>
<td>Permissions</td>
<td>471</td>
</tr>
<tr>
<td>Planning a Conferencing Server site</td>
<td>472</td>
</tr>
<tr>
<td>Installing and configuring Conferencing Server</td>
<td>473</td>
</tr>
<tr>
<td>Post-installation</td>
<td>478</td>
</tr>
</tbody>
</table>
Chapter 22: Securing Your System ........................................... 483

Windows 2000 Security ......................................................... 483
  Active Directory Service ................................................. 483
  Access control ............................................................. 485
  Auditing ................................................................. 486
  Kerberos ................................................................. 487
  Certificate Services .................................................... 488
  Encrypting File System ............................................... 490
  Internet Protocol Security ........................................... 490
  TCP/IP filtering ......................................................... 491
  Security Configuration Tool Set .................................. 492

Exchange 2000 Security ...................................................... 494
  Key Management Service .......................................... 494
  Enabling a user on the server and client ..................... 498
  Protecting content by sealing messages .................... 503
  Signing a message ................................................... 504
  Verifying a signature ............................................... 505
  Algorithms and keys ............................................... 506

Offline Usage ................................................................. 507
  Virtual Server security ............................................. 507
  Permissions ............................................................. 509
  Development security .............................................. 510
  Outlook 2000 security highlights ............................... 510
  Virus protection ....................................................... 511

Chapter 23: Topology and Design Considerations ................. 513

Your Business Organization ............................................ 513

Naming Conventions ...................................................... 515

Your Existing Messaging and WAN Environments .................. 518
  Physical server placement ....................................... 518
  Connectors ............................................................ 519
  Authentication and management ................................. 520
  Other areas to investigate .................................... 520

Server Roles ................................................................. 521
  Exchange servers ................................................... 521
  Windows 2000 servers ............................................ 521
  Other BackOffice servers ...................................... 522

Administrative and Routing Groups ................................ 523

Administrative Models .................................................... 524
  Distributed management model ................................. 524
  Centralized management model ............................... 524

Permission and Policies .................................................. 526
  User management ................................................... 526
  Administration and routing management ..................... 526

Data management .......................................................... 526

Real-time collaboration management .............................. 526
## Contents

- Other permission design considerations .................................................. 527
- Policies ........................................................................................................... 527
- Site and Administration Boundaries ............................................................. 527
- Active Directory Service and Replication ..................................................... 528
- Public Folders ............................................................................................... 529
- Other BackOffice Components ................................................................. 529
- Message Routing ......................................................................................... 530
- Bandwidth Management ............................................................................... 530
- Money, Security, Service Levels, and Policies ........................................... 530

### Chapter 24: Exchange on the Internet ............................................................ 533

- The Internet as an Exchange Backbone ....................................................... 533
- Message routing ........................................................................................... 537
- Newsgroups .................................................................................................... 539
- WebDAV ......................................................................................................... 539
- XML .............................................................................................................. 540
- Client/Server Access over the Internet ......................................................... 540
- Special Considerations ................................................................................. 545

### Part V: Administration and Support .............................................................. 547

#### Chapter 25: Easing the Administration and Support Burden ......................... 549

- Documentation ............................................................................................. 549
- Planning and design ..................................................................................... 550
- Ongoing support .......................................................................................... 551
- Standards and conventions document ....................................................... 552

- Standards and Conventions Process .......................................................... 553
  - General policies ......................................................................................... 555
  - Physical ...................................................................................................... 556
  - Logical ........................................................................................................ 557
  - Procedural .................................................................................................. 557
  - Application development .......................................................................... 557

- Administrative Assignments ........................................................................ 558
  - Choosing a candidate ................................................................................ 558
  - Assigning a backup ................................................................................... 559

- Backup and Restore ..................................................................................... 559
- Quality ......................................................................................................... 559
- Troubleshooting in a Standards-based Environment ................................... 560

#### Chapter 26: Daily, Weekly, Monthly, and Yearly Administrative Tasks ........... 561

- Focusing on Real-Time Tasks ..................................................................... 562
- Exchange Monitoring and Status Tool – What to monitor ......................... 562
- Task Manager .............................................................................................. 563

- Dealing with On-Demand Tasks .................................................................. 564
- Scheduling Tasks .......................................................................................... 565
  - Daily tasks ................................................................................................. 566
  - Weekly tasks ............................................................................................. 567
  - Monthly and annual tasks ......................................................................... 568
  - Record keeping ........................................................................................ 570
Chapter 27: Windows 2000 and Exchange Utilities .......................... 571
  Repairing Inboxes with ScanPST ........................................... 571
  Making Database Repairs with ESEUtil ................................ 573
  Using ISInteg to Regain Data Consistency .............................. 576
  Testing RPC Communication with RPCPing ............................... 578
    RPings ........................................................................ 578
    RPingc ........................................................................ 579
  Review and Repair Queues Using MTACheck .............................. 580
  Examining the Event Log .................................................... 582
  Exploring Other Utilities ..................................................... 583
    Error .......................................................................... 583
    FileVer ....................................................................... 584
    Mdbvue32 ..................................................................... 584
    Task Manager ................................................................ 584
    Replmon and RepAdmin ................................................... 584
    Schema Admin ................................................................ 585
    Windows Management Instrumentation ................................. 585
    Exchange 2000 Resource Kit .............................................. 585

Chapter 28: Areas of Special Attention for Support ......................... 587
  Performance and Scalability ............................................... 587
    The smallest organization ................................................. 588
    The intermediate-size organization ................................... 588
    Large organizations ........................................................ 589
    Managing memory .......................................................... 590
  Slow-Speed Links ............................................................. 591
  Mixed Exchange Environments ............................................. 592
  Roving, Wireless, and Remote Users ...................................... 594
  Backups and Restores ....................................................... 595
    Lost information ........................................................... 595
    Exchange-aware backups/restores ..................................... 596
    Where is the data? .......................................................... 597
    How Backup and Restore work ......................................... 598
    Backup strategies ........................................................... 601
    Backing up .................................................................... 603
    Restoring ...................................................................... 604
  Getting by Firewalls ............................................................ 612

Chapter 29: Troubleshooting ...................................................... 613
  The Troubleshooting Process ................................................. 613
    Getting a handle on the problem ........................................ 614
    Using the information available ......................................... 616
    Most common causes of problems ..................................... 619
    Other information tools .................................................... 620
    Basic troubleshooting method .......................................... 624
    Examples of problem-solving ......................................... 625
  Server Role Assignments ................................................... 629
    First server in a site ...................................................... 629
    Bridgehead servers ........................................................ 630
    News servers and conferencing servers ......................... 630
Part VI: Administrating Applications on Exchange 633

Chapter 30: An Introduction to Exchange Applications 635
- Overview of Collaborative Solutions 635
- Types of collaborative solutions 636
- Collaborative solution components 637
- Designing a collaborative solution 638
- Developing Collaborative Solutions with Outlook 2000 639
  - Designing forms 639
  - Working with views 643
  - Programming forms 644
- Developing Collaborative Solutions with Exchange Features 644
- Differences in Collaborative Solution Development Since Exchange 5.x 646
  - WebDAV 647
  - Web Storage System 647
  - Workflow engine 647
  - Workflow designer 648
  - Collaborative Data Objects 649
  - FrontPage 2000 649

Chapter 31: Administrating Directory Integration with ADSI 651
- Exploring the Active Directory System Interface 651
- Programming the Active Directory Object Model 654
  - ADSI fundamentals 654
  - High-level objects 662
  - Groups and users 665
  - Error handling 671
- What Is the Exchange Object Model? 672
  - Active Directory 673
  - Exchange 2000 schema 673
  - Managing recipients 674
  - Retrieving e-mail addresses 674
- How to Retrieve Active Directory Information 675
  - ADSIEdit 675
  - Active Directory Browser 677

Chapter 32: Administrating Exchange Applications 679
- Administrating Outlook 2000 Collaborative Solutions 679
  - Team folders 679
- Administrating Outlook Web Access 686
- Information store administration 687
- Digital Dashboard 690
- Administrating Collaborative Data Object Solutions 695
  - Store events 695
  - Event sinks 697
  - Managing security 697
Appendix A: What’s on the CD-ROM ........................................ 699
Glossary ........................................................................ 705
Index .............................................................................. 715
End-User License Agreement ........................................... 755
CD-ROM Installation Instructions ..................................... 759
Introduction to Exchange Server

Exchange 2000 is built on a rich foundation of standards and traditions in messaging and collaboration. In this part of the book, we explore that heritage. We take you on a quick journey through time, exposing you to the predecessor products, past and current standards, customer requirements, and design goals that were the impetus for Exchange. Then we provide a glimpse of the future. Once we are grounded in both the past and future of Exchange, we delve into some of the substrate technologies and standards that Exchange supports. We know from experience that administrators in messaging environments want some familiarity with those standards and concepts to make understanding the internal workings and troubleshooting of an Exchange messaging environment a little easier. This part provides that familiarity.
Some of us may feel like we’ve been in the information industry since the Egyptians built the pyramids and started using papyrus, but it hasn’t really been that long. This chapter examines the historical context that led to the public’s current collaborative messaging needs, products, and expectations. Take a trip in a dory down the Colorado River in the Grand Canyon and you can see eons of time layered in stone. In the technology industry, though, major changes cover much shorter periods of time—in the case of electronic messaging and collaboration, a scant 30 or 40 years, and in the case of the Internet effect, just a few years. But these immense changes are nothing compared to what will occur over the next few years.

Early E-Mail Systems

During the heyday of IBM, DEC, Data General, and WANG—when mainframes and minis were prevalent (late 1960s, 1970s, and early 1980s)—numerous companies attempted to create sophisticated e-mail systems. E-mail quickly became the crucial communication medium in many companies. Various manufacturers came up with their own proprietary systems, all of which were based on data structures maintained on the central mainframes or mini computers, and some of which had certain common design components (eventually leading to some standards). Although it was a challenge to administer
these early messaging systems because of their limitations, in some ways they were easier to administer than later systems because they were built on a single central machine. In general, people connected to them using dumb terminals, which were capable of only text-based messaging.

Text-based messaging is not at all like messaging today. You had no attachments, rich text, or ability to embed anything—just green or gray letters.

Because early e-mail was text-based, files were smaller than those we see today, which suited the limited Local Area Network (LAN) and Wide Area Network (WAN) bandwidth available at that time. As the pace of technical innovation accelerated, it was no longer economical or necessary to buy one massive mainframe or mini computer, nor was it necessary to be bound to the central mainframe or mini computer architecture for messaging.

As ironic as it is now to think of using an intelligent device (the PC) solely to emulate dumb devices, it made a lot of sense to many people at the time. As corporations expanded their systems, the last thing their employees wanted were six or seven machines on their desk when each was able to run only one application. For one thing, most desks aren’t large enough to hold that much equipment. The use of terminal emulation software on a PC (software that made the PC look like an American Standard Code for Information Interchange [ASCII] or 3270 terminal) enabled the employee to access many different systems using one multipurpose machine. Since most of the real information, and some of the best messaging software, was on the mainframe, employees saw real value in emulation. Administrators also saw value in having fewer machines to support at the desktop, and that meant each machine had to be able to run multiple applications successfully. Companies like 3Com saw the light and added e-mail application software to their disk/file service systems. Now, one desktop could do terminal emulation, share files and printers from a server, and do e-mail. Companies like cc:Mail and Courier built enhanced PC LAN–based mail systems. Soon, e-mail on a PC LAN was as good as anything on a mainframe or mini from the end-user’s perspective even though they were still quite difficult to administer. Pressure built for companies to move toward messaging systems that were on these new LAN-based systems rather than mainframes or minis. Many used the PC-based terminal emulation software to ease the migration of e-mail off of the minis and mainframes and onto the PC platform gradually, rather than doing it in one fell swoop.

The differences in topology of the PC LAN and Mainframe models are shown in Figure 1-1. On the mainframe side, all the terminals are connected to the mainframe in a star topology and run emulation. All messages are stored at all times on the mainframe. On the PC LAN, all devices connect anywhere on the backbone in a bus topology. Messages are sometimes on the local PC in temporary or more permanent storage, or they are on one of the e-mail servers on the LAN.
No LAN Is an Island

Though the end-user experience in PC messaging was getting increasingly robust, administration and messaging from one company to another was becoming exceedingly difficult for a number of somewhat unrelated reasons. First, there were no data communication lines between companies and no good way to connect them to LANs. Second, systems—such as operating systems, e-mail systems, and so on—varied between companies. Because there were few standards, and those that existed did not do a particularly good job of addressing the rapid pace of innovation and advancement in the messaging business, getting the different systems to talk to each other was often very difficult. Even if communication lines existed and standards were in place when many PC servers were used to replace one mainframe or mini, administration became a chore. In addition, because there were so few application interface standards (and the ones that did exist were primitive), application development that relied on a PC e-mail transport system remained a significant problem.
Companies, government, and universities acted as catalysts for smart, community-minded technologists to attack the issues in standards bodies. In the mid-1970s, the International Telecommunication Union (originally CCITT, and now ITU) created a series of messaging standards that helped ease the burden of system-to-system communications. Companies began to adopt portions of the X.25, X.400, and X.500 standards in their e-mail products. While these standards helped a great deal, many software companies weren’t implementing them fully or consistently, and in many cases, the standards did not go far enough to address exploding new requirements thoroughly. For instance, in the early X.400 specification, there was no clear dictate on how to handle rich text attachments. But users were clamoring for this capability, so each manufacturer found a field in the header that was not specified and used it for this purpose. Of course, this made interoperability a challenge, as they did not all choose the same field or even implement their choice in the same way. Another challenge for the industry then (and to some degree today) was that many messaging specialists knew a lot about PC messaging but almost nothing about WAN protocols and communications.

As the 1980s ended and the 1990s began, the Internet, as well as private intranets, helped to solve many of the problems facing messaging system developers. For example, messaging experts solved the “getting the business connected” problem by making it viable to create an expansive wide area network that lets data circle the globe. In a private intranet, companies could contract leased lines and satellite circuits from end to end for their exclusive use. With the Internet, leased lines or satellites could be connected—not just to another node of the company—but into a public network infrastructure, which let companies communicate with anyone on the planet that was also connected to the network. Yet companies still lacked the software to make it all work reliably and universally. Today, most companies (and many individuals) have access to reliable and cost-effective Internet physical connectivity of a decent bandwidth, and the industry has the software to take advantage of that physical connectivity.

Who Should Administer the Corporate WAN?

Many of us who have been in the messaging business for some time have found that there is a lot you cannot do without knowing the WAN side of the business. Some of us even believe that messaging people should own the WAN (at least the routing). But traditionally, most companies have not been organized that way. For many years, messaging personnel were kept away from the corporate WAN.

This issue was very real in the early days of PC computing. For example, at the first Microsoft TechEd conference in March of 1993, Bob Guaraldi, one of the authors, was asked to give a talk about enterprise PC mail. While pleased with the honor, he was also concerned about the challenge this presented. When he inquired why Microsoft had chosen him, the answer was amusing. Microsoft said that finding presenters who knew about Microsoft PC Mail was easy. Finding presenters who knew about WANs was also easy. But finding presenters who had experience with both large enterprise Microsoft PC Mail systems and WANs—as well as being willing to speak about them at TechEd—was difficult.
The surge in PC-based e-mail systems, and the incredible growth of the Internet, helped create the pressure to solve the second problem—the “compatibility and personal connection conundrum” (i.e., end-users expect that e-mail created using e-mail programs from different manufacturers must be able to easily connect and work together)—by giving software companies an incentive to support Internet protocol standards. It became obvious that if software companies didn’t support these standards, they would be left behind. By using both Internet communication protocols and the X. standards from ITU, diverse systems could deliver compelling business and personal utility.

Standards and techniques for centralized management started to gain momentum in the early 1990s and have been evolving ever since. For example, Simple Network Management Protocol (SNMP) and the concept of one centralized, standards-based, manageable Directory System have become more widespread—though they still haven’t gained universal adoption. Hardware features, robust WANs, and new standards for almost every aspect of messaging (like support for clustering) now make it possible to centrally administrate messaging servers on the other side of the world without being physically there.

As messaging became more standards-based, many third-party vendors pushed for a platform on which to develop their software without having to rewrite the basic messaging and collaboration system. Standards evolved both in the Internet sphere and in the Windows computing platform so that a developer could use the best of both worlds. Active Server Pages (ASPs), COM (Component Object Model) and DCOM (Distributed COM), CDO (Collaborative Data Objects), ADO (ActiveX Data Objects), HTML (Hypertext Markup Language), XML (Extended Hypertext Markup Language), and VPIM all evolved to address the diverse needs of developers. Developers get a robust information store and a forward asynchronous messaging system without having to write the entire underlying message handling subsystems.

No LAN is an island anymore when outfitted with a robust messaging and collaboration system. LAN-to-LAN, business-to-business messaging is a reality for large and small businesses alike. We live in a connected world. For many, the decision to be a messaging administrator in a connected world is no longer a career-limiting choice, something that was not always true in the early history of messaging systems. Back then, it was so hard to connect, there was so much incompatibility in messaging products, and maintaining responsible service levels was so illusive that many administrators shied away from choosing a career as a messaging specialist. We’ve come a long way.

**Information, Not Data**

For many, communicating through e-mail is not the same as sitting down and writing a letter. For one, people tend not to put as much care into an e-mail as they would put into a letter (to the dismay of many corporate lawyers). Moreover, trying to converse with more than one person through e-mail is challenging. Ever try to iterate a document in e-mail with 20 people actively contributing? Who has the
most recent version? Although casual one-to-one communication works well in
e-mail, one-to-many interaction requires a different set of techniques. One-to-many
is facilitated by the broadcast capabilities of messaging, but it is still a push method
of communication. Many-to-many requires features similar to that of a bulletin
board, where threaded conversations and pull are more common than push. In the
1990s, many e-mail and messaging systems evolved into collaborative work and
groupware server platforms. What else could your users want? The answer has a
historical parallel. When asked what he wanted during some labor negotiations,
Samuel Gompers, the famous American labor leader, simply said, “More.”
Knowledge-workers also just want more. They want easy and reliable access
from anywhere, anytime.

In the 1980s and early 1990s, asynchronous electronic communication took hold as
the primary medium through which business was conducted. E-mail was every-
where. It was perfect for letting people send their thoughts to others quickly and
reliably. More importantly, the non-real-time nature of e-mail messaging made it eas-
er for people to focus on their tasks without worrying about constant interruption.
(Most of us refer to e-mail as the “In good enough time” system.) For many people,
e-mail became the primary form of communication for efficient work.

The workers of the 1980s and early 1990s became known as knowledge-workers,
who were required to manipulate data coming from diverse sources and in great
quantities. However, as the quantity of data increased exponentially, their inability
to manipulate and disseminate knowledge quickly became the bottleneck of busi-
ness. The problem was exacerbated by geography and the time to travel from one
branch office to another to have a meeting. Being unconnected for hours while on a
plane also made extensive travel difficult. Critical communications would wait until
the knowledge-worker arrived physically or could reconnect to the corporate net-
work for e-mail. Therefore, video conferencing and instant messaging technologies
developed. They took root rapidly in the late 1990s, filling a void that e-mail left.
Knowledge-workers could exchange and access e-mail information with others in
real-time — or close to real-time — from anywhere. Often, they could see the real-
time status of their fellow knowledge-workers. They could tell who was online, who
was busy, and who was away from their desk. Meeting rooms became chat rooms or
video conference rooms. The world became one big office.

Other protocols and standards like Wireless Application Protocol (WAP) and
Telephone API (TAPI) — and new computing devices like BlackBerry, Palm, and
PalmPC — extended the reach of messaging system users. Now, a device the size of
a pager enabled people to collaborate with others and access information using
wireless technology. You could sit down at a browser and access e-mail in HTML
format over HTTP on a server from a different country with ease. Workers in New
York could communicate with workers in Bangkok. Work could be done at any time
of day, either online or offline, and transmitted later.
Today, knowledge-workers have more of the tools they need to access information, and as new modes of communication and advances in hardware are developed, they will have even more. Information access continues to drive the industry’s search for faster, more content-rich environment; it is, in fact, what motivated the development of Windows 2000, Microsoft BackOffice. As part of Microsoft BackOffice, Exchange 2000 is not just an e-mail product, but a communications center, hosting a variety of different modes and means of delivery.

**Summary**

The information industry has evolved from a large, iron, centralized box with proprietary and expensive software to a distributed model using Wintel (Intel and Microsoft) architecture PCs and modestly priced software—from standalone machines to machines connected by the Internet and corporate intranets. Standards have also evolved to the point where many manufacturers support them actively by subscribing to them and by extending them where they are inadequate to satisfy market needs. Exchange 2000 is the synthesis of these trends. Administration in the Exchange 2000 era enables you to build immense boxes or to successfully maintain a more distributed model. Exchange 2000 supports all the crucial standards while ensuring that they are extended where necessary to fit our modern needs. The design of Exchange came from the crucible of evolution and the expressed needs of you and many more knowledge-workers just like you.
The Past and Future of Exchange Server

To understand how to get the most out of Exchange 2000, it helps to understand what problems it was built to solve. Knowing a bit about its history — as well as the demands and challenges facing Administrators with earlier versions or other precursors — will help you when you administer mixed environments and evolve from your current messaging system to Exchange. Exchange 2000 has a rich heritage, evolving over many years into the product it is today. More than any of its predecessors, it relies on the functionality of Windows 2000 Server. It nestles snugly into Microsoft BackOffice, acting as the messaging platform component and handling parts of the collaboration system. Exchange 2000 is well grounded in standards for communication, development, and the Internet. It is bigger, better, and more flexible than its predecessors and provides the best for developers, Administrators, and users. In this chapter, we’ll talk about the lineage of Exchange, especially focusing on differences between Exchange 2000 and Exchange 5.5. To understand where Microsoft is going with Exchange, you need to know a bit more about what Exchange 2000 is today — what flavors it comes in, what features are available in each flavor, and how it fits into BackOffice, Windows Digital interNetwork Application Architecture, and Digital Nervous System architecture.

Studying the Precursor to Exchange — MS PC Mail

Microsoft Mail for Personal Computing Networking (MS PC Mail) is the immediate precursor to Exchange, even though it shares no code base with any version of Exchange. MS PC Mail was born even before Windows NT. Many Administrators considered e-mail the first groupware product, but MS PC
Mail, although advanced for its time, had none of the robust features we have since come to expect in advanced groupware messaging products — threaded conversations, shared replicating folders, client access from the Internet, and robust messaging switch capabilities.

Today’s connected messaging universe started even farther back in time — with the birth of PC networking.

**E-mail capabilities**

When Microsoft entered the networking business in the 1980s, they knew that the most compelling reason people bought networks was to share files and peripherals. However, they also knew that more and more people were buying PC networks for e-mail. Years earlier, manufacturers such as 3Com had sold many a LAN just so the customer could get e-mail. To satisfy the need for e-mail and help drive sales of Microsoft’s LAN Manager product, Microsoft acquired a very successful PC LAN-based e-mail company called Network Courier and reworked its product into the first Microsoft Mail for PC Networking (commonly known as MS PC Mail). MS PC Mail provided companies a PC file-sharing e-mail system. For its time, it had decent gateways (later called connectors in Exchange) for LAN-to-LAN communication. MS PC Mail also had primitive programming interfaces, such as Microsoft Messaging Application Program Interface (MAPI) versions 0 and 1, for third parties to develop on. Several third-party add-ons and several incarnations later, MS PC Mail had gateways to other mail systems (X.400, X.25, FAX, cc:Mail) and for using the Internet as a messaging backbone (SMTP).

One of the critical component contributions of MS PC Mail that is often overlooked was the development of a Messaging Application Programming Interface (MAPI). While there were competing standards (for example, VIM from Lotus), MAPI was the dominant mail API for PC LAN system messaging for many years.

Although MS PC Mail became very successful with the introduction of the Windows client, its underpinning was always a shared-file, single-threaded, 16-bit system with some serious security issues. The 16-bit nature of the system created a limitation for developers. The single-threaded MS PC MAIL MTA was usually run under DOS, and eventually updated to run on OS/2. For the most part, gateways required separate DOS machines. Security issues came from the shared-file system. Users could maliciously or erroneously delete the non-secure post office structure. The gap between what Administrators, regular users, roving users, and remote users required, and what Microsoft developers and Administrators on this platform could accomplish, widened as businesses grew to depend on e-mail.

**Groupware capabilities**

In addition to e-mail, another area of business demand centered on the concepts surrounding groupware. E-mail worked best for one-to-one, non-real-time communication or for broadcast, one-to-many communication. The many-to-many relationship was still a challenge. Providing a place to pull information from was also a
challenge with most e-mail systems. These structural issues, combined with people’s habits of working collaboratively, led to requests for something more than e-mail—for something often referred to as groupware or collaborative knowledge working, which would include such features as threaded conversations and forms.

People who invested in early groupware products (such as Lotus Notes) loved the groupware concepts, but were often dissatisfied by their monolithic nature, proprietary front end, and lack of sophisticated e-mail in these products. Microsoft’s approach was a little different. They saw groupware features as extensions to a robust messaging system integrated with a file system and Internet capability. However, they could not build that capability into MS PC Mail, because MS PC Mail was built on top of operating systems that lacked the necessary sophistication for the task.

**Windows NT**

MS PC Mail suffered from being bound originally to DOS and later to OS/2. DOS and OS/2 lacked the ability to scale as fast as users demanded. Both had very limited subsystems to build on, and stumbled on 32-bit horsepower because they were only 16-bit operating systems. Both contained the notions of a directory, but that notion was pretty limited and primitive. Microsoft built Windows NT from the ground up to be a 32-bit OS with features that messaging and other application servers needed:

- Centralized management and administration
- Security
- Scalability
- Robust multi-threading and multi-tasking architecture
- Full-featured network capabilities

Before Windows NT, nothing in the Windows world could support a sophisticated e-mail, messaging, and workgroup collaboration system. Third-party e-mail and groupware products quickly moved to the NT platform. Microsoft built BackOffice (including Exchange) on top of Windows NT, thus using the Windows NT security systems to give unified security to BackOffice applications.

**Microsoft Office and other BackOffice applications**

Users, Administrators, and developers often require Internet access, database access, host connectivity, and network management to be closely coupled with e-mail to deliver the business utility they desire. In most organizations, data is scattered—stored in different data repositories, in different geographical locations, and on different systems, all of which developers and users need to be able to access with ease. Microsoft built two suites of applications to address this need. One was the server back end, BackOffice. It included the base operating system (Windows NT); database (SQL Server); host connectivity (SNA Server); Internet
Web, gopher, and FTP services (IIS); and software distribution (SMS). It also included a messaging and collaborative work platform (Exchange). The other suite, Microsoft Office, was built for the front end of the client-server model. It included Word, Excel, PowerPoint, Access, and more. For e-mail access, Microsoft Office included Outlook. For lighter e-mail access, they included Outlook Express. Together, the back end and front end facilitated usage and development of robust information systems.

For example, consider a typical business Web site with a form for questions and comments. Being able to build a system that takes data from the form, stores it in a SQL database, generates a report from the data every week, and e-mails the report automatically to a mailing list of roving users has a great deal of value. Being able to use one set of development and administration tools for such a system also has a great deal of value.

Internet Information Server, or IIS, became a cornerstone platform to build applications. SQL Server, Index Server, and Transaction Server became the back-end database applications that stored, manipulated, and processed hierarchical and relational data. Data could include objects such as company internal reports, project documentation, live status briefs, marketing data, or even usage statistics on the Web site. While data could also take the form of e-mail objects, their characteristics were not best suited to the design of SQL server. Because of that limitation, Microsoft built the original Exchange with its own Information Store database.

**Earlier versions of Exchange**

In geology, when two layers of rock that should not normally be next to each other are juxtaposed, it’s called an *unconformity*: layers that should be in between are missing. In a sense, when you layer MS PC Mail next to Exchange, it feels like an unconformity. Clearly they come from different eras, but those eras seem very far apart and discontinuous. Although Microsoft released no messaging products between MS Mail and Exchange, developments in NT and BackOffice affected the way Microsoft thought about messaging and collaboration engines.

Exchange Server introduced new notions, features, and components. Exchange Server 4.0 through 5.5 relied heavily upon Windows NT for security and network connectivity. A separate Exchange administrative program provided all administrative features for managing Exchange, and interplayed with NT administrative programs like User Manager and Event Log Viewer. Private and Public Information Stores were the data structures. The Private Information Store held individual mailboxes; the Public Information Store held public folders for collaborative work and groupware capabilities. Instead of storing e-mail in a flat file system, Exchange stored it in a database based on their JET (Joint Engine Technology) technology. An enhancement to NT Backup allowed it to back up the information stores even when they were open and Exchange was running. Exchange provided a series of
connectors (in MS PC Mail, called gateways) to other popular mail systems and the Internet. Multiple transport standards were supported, both for the connectors and for access from various clients. Connector protocols included X.25, SMTP, and X.400. Client protocols included POP3, IMAP4, and HTML. Other transport and Internet protocols that were supported included RPC (Remote Procedure Call), NNTP, and HTTP. Clients that were supported evolved from the original Exchange client to Outlook and included any client that supported the standards POP3 and IMAP4.

The addition of a directory in early versions of Exchange was a big change from MS PC Mail. This X.500-like directory was self-propagating and had the concept of back fill and other self-healing capabilities built into it. Versions 5.0 and 5.5 introduced the ability to make LDAP calls to the directory, further extending its accessibility to standards-compliant third-party systems or internal development.

On the development side, Exchange introduced forms and form applications. The forms were extendable using VB. APIs for calls to the information store service were published. Similarly, client APIs and other server APIs were published so that companies could build add-on applications such as voice mail, wireless messaging, fax, and so on. ASPs (Active Server Pages) probably first appeared in a production application in Exchange to deliver access to mailboxes hosted in Exchange to users coming in with a browser and using HTML.

Exchange also introduced robust public folders for groupware capabilities. Coupled with e-mail, Exchange forms, and powerful APIs, developers had a mighty set of tools from which to weave their application magic. Developers could now automate processes and ticket-based systems with Exchange and Outlook.

Using Exchange, Administrators could administrate large, organization-wide messaging systems centrally. With the help of sophisticated integrators, numerous multinational Exchange environments that supported thousands of users were built. Administrators could take advantage of the integration between NT, BackOffice, and Exchange. For example, they could build one set of accounts for network access or use SMS to roll out clients. Sometimes they used NT cluster capabilities to gain fault tolerance. Using these techniques and products as well as many others, they built large, highly manageable messaging infrastructures on Exchange. Yet they were not the only ones to embrace Exchange. Administrators in even the smallest environments loved the new administrative features found in Exchange, and adopted it in small and medium-size companies.

**Upgrading from Exchange 5.5 to Exchange 2000**

There are major differences and vast improvements between Exchange 5.5 and Exchange 2000. For example, don’t look for admin.exe, as shown in Figure 2-1, in Exchange 2000.
Figure 2-1: In earlier versions of Exchange, admin.exe brought you to this familiar screen, where you could do almost all of the administration.

That role is now taken over by Windows 2000 Microsoft Management Console (MMC) framework. Exchange 2000 provides the snap-ins to the MMC for more granular and integrated administration, as shown in Figure 2-2. The MMC also handles other functions, such as managing address lists, folders, message tracking, conferencing, and chat services.

Figure 2-2: In Exchange 2000, you can use snap-ins to do management through the Windows 2000 MMC.
The Exchange System Manager console, as shown in Figure 2-3, is used for server-centric activities such as managing connectors. It resembles Admin.exe in version 5.5.

Resources for development, including administrative tools, are plentiful in Exchange 2000. It includes Collaboration Data Objects (CDO) and Collaborative Data Objects for Exchange Management (CDOEXM), Windows Management Instrumentation (WMI) to monitor data, core providers to access the Event Logs, and Exchange providers to access other items, such as connector status.

![Figure 2-3: In Exchange 2000, management not done in the MMC often takes place in the Exchange Management Console (which resembles the old Exchange Admin.exe).](image-url)

The new administrative model provides quite a bit more flexibility than the earlier administrative model. Previously, connectivity between Exchange servers defined sites. Sites defined both message routing and administrative models; this connectivity often forced certain administrative models that may not have been right for your company. With Exchange 2000, message routing and administrative models are decoupled. Physical network topology determines message routing, but you can use the organizational model for your company as the administrative model for Exchange.

Active Directory Service (ADS) is the cornerstone of Windows 2000, and Exchange 2000 takes full advantage of it. Although some of the features you have gotten used to with the Exchange Directory are also found in ADS (like back fill), ADS has much more flexibility, with a schema that allows attributes on all objects. You can now search for objects by attribute.
ADS is organized by Organizational Units (OUs), domains, trees, and forests. Sites and subnets that cross administrative boundaries are possible; you can build topologies that reflect the way your company works. However, definition of the term *site* has changed from what you may remember from Exchange 4.x/5.x. An Exchange 5.5 site is equivalent to an Exchange 2000 Routing and Administrative Group, not to an Exchange 2000 site. In Windows 2000 AADS and Exchange 2000, a site is simply a collection of IP subnets connected with expected high-speed (LAN-level) and highly reliable network.

Exchange 2000 is different from Exchange 4.x/5.x in other ways. The Exchange 5.5 GAL (Global Address List) is now the Exchange 2000 ADS Catalog. Exchange 2000 looks to Windows 2000 for security on objects, rather than using the three security contexts of Exchange 4.x/5.x. Because ADS is part of Windows 2000, Windows 2000, rather than Exchange, now manages replication and repair of the directory.

Installing Exchange 2000 or running a ForestPrep makes changes to ADS, adding to the ADS schema all the objects required for messaging. Subsequently, the System Attendant and other services and users make calls to ADS—to add, change, or request directory, User, Group, and Contacts information. All of this information is now managed by ADS. Exchange-specific attributes for each look like just another property sheet for each object. While objects still have a DN (distinguished name) as in Exchange 5.5, ADS uses Global Unique Identifiers (GUIDs) as the key. Because of that, you can now move users around containers, which you could not do easily in earlier versions of Exchange. Users and mailboxes are now administered through a snap-in for the ADS called Active Directory Users and Computers.

Utilities for compressing or repairing information stores are mostly preserved. ESEUtil has been moved back to \BIN. ADSIEDIT/LDP replaces raw mode edit of the schema. NTDSUTIL is used for low-level ADS management. MTACheck is retained. ISINTEG is built into the store, but the exe is still there.

Storage changes significantly in Exchange 2000. The Information Stores you used are retained. A new store is added: the Web Store, a hybrid database where everything can be accessed by a URL. It is the right repository for managing unstructured data of many different types in one infrastructure. Further, the Web Store stores the content in its native form rather than wrapping everything in a MAPI wrapper, as previous versions of Exchange did. This allows astute users to gain speed of access.

Exchange 2000 has also enhanced data storage. Support for voice mail, sound bytes, and video clips are all improved. Full text search is integrated for increased performance in searches, and the search capability has its own API. The enhanced search engine also searches attachments, such as Microsoft Word and Microsoft PowerPoint.

While Exchange 2000 retains aspects of the old Information Store model, it is significantly expanded and enhanced with significant amounts of new code. Multiple databases are allowed, where previously only one Public and one Private were...
allowed on each server. With the concept of a Storage Group and split databases, you can do concurrent restores of different storage groups on the same server; you are also allowed to have multiple public folder hierarchies. In addition, Exchange 2000 is set up to handle streaming media.

Exchange 2000 still contains a Message Transfer Agent (MTA), but its role has changed a bit. In previous versions of Exchange, sites did almost all of the moving of message objects between information stores and connectors. Now sites are used for pure X.400 connections only; the MTA no longer performs the routing. Instead, the new SMTP Link State Routing Engine performs the default routing between Exchange 2000 servers.

The primary messaging protocol for Exchange 2000 is SMTP. Responsibility for that protocol is now delegated to Windows 2000 (in earlier versions of Exchange, it was X.400). When Exchange 2000 is added to a Windows 2000 environment, SMTP is extended with advanced features such as message tracking, intelligent routing, and queuing. In a sense, it becomes a peer of X.400. In Exchange 2000, each server can host multiple SMTP virtual servers, each afforded its own configuration. By taking advantage of the extensibility built into the SMTP server, developers can customize events and create programs that can, say, add a disclaimer or copyright information to every message.

Because the primary protocol of Exchange 2000 is SMTP, Exchange is even more tightly integrated to IIS than before. Distributed Authoring and Versioning (WebDAV) is available in Exchange 2000 with IIS; you can now use the Web as an extension of your messaging system with ease. Enhancements to Exchange 2000 and to IIS in Windows 2000 make it easier to secure your network from Internet attacks. Controlling the use of your system as an SMTP router is easier than ever before.

Some users want instant communication. Microsoft supplied Instant Messaging and Chat Service in Exchange 2000 to address that need. Common features, such as presence subscriptions, are incorporated into the Instant Messaging. Applications can be written to do intelligent routing based on presence. Multiple chat rooms and multiple chat communities can be supported.

Development takes a great leap forward with Exchange 2000. If you did development in Exchange 5.5 using CDO version 1.2.1 or Outlook Web Access, you should be able to use your applications without modification. Exchange 2000 allows programmers to trigger on events in the Information Store, such as adding an e-mail message. You are allowed to register your event sinks to take action when a certain event takes place. In an incredibly powerful capability, event sinks are also allowed for both SMTP and NNTP protocol events. And in Outlook Web Access, you can move away from the default forms that Outlook displays and substitute custom forms using Active Server Pages.

While clustering is not new, in Windows 2000 it improves radically. Exchange 2000 Enterprise Server users will probably want to take advantage of it.
The great news about Exchange 2000 is that for the most part, you do not have to upgrade Exchange 5.5 clients immediately, although you may wish to gain the certain advantages of advanced clients with Exchange 2000. (Some earlier clients may also work.) If you are accessing Exchange with a Web client, you will want to upgrade your browser—if you are using a Microsoft Internet Explorer, to IE 5.0.

For multinational and international users, Exchange 2000 provides multilingual administration, multilingual clients, localized address books, and browser support in 22 languages. While these are not features of Exchange 2000 only, many Administrators struggled with them in earlier versions of the Microsoft platform. Now the problems melt away like snowflakes on your bare hand.

The differences between Exchange 4.x/5.x and Exchange 2000 are significant. However, your previous investment in Exchange knowledge will help you with Exchange 2000. Many customers will be running older versions at the same time that they are running Exchange 2000. We explain all the new concepts in Exchange 2000, as well as many of the co-existence issues, later in this book.

If you find the information about earlier versions of Exchange to be mystifying, consult Exchange Server 5.5 Secrets. It contains much of the material one author of this book taught (and learned) as a Microsoft PSS-approved TEP instructor, training people (including Microsoft employees) on Exchange 4.x and 5.0.

Overview of Exchange 2000


Getting the wrong version can make for a stressful day at the office. One of our customers called one day to tell us, “We just about hit the 16GB limit on our Exchange server.” We said the 16GB limit was removed in the 5.5 Enterprise Edition. The reply: “We thought we were running Enterprise Server Edition, but found out this morning that we were running just plain Server Edition.” It also happened that the customer still had Circular Logging checked, and was just about out of physical space on the box. For this customer, the box was the hub of the universe. Things could have been worse— but only if the halon system in the computer room had suddenly gone off.

The differences between the two Exchange versions are illustrated in Table 2-1.
Table 2-1
Selected Differences Between Exchange 5.5 and Exchange 2000

<table>
<thead>
<tr>
<th>Feature</th>
<th>Exchange 2000</th>
<th>Exchange 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enterprise Edition</td>
<td>Server Edition</td>
</tr>
<tr>
<td>Information Store Storage Capacity</td>
<td>Unlimited</td>
<td>Maximum Size 16GB</td>
</tr>
<tr>
<td>Multiple Database Support</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Connectivity to Other E-mail Systems</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>X.400 Connector</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Microsoft Outlook 2000 Client</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Development Kit</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Instant Messaging</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Chat Service</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Front-End &amp; Back-End Architecture</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Licensing Maintenance</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Windows Clustering Support</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

In later chapters, we review the differences listed in Table 2-1 and analyze how the choice of version influences the way in which you administer and maintain your Exchange environment. If an explanation of Exchange 2000 does not specify a version, it normally refers to the Enterprise Server Edition. In cases where there is a pertinent difference between the two versions, we explain the difference.

Exchange 2000 Server requires that you run one of the server operating systems of Windows 2000: Standard Edition, Advanced Edition, or Datacenter Server. Your choice of operating system depends on the specific features of Windows 2000 you require to support your messaging implementation. Decisions affecting your choice of operating system include whether to use two-node or four-node clustering, Network Load Balance (NLB), multiprocessor machines with more than four processors, and large memory configurations (more than 4GB). Each of these Windows 2000 editions has its own Hardware Compatibility List.

Active Directory Service

No other aspect of Windows 2000 captures the attention of messaging Administrators the way the move to Active Directory Services does. Exchange 2000 no longer supplies its own Directory Service, as was the case with Exchange 4.x/5.x. In that case, you had two directories, one for NT and one for Exchange. Other BackOffice products also had their own directory-like components. Exchange 2000, on the other
hand, fully utilizes the Windows 2000 Server Active Directory Service (ADS), as do other applications. Since the Exchange 2000 directory structure is an extension of the schema of Windows 2000 Active Directory, there is a single directory replication — and it is handled by ADS, not Exchange. All BackOffice products, and many third-party applications, also take advantage of ADS. ADS looks and feels like the old Exchange Directory Service on steroids. Many aspects of it will feel quite familiar to anyone who really knows Exchange, as you will see in later chapters about Directory.

If you expect to deploy Exchange 2000 and you are in a large environment with a pretty big directory, consider expanding your ADS schema with all the objects that Exchange needs before you propagate the directory. Remember, all schema changes cause a complete replication of the entire directory. Expanding the schema before you populate the directory in a large environment will ensure that you do not overwhelm your WAN with traffic.

Some of the advantages of Exchange 2000 using the Windows 2000 Active Directory are

- **Centralization:** Central administration in Windows 2000, Exchange, and other BackOffice products, including reduction of duplicate security models and a unified login for both network and Exchange.
- **Stability:** The inherent flexibility and scalability of Active Directory removes practical limits. In some cases, you may want to add Microsoft Meta Directory Services above ADS, NDS, Notes, and all the other directories, but it is doubtful you will ever need to abandon ADS.
- **Permissions:** Because each object in Active Directory stores the permissions information with the object, Exchange and other applications access the permissions when they access the objects, eliminating the need to search the directory separately to check permissions or to search other applications for permissions.

**Administration of Exchange**

Exchange has always been easier than competing products to administrate. Exchange 2000, coupled with Windows 2000, takes that ease of administration to an entirely new level.

**Microsoft Management Console**

The majority of the administrative tasks for Exchange 2000 have been moved into the Microsoft Management Console (MMC), where they are integrated with other networking tasks. Snap-ins for address lists, folders, message tracking, conferencing, and chat services are included. Users and Groups (formerly Distribution Lists)
are now property pages in Windows 2000 Users and Groups, managed through the MMC. The MMC manages any item (mostly the user-centric components) whose attributes are contained in Exchange’s extension of ADS.

Managing the rest

The Exchange Server Management Console manages server-centric components. EDO and CDO are used for Exchange management. Exchange 2000 includes a queue viewer and API (Application Programming Interface) for MTAs (Message Transfer Agents, X.400, and SMTP) and for the connectors.

New administrative model

Exchange 2000 eliminates many of the administrative problems caused by the physical and topological limitations of earlier versions of Exchange. The physical relationship among your various Exchange clients, Exchange 2000 servers, the outside world, and any other application servers connected to the Exchange network can now be mapped with few limitations. Yet the physical topology no longer dictates the logical administrative topology; your logical and administrative Exchange environment is now as different from the physical as you desire.

The physical network topology generally determines routing of messages and data. More often than not, this differs for the desired logical organization of Exchange 2000. Generally, you want your administrative model to be determined by your often-changing organizational requirements. You want the administrative topology to reflect the way you do business. The separation of the physical and logical designs makes it easy to create an administrative model that works for you, and allows you to maintain it in the face of administrative changes that occur regularly in most organizations. For example, changing a business unit’s location, altering personnel within a unit, or changing administrative roles from one group to another do not require physical topology changes.

Caution

The freedom to build different topologies applies only to organizations running Exchange 2000 in native mode — in which all Exchange servers in the organization have been upgraded to Exchange 2000. If the servers have not all been upgraded, the network operates in hybrid mode, and the older restrictions still tie physical and logical models together. Similarly, in topologies where Exchange 2000 coexists with other Groupware or messaging, flexibility may be limited.

Users and mailboxes

The addition of users and mailboxes is now integrated with the creation of network accounts in the Windows 2000 Active Directory. This integration reduces the administrative overhead of adding, changing, moving, or deleting user accounts. It also makes it easier to move mailboxes around.
Connectors and Migration Tools

Exchange 2000 includes numerous connectors and migration tools to connect to other types of messaging and collaboration products across a number of transport protocols. Exchange 2000 includes connectors and/or migration tools for SMTP, X.400, Lotus cc:Mail, Lotus Notes, Novell GroupWise, Microsoft Mail, and Schedule Plus Free/Busy. If a connector you are looking for does not exist in Exchange 2000, you may be able to use an Exchange 5.5 connector in mixed mode or a third-party gateway. The connectors are used over LAN, X.25, and asynchronous connections.

Information Store

The Information Store of Exchange 4.x/5.x has changed significantly in Exchange 2000. In its new form, it is more scalable, more reliable, faster, and more flexible, and it has parts you did not see in earlier versions of Exchange.

Web Store

Much more than an e-mail data store, the Web Store is a general object-based data repository for Microsoft products. It provides for storage of expected objects, such as documents, e-mail, and streaming data; it also provides for custom objects, and can therefore present those objects to a broad spectrum of client access applications, such as Outlook, Browser, and Microsoft Office applications. It is crucial to take advantage of the Web Store in Exchange 2000 deployments.

For years, students and customers have asked for architectural advice on storing their data: What model is best? The operating system file system? A database like SQL server? The Exchange Information Store? For many types of information, the answer now is, unequivocally, the Web Store.

The Web Store will become a major information store of the Internet-oriented, computational environment that Microsoft is developing around BackOffice. All items in the Web Storage System can be accessed via a URL address. The Web Store supports both structured and semi-structured data — including documents, contacts, messages, reports, HTML files, and ASP pages — in a hierarchical folder system. It provides full context indexing and search capabilities (including most attachments). Figure 2-4 provides a macroscopic view of the relationship between relational-database storage (Microsoft SQL Server, Oracle, and so on) and the Web Store.
Multiple databases per server (Enterprise Edition only)

Exchange 2000 Enterprise Edition introduces the notion of multiple databases in a storage group and multiple storage groups. Previous versions of Exchange were limited to one private and one public database; while Exchange 5.5 Enterprise Edition removed the 16GB limit to the database, the practical limitations imposed by the need to back up (and, if necessary, restore) a database in a reasonable period of time limited the size of the Information Store.

Exchange 2000 Enterprise Server removes this practical limit: You can have multiple databases, back them up simultaneously, and restore them individually.

![Microsoft Knowledge Management Storage Strategy](image)

**Figure 2-4: Relational Database versus Web Store**

**Workflow Capabilities**

To take advantage of Exchange 2000, you should immerse yourself in Collaboration Data Objects (CDO).

Exchange 2000 CDO libraries include support for messages, appointments, contacts, workflow, folders, and mailboxes. The Workflow Engine and Workflow Designer applications are coupled to the Web Store. Workflow Engine provides routing, state management, audit tracking and other standard tasks common to many workflow applications. These capabilities significantly expand the ability of Exchange to provide an infrastructure for Workflow applications based on Microsoft Office 2000, a Custom UI (such as VB, C, or C++), or a hybrid of these. An example of this is shown in Figure 2-5.
Real-Time Communication

Exchange has always been noted for its store-and-forward, just-in-time messaging and collaboration systems. Exchange 2000 expands the systems with a number of real-time communication options.

Instant Messaging

Instant Messaging in Exchange 2000 provides more than just AOL-style Instant Messaging between users. It can be used to notify users, Administrators, or applications of events that require time-critical attention. By using the presence database to determine who is online, workflow applications can make more intelligent routing decisions.

Chat (Enterprise Edition only)

Exchange 2000 Enterprise Edition includes a Chat engine, which can host multiple chat communities. Many ISPs will be moving to Exchange and taking advantage of the Instant Messaging and Chat capabilities, as well as messaging and Groupware.

Evolving Key Standards

Exchange 2000 supports more standards than any other messaging and collaboration server that is popular today. Many of these standards are Internet standards, while others are older messaging and collaboration standards.
Internet standards


Microsoft continues to do an admirable job of embracing the standards that matter to consumers and corporate customers, while not being trapped in a standard that limits business utility unreasonably. For example, the X.500 standard is a bit anachronistic. If all Microsoft gave us in ADS was X.500, we would all be disappointed and limited. Instead, they took the best features of X.500 and combined them and extended them with other potent models, such as the Internet’s DNS, to give us the power and flexibility we demand.

Microsoft continues to support Internet standards and to work with industry to develop new standards and enrich them with Exchange 2000.

Data access standards

Previous versions of Exchange supported Mail Application Programming Interface (MAPI) and Exchange Data Objects (EDO). Exchange 2000 supports MAPI, EDO, Collaboration Data Objects (CDO), XML, LDAP3, ActiveX Data Objects (ADO), OLE DB, HTTP (enhanced through the WebDAV specification www.ics.uci.edu/pub/ietf/webdav), and WIN32 for data access.

VPIM

For access from phone devices, Exchange 2000 accepts Internet Voice Messaging Protocol (VPIM) messages and stores them in the Exchange 2000 multimedia format. Before messages are sent to an external voice mail system, they can be converted back to VPIM format. This allows access via telephone to voice mail messages stored in Exchange 2000 and third-party software to provide unified messaging solutions by checking all messages via telephone.

SMTP Virtual Server

Exchange 2000’s primary and default protocol is Simple Mail Transfer Protocol (SMTP). It is both the native protocol for mail submission and the native transport protocol. This protocol facilitates the integration of Exchange 2000 with the Internet.
Clustered (Enterprise Edition only)

Exchange 2000 Enterprise Edition provides for four-way active/active clustering. With clustering, as many as four servers can be active at any time with fail-over to the remaining servers if one active server goes offline.

When you are configuring a cluster environment, if the requirement is to have no diminished processing capability in case of a failure, remember to keep the utilization on all servers prior to failure to the limit determined by the following formula (assuming all servers in the cluster have the same processing power; \( L = \) load per server (%) and \( N = \) number of servers): \[
\frac{(L \times N)}{(N - 1)} \leq 100\%
\]

Internationalization of Exchange 2000

Exchange has become the messaging system for many multinational or international companies. Exchange 2000, coupled with Windows 2000, delivers a number of features that make administration in a global environment easier, including:

- Multilingual administration
- Multilingual clients
- Localized address book
- Browser access using Outlook Web Access in 72 languages

Exchange 2000 Conferencing Server

Exchange Conferencing Server is sold as a separate server product but in a sense is a part of Exchange. Client Access Licenses (CALs) are the same as those for Exchange 2000. Thus, the same CALs used to access Exchange 2000 grant you license rights to access Exchange Conferencing Server. Exchange Conferencing Server, a component server of the Exchange 2000 family, was separated in order to increase performance and allow more flexibility in design. If you do not intend to run it, you shouldn’t load it because of the additional burden it adds.

Exchange Conferencing Server has three primary components:

- Conference Management Service: The gatekeeper for Exchange Conference Server. It provides access to scheduling, conferences, and the various technology providers such as Microsoft or other third parties; it also manages resources. The Conference Management Service allows third-party providers to supply content for Exchange Conferencing Server in an integrated format.
Data Conferencing Provider: A technology provider for Conference Management Service. It is based on the T.120 International Telecommunications Union—Telecommunication Standardization Sector (ITU-T) Recommendation for Multimedia Conferencing. Data Conferencing Provider utilizes various forms of data, including multiparty application sharing, chat, file transfer, creating a data, and free-hand drawings to provide real-time conferences. It is optimized to significantly reduce network traffic.

Video Conferencing Provider: Provides multiparty video and audio conferences over multicast-enabled IP networks. Video Conferencing Provider allows conference participants to see and hear each other simultaneously, eliminating the need for a parallel teleconference. The multicast-enabled network optimizes network bandwidth, while distributing audio and video communications to conference participants.

Exchange Conferencing Server allows third-party media providers to interface directly with Conferencing Server and provide content for conferences; the audio/video conferencing is IP-multicast standard, and the data conferencing uses the T.120 standard protocols. This allows the addition of new media formats as they are developed.

Conferencing Server can combine data, voice, still images, and video in a conference. It is a true server application and does not depend on the persistence of any user, including the originator, in the conference. It is integrated with Exchange’s scheduling capabilities.

Exchange Conferencing Server provides for three levels of security: public, public with password, and private. The security level is determined at the time the conference is scheduled. In a private conference, authentication can be required before a user accesses the conference. Exchange Conferencing Server allows the hosting of secure conferences with individuals on the other side of a firewall.

Exchange 2000, BackOffice, and Windows DNA

Exchange 2000 is not just the next version in the Exchange Server line. Nor is Exchange 2000 just a messaging server. Exchange 2000 is a part of a much greater whole.

Exchange 2000 is another piece of the Windows DNA puzzle. It is a building block in the platform for building solutions for businesses—from e-commerce sites to enterprise supply-chain integration. Windows DNA doesn’t stop at the Internet; it allows you to build solutions that transcend Web sites and exchange information and data with devices connected to the Internet, and to build them quickly with pre-existing components.
At the time this book was written, the other members of Windows DNA, as shown in Figure 2-6, were Application Center 2000, BizTalk Server 2000, Commerce Server 2000, Host Integration Server 2000, Internet Security and Acceleration Server 2000, SQL Server 2000, Visual Studio 6.0, and the COM+ component and programming model. The architecture is dynamic. Existing components will continue to evolve while others will be added.

Exchange 2000 brings messaging and collaboration capabilities to the Windows DNA architecture. Not only does it offer a messaging server as in previous versions, it also has more extensible groupware, video/data conferencing, NetMeeting, and Instant Messaging built in. It encompasses the totality of communication and data-sharing. Developers can use it to provide their applications and systems with real-time data messaging. Exchange 2000 is a critical component of the solution platform, built on modular components and standards that Microsoft BackOffice is built to be.
Summary

Exchange 4.x and 5.x were Windows NT applications built to address design requirements and features not available in MS PC Mail and other earlier products. The result was a highly featured, scalable messaging and collaboration platform tied to other BackOffice products. Exchange 2000 provides even greater enhancements for end-user functionality, scalability, administration, collaboration, connectivity, and Internet platform.

Exchange 2000, coupled with Windows 2000, builds on that tradition, introducing features and enhancing components necessary to build sophisticated global messaging and collaboration systems. End-users can now connect from remote sites more easily and take full advantage of the intersection of messaging and collaboration with the Internet. They can use collaboration features and real-time communication features to enhance workgroup productivity. Yet, even with all the new features, your job as an Administrator—charged with delivering functionality in a stable way—remains secure. Your organization can set up Exchange to reflect boundaries in its organizational structure while taking advantage of its physical WAN. Scalability features allow you to build servers that equal mini-computers and mainframes in performance while maintaining the cost profile of PCs. In the future, one of the major components of a world of knowledge-workers connected by the Internet will be Exchange 2000—a supportable, cost-effective, multifaceted platform to use and build on.
References to protocols and messaging standards are everywhere in the world of Exchange. Understanding them becomes even more important in Exchange 2000 because Exchange 2000 is even more standards-based than its predecessors. In Exchange 2000, Simplified Mail Transport Protocol (SMTP) becomes the default routing protocol — earlier versions of Exchange used X.400. Even with the switch, both protocols remain important in most Exchange environments. Between the two of them, they will control most communication within your Exchange organization and to other messaging systems. In any book about a global messaging system, the X.25 protocol will come up. Even with the proliferation of TCP/IP and the Internet, it is highly likely that global designs will find that they require X.25 or X.75 protocols in some countries, at least for a while. When you configure the Exchange X.400 connector, you have to choose whether to use TCP/IP, TP4 (like TCP/IP), or TP0 (X.25) for its transport. TCP/IP is the most important network protocol in use today. This chapter covers the application layer protocols that rely upon TCP and IP for delivery and transport. It also covers application layer protocols that are important to Exchange.

Exchange 2000 transport protocols propel messages across message handling systems, but the messages need a valid e-mail address. How does Exchange help you find those valid e-mail addresses, and what service provides help for you to organize and manage them? This is one of the jobs of the directory. This chapter will expose you to some of the lexicon and concepts that are used in X.500 Directory Standard and the Lightweight Directory Access Protocol (LDAP).
CCITT X.400 Standard

The treaty organization known as Comite Consulatif International Telegraphique et Telephonic (CCITT) developed the X.400 standard. The same committee was previously known in the United States as the International Telegraph and Telephone Consultative Committee (ITT) and is now known everywhere as the International Telecommunication Union (ITU). The committee based the X.400 standard on the Open System Interconnection (OSI) layered reference model and protocols. X.400 is one of the protocols that make global electronic communication between different messaging organizations and environments successful. Another important attribute is that from its earliest conception, the X.400 standard was meant to be hardware- and software-independent.

After many years of work, the X.400 standard was published in 1984. Since then, it has been iterated in four-year increments by study groups. Each period was also referenced by a color.

- The Red Book, published in 1984
- The Blue Book, published in 1988

X.400 faced the same problem that many standards face: the industry’s need to satisfy the demand of the market surge ahead of the agreements embodied by the standard as they get debated, argued, and eventually arbitrated to a consensus. Because of the delay, many manufacturers who want to be compliant with the standard find themselves extending the standard to add functionality to their systems. The result is that issues not covered in the standard suddenly have a series of proprietary solutions that only work on a few products. Often, after a short period of incompatibility between the proprietary extensions of the standard, the standard’s body will address the very issue that caused the extensions. It is not uncommon for customers to adopt a solution implemented by one of the manufacturers or to adopt a hybrid of the solutions produced by a number of manufacturers. Another likely result of the time lag between implementation of proprietary extensions and their absorption into the standard is a delay in compliance and adoption of the released version of the standard. Some vendors wait for resolution of the proprietary extension issues and for specific features to be absorbed into the next release of the standard. When manufacturers wait to build compliant products, the adoption momentum of the standard slows down, sometimes weakening the standard. You should expect latency between the release of any version of the X.400 standard and its implementation in any messaging system claiming to deliver a compliant system. You should also expect vendors to extend the standard between releases with somewhat proprietary extensions.

OSI, X.400, and SMTP

The Open System Interface (OSI) reference model has seven layers, which define communication between applications and the hardware used for that
communication. From top to bottom, the seven layers are application, presentation, session, transport, network, data link, and physical. X.400 and SMTP define messaging specifications that sit at the application layer (layer seven) of the OSI model and rely on the layers below.

The OSI protocol stack is shown in Figure 3-1.

![Figure 3-1: An OSI protocol stack and some corresponding protocols](image)

The bottom four layers of the OSI model handle network interconnection—the way computers communicate with each other over the physical wires and how that communication is routed between networks. The top three layers handle the way the applications and the operating system interact. Thorough, rigorous definitions and specifications delineate how an application at one layer passes information to the next. Because of that the OSI model makes it possible to substitute compliant products and components on one layer without affecting the adjacent or nonadjacent layers. For example, changing your network from Token Ring to Ethernet should have no effect on a higher-level application.

For a broader frame of reference, we produced Figure 3-2. It shows how NT 4, NetWare, and Apple correlate to the OSI model.

Tip

Understanding the OSI model and how protocols relate to it helps you when troubleshooting protocols like X.400. It can speed your ability to zero in on the right part of the problem or even help you discern that the problem is not X.400 at all, but some lower-layer issue.
Figure 3-2: OSI and other familiar network communications protocols
X.400 components

Components worthy of your focus in an X.400 messaging system include:

✦ **User Agent:** The UAs (User Agents) are the processes that act for the user. UAs are immediately visible to the users. Outlook 2000 is an example of a UA.

✦ **Message Transfer Agent:** The role of the MTA (Message Transfer Agent) is to move messages between users and other MTAs. An example in Exchange is the Exchange 2000 MTA Service, which moves messages to and from the connector queues. The MTS (Message Transfer System) is a group of message transfer agents.

✦ **Message Store:** The MS (Message Store) is the definition of the protocols for a data structure to send and receive messages from the UAs and the MTAs. In Exchange this is the Information Store and Information Store Service.

✦ **Message Handling Environment:** An assembly of message handling system components and users is called an MHE (Message Handling Environment). An MHE may include multiple connectors, gateways, MTAs, UAs, and MTSs. It is also possible for the MHE to be a single messaging system for a group of local users (for example, Microsoft Small Business Server). An example of a more robust X.400 MHE is shown in Figure 3-3, where each MHS might be an Exchange 2000 organization.

![Figure 3-3: A Message Handling Environment](image-url)
Distribution Lists: Some terms correlate to different names in different products. X.400 DLs (Distribution Lists) were referred to as Groupnames in MS PC Mail. In Exchange 4.x and 5.x, they became Distribution Lists (DLs). In Exchange 2000, Microsoft uses Windows 2000 Active Directory Service to manage groups, and those groups can also act like Distribution Lists.

Directory: Many people remain confused about the relationship between X.400 and the directory. The directory was implemented in 1988 as an optional feature but is still considered an X.400 object.

Access Units: AUs (Access Units) are synonymous with gateways and connectors. An AU is the component you use to link different communications systems like FAX, the U.S. Postal Service, PROFs, cc:Mail, or Telex. Sometimes the AU will translate from one communications system to another, as well as link them.

Interpersonal Messaging System: IPMS (Interpersonal Messaging System) is the system of MTSs that the user communicates with to transmit a message to other users.

P1 Envelope: In the lexicon of messaging, you will often hear people mention the P1 Envelope, which is a reference to the X.400 specification. The more commonly used P designations are delineated in Table 3-1.

<table>
<thead>
<tr>
<th>X.400 Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>(envelope) MTA to MTA communications</td>
</tr>
<tr>
<td>P2</td>
<td>(body) 1984 Message Content, UA to UA communications</td>
</tr>
<tr>
<td>P7</td>
<td>Client-MS communication</td>
</tr>
<tr>
<td>P22</td>
<td>1988 Message Content</td>
</tr>
</tbody>
</table>

X.500 relationship to X.400 and Windows 2000 Active Directory Service

X.500 is a standard that covers the internal object structure of a mail system and the rules a directory object must follow when interacting with that internal structure (you can think of it as a database and the rules that govern placing, retrieving and using objects in the database). Like X.400, X.500 is now managed by the ITU and was released every four years. X.400 calls for a directory object, but does not call specifically for X.500. While X.500 is not a mandatory part of the X.400 specification, it would be rare to find an X.400-compliant system that did not include
support for an X.500-like directory. Exchange 2000 takes advantage of the Windows 2000 Active Directory Service (ADS). ADS is both X.500-like and supports DNS (Domain Name Service). The result is that you have the advantages of both a hierarchical X.500 directory structure and DNS dynamic name resolution. This translates to an Internet-compatible system (where DNS is prevalent) and a solid, scalable, flexible, extensible messaging directory.

The DUA (Directory User Agent) and the DSA (Directory System Agent) are commonly thought to be part of the X.400 specification. They are actually part of the X.500 specification, but you will find them used in many discussions about messaging and X.400. The DUA accesses the directory for the user. It serves a single user, thus controlling access to the directory by name. The DSA provides information to the DUAs and other DSAs and is a part of the Directory.

ADMD and PRMD
Message Handling Systems (MHS) are made up of MDs (Management Domains). They come in two varieties: The PRMD (Primary Management Domain) and the ADMD (Administrative Management Domain). Your Windows 2000/Exchange 2000 organization is an example of a PRMD. The public carrier that delivers your SMTP or X.400 system is an ADMD. For example, our PRMD is our Exchange organization, which is @Ilmarin. Our ADMD is our carrier or Internet Service Provider—a company called Metro 2000 out of Manchester, New Hampshire.

The ADMDs are usually Registered Private Operating Agencies (RPOAs) but can also be systems managed by just an Administrator. The best way to think about the PRMD is as a unique subscription identifier to an ADMD, unique because the ADMD must know how to resolve addresses absolutely. Technically, PRMDs are supposed to send and receive messages from an ADMD only, but it is common practice for PRMDs to communicate with each other also, as indicated in Figure 3-4.

MTA and MTS
An MTS (Message Transfer System) is made up of two or more MTAs. The MTAs communicate to MS (Message Stores) or processes that deliver the message outside of the MTS. The MTA in Exchange 2000 has some important roles that include:

✦ Moving messages
✦ Storing and forwarding messages and, in the process, copying them as needed
✦ Expanding distribution lists

The Messaging Transfer System (MTS) is made up of all the MTAs required to transport a message, as shown in Figure 3-5.
The envelope, please

When using the X.400 lexicon, each e-mail is called an IPM (Inter Personal Message). It consists of two parts: the P1 envelope and the P2/P22 content. The properties that are significant to X.400 are illustrated in Figure 3-6 and listed below.

- The P1 envelope includes a unique X.400 address, the urgency field, and all or part of the routing experienced by that message.
- The P2/P22 content displays a header containing the recipient and the originator’s address and the body containing the message itself, various attachments, and inserted objects.
- Confidentiality is ensured, because the contents are not available for viewing by system services like MTAs. They act only on information found in the envelope.
The envelope contains the Originator/Recipient (O/R) addresses. To handle most global addressing requirements, envelopes fall into one of four categories: mnemonic, numeric, terminal, and postal. The mnemonic address identifies either a distribution list or a user via the ADMD using letters and numbers to access the user. The mnemonic address is the most commonly used envelope address type in current X.400 messaging systems, including the Exchange 2000 X.400 connector. The numeric address identifies a distribution list or user via the ADMD, where only numbers are used for addressing. The terminal address identifies recipients via the ADMD by using either a network address or a terminal ID. The postal address is the snail mail address most are acquainted with. This address can be reached via a physical delivery system (for example, the national postal system). For each type of address, some fields are optional and some are mandatory.
Addressing

The objective in composing a valid X.400 address is to use all the mandatory fields and sufficient optional fields to create an address that uniquely identifies you. Using this process, X.400 addresses may look very different from one another and yet may all be valid. Similarly, they may look valid even when they’re not, because they yield a name that is not unique. The English language, mnemonic address field options are shown in Table 3-2.
Table 3-2
English Mnemonic Address Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Abbreviation</th>
<th>Required Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given Name</td>
<td>G</td>
<td>No</td>
</tr>
<tr>
<td>Initial</td>
<td>I</td>
<td>No</td>
</tr>
<tr>
<td>Surname</td>
<td>S</td>
<td>Yes</td>
</tr>
<tr>
<td>Generation Qualifier</td>
<td>Q</td>
<td>No</td>
</tr>
<tr>
<td>Common Name</td>
<td>C, CN</td>
<td>No</td>
</tr>
<tr>
<td>Organization</td>
<td>O</td>
<td>No</td>
</tr>
<tr>
<td>Organizational Unit 1</td>
<td>OU1</td>
<td>No</td>
</tr>
<tr>
<td>Organizational Unit 2</td>
<td>OU2</td>
<td>No</td>
</tr>
<tr>
<td>Organizational Unit 3</td>
<td>OU3</td>
<td>No</td>
</tr>
<tr>
<td>Organizational Unit 4</td>
<td>OU4</td>
<td>No</td>
</tr>
<tr>
<td>Private Management Domain Name</td>
<td>P, PRMD</td>
<td>Yes</td>
</tr>
<tr>
<td>Administration Management Domain Name</td>
<td>A, ADMD</td>
<td>Yes</td>
</tr>
<tr>
<td>Country</td>
<td>C</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Acceptable X.400 addresses showing some of the variations possible are indicated in Listing 3-1.

**Listing 3-1: Acceptable X.400 Addresses**

C=US/ADMD=ATT/PRMD=@Ilmain/O=MARCOM/OU=TRAINING/S=Guaraldi/G=William
C=US/A=MCI/P=@Ilmain/O=MARCOM/OU=PUBLISHING/S=Guaraldi/G=Paul
C=US/A=ATT;P=Lucent;O=NewEngland;OU=Engineering;S=Often;G=Mat
C=US/A=MCI;P=@Ilmarin;O=Corporate;OU=Execom;S=Guaraldi;G=Robert
C=US/A=ATT;P=MSFT;S=EMAIL
C=US/A= ;p=@Ilmarin;o=Corporate;;s=SamGill

**Case and space insensitivity**
In most X.400 systems, the letters designating the X.400 fields can be upper- or lowercase. Occasionally, you will run into a system where the letters are case-sensitive.
For the sake of simplicity, handle all letters designating X.400 fields as if they were case-sensitive and teach your end-users to do the same. This will help avoid any problems with case intolerant X.400 systems.

In Listing 3-1, there are three interesting things we should point out. The first is that X.400 address component acronyms have abbreviations, and even shorter versions, that are acceptable. For example, ADMD can be represented with A. To see the second, take a look at the last line of the listing. Do you see a value for the A? If you look carefully, you will see that a space was entered. If the space was missing, that address would not be valid. The X.400 standard requires at least one printable ASCII character, and some systems cannot accept a null value. Using a space satisfies both issues. The space is a valid ASCII character if the address represents a user on an internal private network. The last interesting thing is that delimiters may change from system to system. Make sure you know what delimiters the receiving system will accept. This is less of an issue today than it was years ago.

**IPMS and notification**

User systems interface with an IPMS (Interpersonal Messaging System) to transmit mail. An IPMS is made up of the MTSs required to transmit the message all the way from the sender to the recipient. IPMS messages come in two types. The first and most familiar is the IPM (Interpersonal Message), or standard user e-mail message. The second is an IPN (Interpersonal Notification). This type of message is generated by the system (could be the server or the client) and is used to produce notification messages for cases like success or failure of message transmission (receipt and non-receipt of the IPM). IPNs are secondary class messages, as shown in Figure 3-7. Non-receipt notifications are called NRs and Receipt Notifications are called RNs.

![Figure 3-7: An IPN](image-url)
Ever try to shut down your Outlook client when you are in the field, only to discover that Outlook thinks you have mail in your Outbox even though you did not generate any e-mails that session? It’s likely that you received an e-mail with a return or read receipt. When you read the message with the receipt, your Outlook client generated an IPN, and that’s the message that is in your Outbox ready to transmit when you next connect.

The construction of the IPM resembles the standard business letter on which it was modeled. As previously noted, the two parts are the header and the content. The header contains fields that identify the users involved and the relationship among the IPMS, as shown in Table 3-3.

<table>
<thead>
<tr>
<th>Category</th>
<th>Field</th>
<th>Underlying Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Involved User Identification</td>
<td>Originator</td>
<td>O/R Descriptor</td>
</tr>
<tr>
<td></td>
<td>Authorizing-users</td>
<td>Send on behalf of</td>
</tr>
<tr>
<td></td>
<td>Primary-recipients</td>
<td>To:</td>
</tr>
<tr>
<td></td>
<td>Copy-recipients</td>
<td>cc:</td>
</tr>
<tr>
<td></td>
<td>Blind-copy-recipients</td>
<td>bcc:</td>
</tr>
<tr>
<td></td>
<td>Reply-recipients</td>
<td>O/R Descriptor</td>
</tr>
<tr>
<td>IPMS</td>
<td>This-IPM</td>
<td>IPM identifier</td>
</tr>
<tr>
<td></td>
<td>Replied-to-IPM</td>
<td>IPM identifier</td>
</tr>
<tr>
<td></td>
<td>Obsoleted-IPM</td>
<td>List of IPM identifiers</td>
</tr>
<tr>
<td></td>
<td>Related-IPMs</td>
<td>List of IPM identifiers</td>
</tr>
<tr>
<td>Pertinent to IPM times</td>
<td>Expire-time</td>
<td>Time</td>
</tr>
<tr>
<td></td>
<td>Reply-time</td>
<td>Time</td>
</tr>
<tr>
<td>Information describing the IPM</td>
<td>Subject</td>
<td>Maximum of 128 characters</td>
</tr>
<tr>
<td></td>
<td>Importance</td>
<td>High, Low, Medium</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>Confidential, etc.</td>
</tr>
<tr>
<td></td>
<td>Language</td>
<td>P22</td>
</tr>
<tr>
<td></td>
<td>Incomplete-copy</td>
<td>P22 (conversion could not finish)</td>
</tr>
<tr>
<td></td>
<td>Autoforwarded</td>
<td></td>
</tr>
</tbody>
</table>
Table 3-4 helps to explain the design breadth of the X.400 standard and provides the IPM structure references for each part.

<table>
<thead>
<tr>
<th>Body Part</th>
<th>Body Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>IA5 Text</td>
<td>0</td>
</tr>
<tr>
<td>Telex (ITA2 5 bit)</td>
<td>1</td>
</tr>
<tr>
<td>Voice</td>
<td>2</td>
</tr>
<tr>
<td>G3 Facsimile (one flavor of fax)</td>
<td>3</td>
</tr>
<tr>
<td>Text Interchange Format (TIFO)</td>
<td>4</td>
</tr>
<tr>
<td>Telex (T.61)</td>
<td>5</td>
</tr>
<tr>
<td>VideoTexet</td>
<td>6</td>
</tr>
<tr>
<td>Nationally Defined</td>
<td>7</td>
</tr>
<tr>
<td>Encrypted</td>
<td>8</td>
</tr>
<tr>
<td>Forwarded IPMessage</td>
<td>9</td>
</tr>
<tr>
<td>Simple Formatable Document (SFD)</td>
<td>10</td>
</tr>
<tr>
<td>Text Interchange Format 1 (TIF1)</td>
<td>11</td>
</tr>
<tr>
<td>Octet String</td>
<td>12</td>
</tr>
<tr>
<td>ISO6937 Text</td>
<td>13</td>
</tr>
<tr>
<td>Bilaterally defined (Binary)</td>
<td>14</td>
</tr>
<tr>
<td>Binary File Transfer</td>
<td>15</td>
</tr>
</tbody>
</table>

**SMTP — The Basics**

SMTP (or Simple Mail Transport Protocol) is extremely popular today. It was designed with two objectives:

- To transfer mail reliably and efficiently, independent of the underlying transmission system
- To relay mail across transport service environments and deal with verification and expansion of names

In terms of the OSI model, SMTP is a seventh layer, application layer protocol.
SMTP was not built to securely transfer mail. Security was not part of the design criteria, and messaging system administrators and manufacturers deal with security with little help from SMTP itself. If you run a protocol analyzer and examine the data stream passed in an SMTP session, you’ll notice that all the information is transferred clear text—there is no encryption of information between points in a regular SMTP session.

In the world of Exchange 2000, SMTP has additional importance because it is the primary routing protocol. Exchange utilizes SMTP’s capability to relay mail across transport service environments that make up an Interprocess Communication Environment (IPCE).

Do not confuse message routing with lower-level network routing. In a robust WAN, routing can take place at many points.

RFC 821 and 822, maintained by the Institute of Electrical and Electronics Engineers, Inc. (IEEE), are the two primary document sources for basic SMTP understanding. Here is a brief synopsis of each.

**RFC 821**

The process starts when a user mail request is generated. The sender-SMTP establishes a bi-directional transmission channel to a receiver-SMTP usually over port 25 (octal 31). The receiver-SMTP could be either an intermediate destination for the message (part of the route) or the ultimate destination. Roles of the sender and recipient are different. All SMTP commands are generated by the sender-SMTP and sent to the receiver-SMTP. All SMTP replies are sent from the receiver-SMTP to the sender-SMTP in response to the commands. Roles can later be exchanged to facilitate delivery of mail in the opposite direction.

Once a channel is established (in most systems this requires getting physical connection, penetrating fire walls, and sometimes being authenticated), the SMTP-sender identifies the sender with a `MAIL` command. A confirmation in the form of an “OK” from the SMTP-receiver means it can accept mail. An RCPT command from the SMTP-sender follows, identifying a recipient of the mail (there may be more than one). Once again, a confirmation in the form of an “OK” from the SMTP-receiver means the receiver can accept mail for that recipient. Alternatively, the SMTP-receiver can reject the recipient, and the negotiation may continue if there are other recipients involved. When the last negotiation with this receiver is complete, the SMTP-sender transmits the data, which is terminated with a period on a line alone. When and if all goes well, a response in the form of an “OK” is generated. Figure 3-8 shows the 100,000-foot view.

Relay (and ultimately routing) capability requires the SMTP-server to be provided with the name of the ultimate destination host as well as the destination mailbox name. For efficiency reasons, SMTP encourages a single instance of a message be transmitted when there are multiple recipients at the same host where bursting will take place or be managed.
Bursting is the term used to describe creating multiple instances of a message for each recipient on the To, cc, or bcc list.

Commands and replies are not case-sensitive, but are rigid in their syntax. However, for some hosts, user names are case-sensitive.

Listing 3-2 is an example of a simple mail message transmission.

**Listing 3-2: The Dialog of a Simple SMTP Mail Transmission**

Sender: MAIL FROM:<RGuaraldi@Ilmarin.com>
Receiver: 250 OK
Sender: RCPT TO:<MOften@Beta.Londonderry>
Receiver: 250 OK
Sender: RCPT TO:<SGill@Beta.Londonderry>
Receiver: 550 No such user here
Sender: RCPT TO:<PGuaraldi@Beta.Londonderry>
Receiver: 250 OK
Sender: DATA
Receiver: 354 Start mail input; end with <CRLF>.<CRLF>
Sender:
From: BobbG <RGuaraldi@Ilmarin.com>
To: Mat <MOften@Beta.Londonderry>, Paul <PGuaraldi@Beta.Londonderry>
Date: Thu, 25 Jul 2000 23:33:00 -0400
In our example, the message was originated by RGuaraldi. A physical and logical channel was successfully established. MOften and PGuaraldi were accepted. But what if the destination information in the forward path is incorrect? Then you might see one of the responses shown in Listing 3-3.

**Listing 3-3: Response Possibilities When Destination Information Is Wrong**

```plaintext
Sender: RCPT TO:<Gandalf@Bluesock.org>
Receiver: 251 User not local; will forward to <Gandalf@Bluesock.org>

Or

Sender: RCPT TO:<Gandalf@Bluesock.org>
Receiver: 551 User not local; please try Lowell@Bluesock.org
```

SMTP also has the command structure to expand lists and verify names. These are accomplished through the VRFY and EXPN commands. Listing 3-4 shows an example of verify and expand.

**Listing 3-4: VRFY and EXPN Command Usage Examples**

```plaintext
Sender: VRFY Guaraldi
Receiver: 250 Robert L. Guaraldi <RGuaraldi@Ilmarin.com>

Sender: EXPN Writers
Receiver: 250-Robert L. Guaraldi <RGuaraldi@Ilmarin.com>
Receiver: 250-Paul Guaraldi <PGuaraldi@Ilmarin.com>
Receiver: 250-Sam Gill <SGill@MSN.com>
Receiver: 250-Will Guaraldi <WilliamG@Bluesock.org>
Receiver: 250-<MSwaim@Metro200Mail.net>
```
Relaying
SMTP makes a distinction between a mailbox and a route. A mailbox is a final destination originally contained in To, cc, or bcc fields of the message. A route is the path to get there. The routing information is originally contained in the forward path, and as the hosts receive and relay the message, the designation of each host in the forward path should be moved to the reverse path by the server routing the message. The forward path maps how to get to the destination while the reverse path maps how to get back to the message originator. The reverse path “how to get back” information is important both if the message does reach its destination and if a reply is necessary. A forward path might look like this example. It shows, FIRST, SECOND, and THIRD as intermediate host names for routing and FOURTH as the final host name where the mailbox should be found.

@FIRST,@SECOND,@THIRD:Bob@FOURTH

What happens if the message cannot be delivered? The SMTP host holding the message at the time the conclusion that the message cannot be delivered is reached will generate an undeliverable mail message notification. (In X.400, an NR, Non Receipt, notification.) The reverse path is used to send that NR back to the originator.

Domains
Domains were introduced to provide a hierarchical structure to host names. Prior to their introduction, a flat structure existed that was somewhat unwieldy. The idea was for names to be listed and separated by a period, with the most general on the right and the most specific on the left. For example, comune.cento.fe.it, where it stands for Italy, fe stands for Florence, cento is an ISP, and commune is a specific machine on the cento.fe.it network. The success of the Internet today suggests that the domain structure model works well.

SMTP commands and syntax
SMTP commands are transmitted by the user or user process. They are character strings that end in a \texttt{<CRLF>}. A \texttt{\backslash} character specifies that the next character is to be taken literally. For example, Guaraldi\texttt{\backslash}, Robert means “Guaraldi, Robert.” As part of the transaction, a time stamp and return path are transmitted. Minimum values for variables are established in the specification.

Vendors are encouraged by the specification to make the values in their implementations unlimited. For instance, you could have names of unlimited length. However, vendors are not forced to implement unlimited length, so a vendor could be compliant with limitations that satisfy the minimums set in the RFC, but not the request for unlimited value implementations. This can lead to some incompatibility complications on occasion. You may wish to take a look at the specifications when establishing your naming conventions.

Note
Only official domain names are allowed — no nicknames or aliases.

Table 3-5 lists different SMTP commands.
<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hello (Helo)</td>
<td>HELO &lt;SP&gt; &lt;domain&gt; &lt;CRLF&gt;</td>
<td>Identifies user and domain at opening</td>
</tr>
<tr>
<td>Mail</td>
<td>MAIL &lt;SP&gt; FROM:&lt;reverse-path&gt; &lt;CRLF&gt;</td>
<td>Initiates mail to a mailbox</td>
</tr>
<tr>
<td>Recipient (RCPT)</td>
<td>RCPT &lt;SP&gt; TO:&lt;forward-path&gt; &lt;CRLF&gt;</td>
<td>Identifies individual mail recipient</td>
</tr>
<tr>
<td>Data</td>
<td>DATA &lt;CRLF&gt;</td>
<td>Indicates that all lines following this command are the actual e-mail data</td>
</tr>
<tr>
<td>Send</td>
<td>RSET &lt;CRLF&gt;</td>
<td>Initiates a mail transaction to a terminal</td>
</tr>
<tr>
<td>Send or Mail (SOML)</td>
<td>SEND &lt;SP&gt; FROM:&lt;reverse-path&gt; &lt;CRLF&gt;</td>
<td>Initiates a mail transaction to a terminal or a mailbox</td>
</tr>
<tr>
<td>Send and Mail (SAML)</td>
<td>SOML &lt;SP&gt; FROM:&lt;reverse-path&gt; &lt;CRLF&gt;</td>
<td>Initiates a mail transaction to a terminal and a mailbox</td>
</tr>
<tr>
<td>Reset (RSET)</td>
<td>SAML &lt;SP&gt; FROM:&lt;reverse-path&gt; &lt;CRLF&gt;</td>
<td>Aborts the current mail transaction</td>
</tr>
<tr>
<td>Verify (VRFY)</td>
<td>VRFY &lt;SP&gt; &lt;string&gt; &lt;CRLF&gt;</td>
<td>Requests a confirmation that the argument identifies the user</td>
</tr>
<tr>
<td>Expand (EXPN)</td>
<td>EXPN &lt;SP&gt; &lt;string&gt; &lt;CRLF&gt;</td>
<td>Requests a confirmation that this argument identifies a mail list and provides the members</td>
</tr>
<tr>
<td>Help</td>
<td>HELP [&lt;SP&gt; &lt;string&gt;] &lt;CRLF&gt;</td>
<td>Causes the receiver to send helpful information</td>
</tr>
<tr>
<td>NOOP</td>
<td>NOOP &lt;CRLF&gt;</td>
<td>Request for an “OK” in response</td>
</tr>
<tr>
<td>Quit</td>
<td>QUIT &lt;CRLF&gt;</td>
<td>Request for an “OK” and termination of channel transmission</td>
</tr>
<tr>
<td>Turn</td>
<td>TURN &lt;CRLF&gt;</td>
<td>Request to reverse roles after an “OK.” Refusal returns a 502.</td>
</tr>
</tbody>
</table>

If you take any Exchange courses or read material that provides examples of SMTP sessions, you might notice that the first command the client sends the server is \texttt{EHLO}. This is not a typo. A client that supports SMTP extensions should start off the session with \texttt{EHLO} as opposed to \texttt{HELO}. SMTP was defined over a decade ago. Since then, various extensions have been added to address additional functionality that was required. Some extensions include \texttt{8BITMIME, HELP, and EXPN}. Many extensions are defined in their own RFCs. \texttt{EHLO} is defined in RFC 1869.
Replies
As stated earlier, SMTP commands always originate with the sender. Replies are generated to keep requests and actions during the mail transaction synchronized and to ensure that the sender knows the state of the receiver. Exactly one reply is generated by every command, but that reply may be more than one line (often referred to as multi-line). When commands and replies are combined, you have an alternating, elegantly structured conversation. Interruptions in such a conversation are clear to both sides as is the reason for the interruption and the solution for completing the transaction. The absolute minimum number of commands for an SMTP dialogue—and the order they need to come in—are provided in the following list:

HELO
MAIL
RCPT
DATA
RSET
NOOP
QUIT

RFC 822
RFC 822 deals with the format of SMTP messages and includes valuable information for people designing SMTP-compliant systems. For the normal Administrator, there are a few items of note.

Appendix A and the CD that accompanies this book contain URLs where you should be able to find the RFCs.

The SMTP headers defined in RFC 822 are case-sensitive. A few—such as To:, From:, cc:, Received:, Return-path:, Subject:, and Date:—are used everywhere. You might, however, see additional headers in the message. When they show up, they are usually added by e-mail servers and clients for organization and management purposes. RFC 822 also handles formatting when forwarding a message, trace fields for tracing a message, and the return path.

OSI, X.500, LDAP, and DNS
Electronic directories are very similar to the more common paper directories, such as your telephone book. They render information about people (for example, telephone number, street address, work department) in an organized manner. Early electronic directories tended to be written for specific applications. Because those proprietary, early directories were tied to specific applications, the same user information might be stored in many different directories on the same LAN (and sometimes the same server). Keeping simple objects like e-mail addresses or permissions synchronized across all the proprietary directories was nearly impossible, and the need for common nonproprietary directories became apparent. The X.500
specification was written to address this need, among others. Initially, it was written specifically for use with the X.400, to satisfy the X.400 directory object requirement, but was never considered a part of X.400. Since then, the scope of the vision increased to include other applications that now also take advantage of X.500.

X.500 today defines a popular, global model for directory services in the Open System Interconnect (OSI) world. The X.500 model includes both the namespace and the protocol used to access and update the namespace database called Directory Access Protocol (DAP). DAP is feature-rich, but considered quite complex to fully implement. Some companies, however, decided to implement it. Their applications tended to result in large programs—too large for many client applications in the PC world. When DAP appeared too “heavy” to ever be ubiquitous, Lightweight Directory Access Protocol (LDAP) was created. Although the two protocols are similar in functionality, the major difference is that LDAP was built to run directly over the TCP/IP stack or other protocol, avoiding the presentation and session layer overhead. LDAP is also missing some of the more esoteric features of DAP, which allowed manufacturers to write LDAP implementations small enough to run on current PCs without losing anything commonly used. For most requirements, LDAP is sufficiently robust.

LDAP has gone through several iterations. Throughout this book, the use of LDAP refers to LDAP 2 or LDAP 3, unless otherwise noted.

X.500 and the Internet’s Directory Name Service (DNS) are both name services that are important to Windows 2000 and Exchange 2000, but they differ in purpose. X.500 is more general and is used to serve electronic information, such as that found in white pages and yellow pages. X.500 provides greater access to the information it stores than DNS. While X.500 is a searchable directory, DNS can be thought of as more of a look-up system. There are other differences as well. For example, X.500 can be accessed securely via point-to-point encryption—DNS has no such feature. X.500 provides a layer of naming above DNS domains, as X.500 directories are more hierarchical and produce trees with many more levels than the trees produced with DNS. They both have a significant role in directory services and name resolution, are often partnered up and used as a team.

Exchange 2000 takes advantage of the Windows 2000 Active Directory Service and its support for X.500. The ADS is both X.500-like and is integrated with DNS. ADS supports LDAP3 calls to access and update information in the directory. With Windows 2000 and Exchange 2000 directory integration, Exchange services and other applications can access (when permitted) the information maintained by the Active Directory Service, eliminating the need to write or maintain a directory for each application.

**ITU X.500 Standard**

The X.500 standard is the Open System Interconnection (OSI) directory designed by the International Telecommunication Union (www.itu.int) and the International
Organization for Standards (www.iso.ch). What follows is a quick overview of the lexicon concepts that are germane to Exchange and Windows 2000 ADS.

In its simplest form, information in X.500 directories is accessed by a Directory User Agent (DUA) using a typical client/server model and DAP or LDAP protocols. X.500 directories are stored on one or multiple servers in what is known as a Directory Information Database (DIB).

Note: Windows 2000 owns the DIB in Exchange 2000. In Exchange 4.x and 5.x, Exchange had its own Directory Service (DS) and owned the DIB.

**Directory User Agent — client**

The Directory User Agent (DUA) is the application on the client-side that supports the standardized directory functionality, including browsing through directories and accessing directory information. In Exchange, various “client” services access the directory. Exchange has multiple DUAs, such as Exchange Client programs and Exchange Administrator programs.

Note: From the perspective of the client/server model, even if the service is an Exchange Server service, it is acting like a client in its call to the directory.

DUAs are available for Web browsers and almost all kinds of clients (such as Windows, DOS, MACs, UNIX, Linux, and so on).

**Directory Information Database — server**

The Directory Information Database (DIB) stores any objects that might be queried for information by a user or a computer process on behalf of a user. Since the X.500 specification was built to create the directory for all objects of interest to all mankind, the creators needed to impose some structure on the data and some rules for handling the data in that structure and the structure itself. The structure is facilitated by the DIB, which acts as an interface between services that use the directory (like the DSA) and the users. Objects in the directory and entries in the DIB have a one-to-one relationship. Objects are categorized in classes, with each type of object constituting an object class. In order to be created, an object must be recognized as appropriate for an established object class or it will not be allowed. Objects in a class must adhere to the rules of that class.

Note: The DIB can be organized however a user wants it to be, subject to certain requirements for consistency.

**Directory System Agent — server**

A Directory System Agent (DSA) controls access to the directory database. Its role is to provide access to the DIB and DUAs and/or other DSAs. When a client request is received, a DSA may use information in its own database or communicate with
other DSAs to carry out the request. Alternatively, the DSA may refer the client to another DSA that can service the request.

The look-up model allows for distributed DSAs, where each DSA is responsible for only a fragment of the directory and knowledge. In this design, it is not uncommon for a DSA to consult with other DSAs in order to fulfill a request.

The X.500 specifications use the acronym DSA to mean the Directory System Agent. However, in some versions of Exchange, the DSA means the Directory Service Agent. In this book, instances of DSA refer to the X.500 acronym, unless specifically referring to the Exchange DSA.

The Directory Information Tree — server

As the movement to centralize directories has grown, so have the DIBs. It should not surprise anyone that a DIB with 40,000 names can be quite large. To prevent DIBs from becoming unwieldy, X.500 provides the Directory Information Tree (DIT). The DIT provides hierarchical, navigable organization for the DIB. DIT structure and components (Root Entries, Leaf Entries, Non-Leaf Entries, and containers) are illustrated in Figure 3-9.

![Diagram of Directory Information Tree components and structure](image)

**Figure 3-9:** Directory Information Tree components and structure

The components that make up the DIT and their descriptions are delineated in Table 3-6. Many of these terms will be used in this book and others on Exchange 2000 and Windows 2000 ADS.
Table 3-6  
DIT Components

<table>
<thead>
<tr>
<th>Object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root</td>
<td>The root of the tree. No data or entries are allowed at the Root.</td>
</tr>
<tr>
<td>Containers</td>
<td>Objects that have properties and attributes and contain other objects.</td>
</tr>
<tr>
<td>Leaf Objects</td>
<td>The objects of interest that have properties and attributes found in containers. (Leaf objects cannot contain other objects or they become containers.)</td>
</tr>
<tr>
<td>Non-Leaf Objects</td>
<td>Objects that hold the DIT structure information.</td>
</tr>
</tbody>
</table>

Naming Conventions

X.500 is hierarchical. X.500 ensures that a reference does not point to more than one object in the hierarchy by making certain that each entry has a unique name that is used to help define its location in the hierarchy. Uniqueness is achieved by combining name data and location. The unique name value assigned to each object is combined with the specific location of that object. This combination results in a distinguished name (DN) for each object. The DN is an ordered series of Relative Distinguished Names (RDNs) with the final addition of the unique name. In X.500 parlance, the term name expands from the usage earlier in this paragraph to be the unique data value assigned in the DIB and DIT entry for an object of interest.

Unambiguous name

While unique, a name may not be unambiguous across the entire DIB. Within each level of the DIT, names must be unique. So, while there may be two Jane Does within a single directory, there cannot be two Jane Does within the same leaf of the tree or scope. Names must be unambiguous within the scope and unique within the entire DIB.

In large, global organizations, the likelihood that you will have duplicate names in your GAL, which satisfies the X.500 requirement for uniqueness in the DIT but not in the DIB, is about 100 percent. (How many Jian Chens live in China or Joe Smiths in England?) For that reason, thinking about naming conventions is important and will be discussed in other chapters.

Distinguished Name

An object’s Distinguished Name (DN) consists of the path to the object as defined by the location of the ordered series of container objects that the object falls under. As previously stated, this is the path to the object and the unambiguous name.
Relative Distinguished Name

For any object, the Relative Distinguished Name (RDN) is the last part of the Object Distinguished Name, as shown in Figure 3-10.

Directory schema

The X.500 specification calls for the Directory to enforce a set of rules. These rules, called the Directory schema, ensure that the DIB remains structurally intact. They prohibit activities that would cause entries to be malformed. The schema consists of

✦ **Object classes:** One for each type of object as noted.

✦ **Attributes for each class object:** Common name, distinguished name, date of creation, parent, and so on.

✦ **The hierarchical structure:** Organization, forests, leaf objects, and so on.

✦ **Rules:** Governing object creation and changes (for example, which container an object can be created in).

Directory Management Domain

A Directory Management Domain (DMD) is the set of DUAs and DSAs contained within an organization. X.500 specifies two types of DMDs: the Private DMD (PRDMD) and the Administrative DMD (ADDMD).
These two types are analogous to the ADMD and PRMD of an X.400 system. The Administrative DMD is used when service is provided through a public service.

**Directory topology**

Directories come in two topologies: distributed and centralized. A centralized directory is entirely within a single DSA. A distributed directory is shared among more than one DSA. In most cases, you will see distributed directories, especially for medium- to large-sized companies with lots of objects.

**Fragment**

In a distributed directory environment, each DSA only holds a portion of the directory. That portion is called a *fragment*. When a DSA or DUA queries a DSA for information, the queried DSA will determine if it has the information requested in its fragment. If not, the DSA will send the request to the responsible DSA using one of two processes: chaining or referral. In an X.500 directory implementation, both processes are allowed, although one might be chosen over another to diminish latency or for other system-level reasons.

**Chaining**

When a DSA doesn’t have the information the request is looking for, it passes the request off to another DSA, a process called chaining. The responsible DSA then passes the results of the query to the DUA requesting the information.

**Referral**

Redirecting the DUA to make its query directly to the responsible DSA that holds the requested data is known as referral. The DSA is capable of chaining and referral because all DSAs are aware of the information in fragments of all other DSAs, as well as the information in their own fragment.

**Directory protocols**


**Directory Access Protocol**

Directory Access Protocol (DAP) defines the exchange of requests and results between a client DUA and a server DSA.

---

*Note* LDAP does the same job as DAP and is more common today but is not mentioned in the 1983 X.500 specification, as it did not exist at the time.
Directory System Protocol

Directory System Protocol (DSP) defines the exchange of requests and results between two DSAs. In other words, the DSP is the access protocol that makes calls on the directory.

For service element definitions, go to the X.219 Remote Operations Service (ROS) specification.

The access protocol of greatest interest is Lightweight Directory Access Protocol (LDAP), because it is the protocol that is used throughout Windows 2000 ADS and Exchange 2000.

Lightweight Directory Access Protocol

Lightweight Directory Access Protocol (LDAP) was created as an access protocol for X.500-compliant directories that would relieve user applications from the memory and processing overhead associated with the DAP protocol. Though LDAP was originally written for simple read/write interactive sessions, it has evolved into a robust, client-server alternative for a directory access protocol.

Since LDAP is a lightweight version of DAP, it cuts a few corners here and there. For instance, as LDAP clients make requests of the server, the only commitment from the server is that the client will eventually get a response to its request. There is no notion of synchronous behavior, where the requests are handled in order by the server. However, the server will attempt to procure the information and referrals itself, rather than pushing them back to the client for execution, once again eliminating complex code from the user client applications.

LDAP was designed to work on connection-oriented reliable transports. The most popular is TCP/IP. In a TCP/IP environment, the server typically listens on port 389.

Since LDAP is primarily a directory access protocol, we’ll discuss the basics of its command set.

Command definitions

A common programmatic envelope called the LDAPMessage encapsulates all protocol operations to provide the fields common in all subsequent operations. An example is the message ID. The message ID must be unique during each LDAP session. During that session the unique reference must be contained in each and every responses to that session. The LDAPMessage is defined in Listing 3-5.
### Listing 3-5: LDAP Session Syntax

```
LDAPMessage ::= SEQUENCE {
    messageID      MessageID,
    protocolOp     CHOICE {
        bindRequest       bindRequest,
        bindResponse      bindResponse,
        unbindRequest     unbindRequest,
        searchRequest     searchRequest,
        searchResponse    searchResponse,
        modifyRequest     modifyRequest,
        modifyResponse    modifyResponse,
        addRequest        addRequest,
        addResponse       addResponse,
        delRequest        delRequest,
        delResponse       delResponse,
        modifyRDNRequest  modifyRDNRequest,
        modifyRDNResponse modifyRDNResponse,
        compareRequest    compareRequest,
        compareResponse   compareResponse,
        abandonRequest    abandonRequest
    }
}

MessageID ::= INTEGER (0 .. maxInt)
```

Other definitions used in the protocol are included in Table 3-7.
Chapter 3  ♦  Review of Underlying Technological Concepts

Table 3-7
Command Definitions

<table>
<thead>
<tr>
<th>Command</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAPString ::= OCTET STRING</td>
<td>Notational Convenience</td>
</tr>
<tr>
<td>LDAPDN ::= LDAPString</td>
<td>Representation of a DN</td>
</tr>
<tr>
<td>RelativeLDAPDN ::= LDAPString</td>
<td>Representation of a RDN</td>
</tr>
</tbody>
</table>

The **BindRequest**, which is required to initiate a protocol session, must be the first command transmitted by the client application and received by the server. It is shown in Listing 3-6.

**Listing 3-6: BindRequest Syntax**

```
BindRequest ::= [APPLICATION 0] SEQUENCE {
  version  INTEGER (1 .. 127),
  name    LDAPDN,
  authentication CHOICE {
    simple        [0] OCTET STRING,
    krbv42LDAP    [1] OCTET STRING,
    krbv42DSA     [2] OCTET STRING
  }
}
```

In the authentication section, the `simple` option provides for clear text that may be transmitted over the Internet, leaving your password and identification exposed to someone with a protocol analyzer. The Kerberos authentication options provide greater security, as covered in more detail in Chapter 22.

The server will respond with a **BindResponse**.

```
BindResponse ::= [APPLICATION 1] LDAPResult
```

When the session needs to be terminated, an **UnbindRequest** is used.

```
UnbindRequest ::= [APPLICATION 2] NULL
```

The **LDAPResult** is used by the server to communicate the results of the query to the client. It takes the form shown in Listing 3-7.
One of the more powerful and common uses for LDAP is for the client to make an LDAP call to ask the server to do a search for it. The Search command with filtering options is shown in Listing 3-8.
Listing 3-8: LDAP Search Command with Filtering Syntax

SearchRequest ::=  
  [APPLICATION 3] SEQUENCE {  
    baseObject    LDAPDN,  
    scope         ENUMERATED {  
      baseObject            (0),  
      singleLevel           (1),  
      wholeSubtree          (2)  
    },  
    derefAliases  ENUMERATED {  
      neverDerefAliases (0),  
      derefInSearching  (1),  
      derefFindingBaseObj (2),  
      derefAlways       (3)  
    },  
    sizeLimit     INTEGER (0 .. maxInt),  
    timeLimit     INTEGER (0 .. maxInt),  
    attrsOnly     BOOLEAN,  
    filter        Filter,  
    attributes    SEQUENCE OF AttributeType  
  }

Filter ::=  
  CHOICE {  
    and                [0] SET OF Filter,  
    or                 [1] SET OF Filter,  
    not                [2] Filter,  
    equalityMatch      [3] AttributeValueAssertion,  
    substrings         [4] SubstringFilter,  
    greaterOrEqual     [5] AttributeValueAssertion,  
    lessOrEqual        [6] AttributeValueAssertion,  
    present            [7] AttributeType,  
    approxMatch        [8] AttributeValueAssertion  
  }

SubstringFilter  
  SEQUENCE {  
    type               AttributeType,  
    SEQUENCE OF CHOICE {  
      initial        [0] LDAPString,  
      any            [1] LDAPString,  
      final          [2] LDAPString  
    }  
  }

}
The results are returned by the server in the form of a Search response, as shown in Listing 3-9.

**Listing 3-9: LDAP Search Response Syntax**

```plaintext
Search Response ::= 
  CHOICE {
    entry [APPLICATION 4] SEQUENCE {
      objectName LDAPDN,
      attributes SEQUENCE OF AttributeSet,
    },
    resultCode [APPLICATION 5] LDAPResult
  }
```

When a client wants to modify the DIB, the modification cannot be done directly. Instead, the client sends an LDAP ModifyRequest to the server. The ModifyRequest takes the form shown in the following example in Listing 3-10.

**Listing 3-10: ModifyRequest Syntax**

```plaintext
ModifyRequest ::= 
  [APPLICATION 6] SEQUENCE {
    object LDAPDN,
    modification SEQUENCE OF SEQUENCE {
      operation ENUMERATED {
        add (0),
        delete (1),
        replace (2)
      },
      modification SEQUENCE {
        type AttributeSet,
        values SET OF AttributeValue
      }
    }
  }
```
The server responds to a request with a `ModifyResponse`.

```
ModifyResponse ::= [APPLICATION 7] LDAPResult
```

The `ModifyResponse` is only affirmative after all requests are completed. Should any fail, all will fail, and an affirmative will not result. Only one response is provided for all the changes in one `ModifyRequest`.

The `AddRequest` is used when a client requests that a server add an entry on its behalf. The form is illustrated in Listing 3-11.

### Listing 3-11: `AddRequest` Syntax

```plaintext
AddRequest ::= [APPLICATION 8] SEQUENCE {
    entry           LDAPDN,
    attrs           SEQUENCE OF SEQUENCE {
        type          AttributeType,
        values        SET OF AttributeValue
    }
}
```

The `AddResponse` is the return from the server identifying success or failure.

```
AddResponse ::= [APPLICATION 9] LDAPResult
```

The `DelRequest` and `DelResponse` are the two commands used when attempting to delete an entry by specifying its distinguished name. The request is generated by the client and the response is generated by the server. They take the following form:

```
DelRequest ::= [APPLICATION 10] LDAPDN
DelResponse ::= [APPLICATION 11] LDAPResult
```

The `ModifyRDNRequest` allows the client to request that just the last part of the name in an entry be altered. It looks like this:

```
ModifyRDNRequest ::= [APPLICATION 12] SEQUENCE {
    entry           LDAPDN,
    newrdn          RelativeLDAPDN,
    deleteoldrdn    BOOLEAN
}
```

The response is

```
ModifyRDNResponse ::= [APPLICATION 13] LDAPResult
```
The **CompareRequest** is used to request that the server compare an entry in the DIB with an entry provided by the client. It looks like this:

```
CompareRequest ::= [APPLICATION 14] SEQUENCE {
  entry        LDAPDN,
  ava          AttributeValueAssertion
}
```

This request gets a response of

```
CompareResponse ::= [APPLICATION 15] LDAPResult
```

If the client wants the server to abandon a request, it issues an **AbandonRequest** in the following form:

```
AbandonRequest ::= [APPLICATION 16] MessageID
```

Both encoding improvements and security improvements have been implemented as LDAP has evolved.

### LDAP Application Program Interface

With the components delineated above, you can put programs together that do the many things your users want to do with a directory. With a bit of effort and a little creativity, you can build sophisticated utilities. LDAP, for example, has an Application Program Interface (API) that is described in RFC 1823.

### An example of an LDAP program

So you can see what all this looks like when it is put together in an application, we have put an application example on the CD. One of the stars at Microsoft, Erik Ashby, put together a presentation at Fusion a few years ago using LDAP with Exchange 5.x in preparation for Windows and Exchange 2000. He has graciously allowed us to put it on the CD. Dr. Sam Gill adapted it and built the forms.

*Example 5-X Distribution List Manager*

The application on the CD could also be made to work with Exchange 5.5, which supported LDAP reads and writes.

This application was written in VB and can be used to automatically update a distribution list with the Exchange members as defined by the distribution list rule. For example, a rule can be created for all users that are part of the finance department. The distribution list manager will search for all members who have finance in the department attribute and add them to a distribution list. The rule will also take out any entries that do not belong in the list. It consists of two main modules. Enjoy.
X.25

X.25 — produced by the International Telecommunication Union (ITU) — is the recommended interface between two computing devices. Originally published in 1976, the X.25 recommendation has been updated regularly every four years. X.25 was intended to be conceptual rather than a guide for implementation. It defines public data packet switched network interfaces and characteristics for connecting DTE (Data Terminal Equipment) and DCE (Data Communication Equipment) over a Public Service Switched Network and dedicated networks.

X.25 covers the three bottom layers of the OSI specification: Physical, Datalink, and Network. We’ll only be referring to its duties at the network layer in this book. Figure 3-11 shows how it fits into the OSI-layered model.

![Figure 3-11: X.25 in relation to the OSI model](image)

**X.25 Terminology**

As an Exchange Administrator implementing or designing for X.25 connections, there are some terms that you will want to be familiar with. Becoming familiar with these terms will help you interface with the network management team so that together you can design an effective solution. Specifically, within the X.25 specification there are two other protocol terms of occasional interest to Exchange Administrators. These terms are LAPB and MLP.
LAPB

LAPB is the two-layer protocol used to carry X.25 packets. The format of a standard LAPB frame is shown in Figure 3-12. There is no master and slave relationship in LAPB. LAPB works in Asynchronous Balance Mode (ABM). The DTE and DCE devices are treated like equals and send frames at any time. LAPB supports two window sizes. The normal window size is modulo 8 (8 bits) and the extended window size is modulo 128 (16 bits). Modulo 128 has a larger information area and is used for situations like satellite transmissions, where the acknowledgement delay is expected to be large.

<table>
<thead>
<tr>
<th>Flag</th>
<th>Address Field</th>
<th>Control Field</th>
<th>Information</th>
<th>FCS</th>
<th>Flag</th>
</tr>
</thead>
</table>

Figure 3-12: LAPB frame

Multilink Protocol

The Mulitlink Protocol (MLP) operates between the packet layer and various Single Data Link Protocol (SLP) functions. It resides at an upper sublayer of the Datalink layer.

Permanent virtual circuits and public switched networks

Permanent virtual circuits (PVC) and public switched networks are two alternatives for circuits. In either case, X.25 provides a robust transport for a reliable connection. As you recall, that is a requirement of many customers for their e-mail system.

Packet assembly disassembly unit

Other parts of the X.25 specification are not used in the Exchange world. One is called a PAD (packet assembly disassembly) unit. Connecting a DTE device to a PAD is common in public networks, but not used in Exchange. In Exchange, a DCE device connects to a DTE device without a PAD. To provide for timing, if the circuit is private you need a synchronous modem on one side of the circuit. Alternatively, if the circuit is public, it is possible you will have to provide the timing, but it is also possible that the timing circuit is supplied by the carrier.

X.75

X.75 is the signaling system used to connect X.25 when connecting international circuits. X.25 and X.75 are the same on the Datalink layer and use LAPB. On the Network layer these are the same, except that X.75 has a field of variable length for network utilities/facilities, whereas X.25 just has a variable length field for facilities.
TP0 and TP4

Protocols in the OSI model are numbered from TP0 to TP4. TP0 and TP4 are the two commonly used protocols. TP0 is X.25. TP4 is Connectionless Network Layer Protocol (CLNP). TP4 has a number of features that make it sound remarkably like TCP. It provides guaranteed and reliable delivery of packets by using sequence numbers. It also uses checksums to ensure data integrity. Lost packets are retransmitted.

TP4 allows multiple programs to use it by providing additional entrance and exit points from the protocol. They exist at the Presentation, Session, and Transport layers. They are referred to as the Presentation Access Point (PSAP), the Session Access Point (SSAP), and the Transport Access Point (TSAP). By using the access points, each program is distinguished uniquely. An application can enter the stack at any one of these points, as illustrated in Figure 3-13.

![Figure 3-13: TP4 programmatic access points](image)

Caution

Many of the simpler protocol analyzers cannot decode CLNP packets properly.

Internet Protocol

The Internet Protocol (IP) is the method or protocol by which data is sent from one computer to another on the Internet. Gateways and connectors use it for routing (that is, finding paths to other nodes and networks) and to identify networks.
IP addresses are tightly controlled to ensure uniqueness. While you can use any scheme for assigning IP addresses within your own internal network, you are required to use legal addresses on any device that will be exposed to the Internet. An IP address consists of four octets separated by periods. An IP address might look like 206.6.192.1. This is a class C address. It uniquely identifies computer 1 on network 206.6.192.

IP also breaks up the data into smaller units and places them in packets for transmission. Data can be broken up into many packets, each containing routing information. Routers will look at that information and send the packet on its way. If the most direct route is down, routers will try an alternative path. IP has no notion of a session—sessions are handled by the next level up.

The basics of Transmission Control Protocol

Transmission Control Protocol (TCP) is one of the de facto protocols for network computing on the Internet. TCP sits on top of IP and is a connection-oriented protocol. TCP guarantees delivery of all data in the order that it was transmitted. TCP/IP is built into almost all modern operating systems as the default protocol, including Windows 2000, all varieties of Unix, and Linux. Others—such as NetWare that have their own protocol (IPX)—still support TCP/IP, though not as the default protocol. It would be hard for any operating system to avoid support for TCP/IP, as TCP/IP allows the operating system to easily connect to the Internet.

The Application layer protocols sit on top of TCP. Some Application layer protocols include FTP, SMTP, Telnet, HTTP, and NNTP. These all rely on TCP to provide guaranteed delivery of all data in the order in which it was sent. See Table 3-8.

<table>
<thead>
<tr>
<th>Protocol or Application</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTP — File Transfer Protocol</td>
<td>Used for uploading and downloading files</td>
</tr>
<tr>
<td>SNMP — Simplified Network Management Protocol</td>
<td>Used to manage networks</td>
</tr>
<tr>
<td>Telnet — Telnet</td>
<td>Used for terminal service</td>
</tr>
<tr>
<td>SMTP — Simplified Mail Transport</td>
<td>Used to transport e-mail</td>
</tr>
<tr>
<td>NNTP — Network News Transport Protocol</td>
<td>Used for Usenet and news postings</td>
</tr>
<tr>
<td>HTTP — Hypertext Transport Protocol</td>
<td>Used to access and display WWW sites</td>
</tr>
</tbody>
</table>

The TCP/IP stack and its Application layer protocols don’t quite match up to the OSI layers perfectly, but for a close approximation of TCP, IP, and SMTP, see Figure 3-14.
TP4 is the OSI protocol used with CLNP. The X.400 connector will use either TP4 (TCP/IP) or TP0 (X.25) for its transport.

**Domain Name Service**

The need for a Domain Name Service (DNS) is obvious. Users yearn for names that are more easily remembered than IP addresses. DNS is the hierarchical distributed database that allows TCP/IP applications to resolve IP addresses from the more easily remembered hostnames.

TCP/IP implementations with access to the Internet will almost always have access to DNS servers. They may be local to your company on your internal network or external to your company at your ISP or some other third party. Somewhere, a DNS server that is accessible to your applications will exist. Windows 2000 supplies a DNS service, which you may choose to use in your Exchange implementations.

The model for DNS name resolution usage involves a client and a server. The client connects to the DNS server and sends a hostname for resolution to IP address to the resolver, as shown in Figure 3-15. The resolver is the agent program that looks for the resolution or refers the request to another DNS server that will resolve the name. This all occurs before the application makes a connection to the destination client or server.
DNS uses a hierarchical naming structure that takes the form of a tree and has a special node with a null label at the root, as illustrated in Figure 3-16. The maximum size of any node, except the root, is 63 characters. A domain name for each node is the name that starts at the node and works its way back up the tree until it reaches the root. Suffixes of labels in a domain tree are also called domains.

In Figure 3-16, a domain name of comune.fe.it would represent the node, comune (which is a domain), the node, fe (which is the domain name for Ferrara), and the domain name, it (which represents Italy).
Chapter 3  ✦ Review of Underlying Technological Concepts

The top-level Internet domain names have specific uses by convention. Most companies take this into consideration when creating their domain names so they can expose them directly to the Internet without alteration. Table 3-9 delineates some of the naming conventions.

<table>
<thead>
<tr>
<th>Domain Name</th>
<th>Usage and Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arpa</td>
<td>Special domain used for address-to-name mapping</td>
</tr>
<tr>
<td>Gov</td>
<td>Government institutions</td>
</tr>
<tr>
<td>Com</td>
<td>Commercial organizations</td>
</tr>
<tr>
<td>Net</td>
<td>Major network support centers and recently commercial organizations</td>
</tr>
<tr>
<td>Edu</td>
<td>Educational institutions</td>
</tr>
<tr>
<td>2-letter country codes</td>
<td>Country codes, geographical domains</td>
</tr>
<tr>
<td>Org</td>
<td>Not-for-profit organizations</td>
</tr>
</tbody>
</table>

Subtrees of the DNS hierarchy that are administered separately are called zones. Zones can be broken up into smaller zones. When a new machine is added to a zone, the Administrator has the responsibility to see that the name and IP address are added to the database. This will entail either a manual entry, or the Administrator will rely on a service such as DHCP to handle both the IP assignment and the DNS entry. A name server is the computer that has responsibility for the DNS database in a zone. Mapping a name to an IP address is called name resolution.

If a name server is contacted with a request for a name resolution and it does not have the information, the name server contacts another server to get the information. All servers do not know about all other servers, but each knows how to contact root servers, and usually they know about multiple servers higher up the tree.

The tree spoken of here is logical. The actual physical location of the servers can be anywhere. Obviously, when separated geographically, connectivity bandwidth and latency become issues.

To speed up name resolution, name servers cache the names that are requested so that subsequent calls for that same name will get a more speedy resolution.
Hypertext Transport Protocol

HyperText Transport Protocol (HTTP) is the client server protocol that Web clients, such as Microsoft’s Internet Explorer, use to talk to Web servers like Microsoft’s Internet Information Server. Outlook Express can also access a mail server that supports HTTP. HTTP is a client server protocol and the one universally used by the World Wide Web. RFCs 1945 and 2065 cover versions 1.0 and 1.1 of HTTP, respectively.

HTTP uses a “request-response” method of communication. The client generates a request per the specification. The form of the request is the method, the Universal Resource Identifier (URI), and the protocol version followed by a MIME-like (Multipurpose Internet Mail Extensions) message containing request modifiers, client information, and, in some cases, body content. The server responds with a status line that includes the message’s protocol version and an error or success code followed by a MIME-like message containing the server information, meta information, and possibly body content — then often closes the connection.

Communication usually takes place between a User Agent (UA) and an Origin Server (O), but that is not the only possibility. There can be other servers between the two acting as proxies, gateways, or tunnels.

Note: The default port for HTTP is 80.

In Exchange 2000, HTTP is used for Outlook Web Access (OWA). This allows computers that may not have a POP3, IMAP4, or Exchange client available to connect to Exchange. Essentially, users anywhere can receive their e-mail — including on wireless devices. HTTP is also an allowable protocol for Outlook Express.

Network News Transport Protocol

Network News Transport Protocol (NNTP RFC 1738) is the protocol primarily used for managing posts on Usenet newsgroups.

NNTP traditionally operates on port 119. Exchange uses NNTP to allow access to Public Folders. These Public Folders can be exposed to the public, accessed anonymously, or configured for access only after authentication.

Post Office Protocol Version 3

Not all users are endowed with PCs robust enough to run Outlook 2000 in all situations. Sometimes they lack hardware-related memory, disk space, or LAN IP connectivity. Sometimes they lack software-related licenses for Outlook 2000 or financial resources to acquire one. To satisfy those types of resource-challenged users, as well as certain others, the Post Office Protocol version 3 (POP3) was created. POP3
is the latest version of the Post Office Protocol, which was created to enable clients to access e-mail on e-mail servers. It handles downloading and managing e-mail on the server. Traditionally, POP3 clients will download and delete the e-mail off the server, storing it locally on the client.

In essence, POP3 allows a limited User Agent (UA) to receive e-mail from a mail drop and download it to a local machine where it can be worked on offline.

**Tip**

POP3 does not send e-mail. POP3 clients typically use SMTP for this function.

The POP3 transaction looks something like this:

1. The server listens on port 110.
2. When the client wishes to use the maildrop, it makes a TCP connection with the server.
3. The POP3 client sends a greeting when the connection is established and identifies and authenticates (RFC 1734) itself.
4. The POP3 server procure the maildrop resources of the client and then sends a greeting.
5. Alternatively, the POP3 client and server exchange commands and responses until the connection is closed or aborted.

**Note**

No particular authentication method is required of POP3 servers, but some authentication method must be used.

Exchange 2000 has a POP3 connector that enables any e-mail client that supports POP3 to connect and retrieve e-mail. Supporting POP3 and including OWA allows your organization to support machines and operating systems where an Exchange client might not be available.

**Caution**

When deciding which protocol to use, you will want to do thorough research. While POP3 provides you with a light client, it does so at a price. White paper, *MS Exchange Implementation: Client Traffic Analysis*, on the Microsoft Web site, states that POP3 not only lacks features of Outlook 2000 but also may create more traffic because it authenticates differently and more often than an Outlook 2000 client.

### Internet Message Access Protocol Version 4

Internet Message Access Protocol version 4 (IMAP4) is also used with resource-challenged clients. It is covered in RFC 1730. A major difference between POP3 and IMAP4 is that POP3 clients traditionally cause a download of all e-mail followed by a removal or retention of a copy of that e-mail for some period of time. IMAP4 traditionally stores e-mail on the server rather than downloading it to your client (though you can download it if you wish). With IMAP4, you may leave some of the mail on the server, download it all, or use any combination of the two. IMAP4 allows
you to manipulate remote message folders (mailboxes) on the server, including cre-
ating, deleting, renaming, setting and clearing flags, checking for new messages,
retrieving messages, and permanently removing messages. POP3 and IMAP4 also
differ significantly in how they authenticate with the server and how often.

**Note**

IMAP4 does not specify a way to post (send) mail. Most often, this function is han-
dled by SMTP.

IMAP4 requires a reliable data stream supplied by a lower-level protocol. Usually,
that is provided by TCP. When it is, the port used is 143.

Once a client attempts a connection, IMAP4 servers are in one of four states during
the rest of the transaction. Most commands are valid only in certain states. The
states are

- **Non-Authenticated State**: Once the connection is established by the client,
  this state begins, unless the client has been pre-authenticated.
- **Authenticated State**: The user is now authenticated and has to select a
  mailbox to access before commands on messages will be handled.
- **Selected State**: A mailbox has been selected and activities against that
  mailbox are acted on.
- **Logout State**: The session is being concluded. The server will close the
  connection.

All transactions use a client server model being initiated asynchronously by the
client (first) and then by the server. At all times during the transaction, the client
must be prepared to accept information it did not request from the server. For
instance, the server may need to update information about the mailbox, such as the
size, if some other process altered it while the connection was in progress. This dif-
fers from POP3, where the server generally takes exclusive ownership of and locks
the mailbox, while the connection is in place.

**Caution**

As with POP3, you will want to delve deep into the issues when deciding which
e-mail retrieval protocols you will support. While IMAP4 provides you with a client
that can be successful when running on a resource-challenged machine, it does so
at a price. White paper, *MS Exchange Implementation: Client Traffic Analysis*, on
the Microsoft Web site, states that IMAP4 not only lacks features of Outlook 2000
but also may create more traffic because it authenticates differently and more
often than an Outlook 2000 client.

**Summary**

Both SMTP and X.400 are standards that the industry relies on for specifying the
rules by which applications can generate and transmit mail successfully with other
compliant systems across different transports. They are both commonly used in
Exchange organizations. They each contain lexicons, acronyms, and concepts that
modern messaging specialists will want to be acquainted with. Both are OSI model application layer standards. Neither deals with security explicitly within their standard but rather gets whatever security is relevant from other applications (underlying operating systems, client applications, signing and sealing, and so on) and network transmission devices.

X.500 concepts dominate virtually all the modern directories and meta directories. Accessing X.500 today is commonly done with LDAP. Digesting the lexicon and some of the structure and concepts is important for later chapters, where we get to Windows 2000 directory and how Exchange fits into it and uses it. Similarly, grounding in LDAP will help you with the chapters on development, as well as the ones on the directory.

You’ve been exposed to some of the basics of X.25. You’ve also learned about TP4 in relation to application entry and exit points.

TCP/IP is a protocol suite used on the Internet. We discussed the basics of its two most important parts: TCP and IP. IP is a routing protocol that routes data packets from one place to another. TCP is a connection-oriented transport protocol. TCP/IP working together guarantee that all data will arrive at the destination intact and in the order in which it was sent.

Windows 2000 and Exchange 2000 utilize standard Internet application layer protocols to allow Administrators to support diverse operating systems and machines. Understanding these Internet application level protocols and the other technologies discussed will help in later chapters.
Exchange 2000 is a sophisticated product composed of many services and subsystems. It is integrated into the Windows 2000 platform and relies on many of its protocols, services, and subsystems. Together, they deliver on the promise of a scalable, easy-to-manage messaging and collaboration platform. In order to understand how to administrate Exchange, it is important to understand the core components of Exchange 2000 and its dependencies on Windows 2000. In this part of the book, you explore the components and dependencies that lie at the core of almost any Exchange 2000 installation. Knowledge of the underlying components will provide a solid foundation for building your understanding of Exchange administration.
Server Overview and Architecture

This chapter is an introduction to the Exchange 2000 architecture, in which we discuss the different components and features associated with each version of Exchange. Once you have the component architecture under your belt, you will have a better idea how Exchange works holistically. And knowing how all the pieces fit together and which Exchange version must be purchased to procure each feature is covered in this chapter.

Exchange 2000 Architecture

The Exchange 2000 architecture views the messaging and communications platform as the backbone of a messaging system. The primary function of the Exchange 2000 platform is to send e-mail, including the message header and body. Exchange 2000 supports many different client standards. It also provides flexibility in how clients access servers, with virtual server and front-end/back-end server optional configurations.

The Exchange 2000 architecture is composed of a client/server messaging system, as shown in Figure 4-1. A client/server messaging system distributes the processing of information between the client and the server. In this type of system, executable program code that runs on the server processes requests received from the client.

The Exchange 2000 architecture facilitates the use of virtual servers. Virtual servers enable the Exchange administrator to specify Internet Protocol (IP) settings as though two separate servers were available, as shown in Figure 4-2. Using virtual servers, an administrator can configure different clients to use different virtual servers.
The Exchange 2000 architecture includes front-end/back-end servers, as shown in Figure 4-3. The front-end server is the point where all clients connect, and the back-end server holds the databases. In comparison to earlier versions of Exchange, the front-end/back-end architecture allows the messaging designer a greater degree of...
freedom to distribute and tune the Exchange services to better meet the requirements of the organization. In addition, the use of front- and back-end servers facilitates load balancing. The front-end server provides a unified namespace for all clients. For example, if you didn’t implement front- and back-end servers, and you had multiple mailbox servers, each client would have to specify their own mailbox server. When you implement multiple front- and back-end servers, clients only have to specify their front-end server. Depending on the client access load and the back-end storage capability, you can configure multiple front-end/back-end servers.

![Figure 4-3: Front-end/back-end servers](image)

Exchange 2000 supports several different clients:

- **MAPI**: The workgroup installation of Outlook is a Messaging Application Programming Interface (MAPI) client. The workgroup installation of Outlook uses Remote Procedure Calls (RPCs) to connect to Exchange 2000. Outlook can connect to message and directory information.

- **POP3**: Outlook Express and the Internet installation of Outlook support Post Office Protocol, version 3 (POP3). Other clients, such as Netscape Navigator, also support the POP3 client. POP3 is a read-only protocol, which means that you can use POP3 to read, but not send, messages.

- **IMAP4**: Outlook Express and the Internet installation of Outlook support Internet Message Access Protocol, version 4 (IMAP4). Other clients, such as Netscape Navigator, provide IMAP4 support and can connect to Exchange 2000. IMAP4 is like POP3 in that it is a read-only protocol. However, IMAP4 provides additional support, such as reading from multiple folders — not just the Inbox.

- **SMTP**: POP3 and IMAP4 clients leverage SMTP to send mail. Exchange 2000 supports SMTP clients like Outlook Express and Eudora.

- **LDAP**: Exchange 2000 leverages the Lightweight Directory Access Protocol (LDAP) support in Windows 2000. In essence, LDAP clients can connect to the Active Directory Service (ADS) and global catalog servers in Windows 2000 to retrieve address lists. Outlook Express and the Internet installation of Outlook support LDAP. Other clients, such as Netscape Navigator, also provide LDAP client support. This allows clients to select addresses when sending e-mail.
✦ **NNTP:** Network News Transport Protocol (NNTP) is most commonly used for USENET groups. NNTP is an Internet standard for sharing large folders of information. Exchange 2000 supports NNTP, as does Outlook Express and the Internet installation option of Outlook. Other clients, such as Netscape Navigator, also support NNTP client access.

✦ **HTTP:** Outlook Web Access provides Hypertext Transfer Protocol (HTTP) client access to Exchange 2000. Web browsing clients that support frames, JavaScript, and Dynamic Hypertext Markup Language (DHTML) (such as Internet Explorer) can take advantage of Outlook Web Access features that have been greatly improved since Exchange 5.5.

✦ **ExIFS:** Exchange Installable File System (ExIFS) enables users to view and read mail using Microsoft Office 2000. It provides users with the ability to open mail just as they would open any other file. You can store Office 2000 documents in, and retrieve them directly from, the Web Storage System in Exchange 2000. The ability to use the Web Storage System in this manner enables you to design a consistent and centralized work environment.

Each client supports multiple message types. Each message type has two basic parts: the message header and the message body. The message header carries information about the recipients of the message, as well as instructions about how the message will be handled during transport. This information is stored in a sequence of header fields. These fields describe identifying information such as the message sender, the recipient of the message, and the time and date the message was sent.

The message body contains the actual information of the message. The body is composed of a sequence of body parts. Each part is an information object intended to convey information between users. Examples of some possible information objects include the text of a message or a Microsoft Excel spreadsheet attached to a message.

Regular e-mail messages move across connectors (previously called gateways) both within your messaging environment and outside of it. The most ubiquitous connector in use today is the SMTP connector, which can be used to connect your environment to others on the Internet.

Other messages move across new message models like instant messaging. They use a different format and different connection model. In a very broad sense, conferences could also be thought of as a new message model. Once again, a different connection and message model is in use with conferencing.

Keeping all the pieces of the messaging system working involves storage, monitoring, utilities, fault tolerance, components of Windows 2000, programmatic platform components, and third-party add-ons.
Versions of Exchange 2000

Exchange 2000 comes in three flavors, each of which has different capabilities and components. You want to ensure that you get the right version with the right mix of components and features for your needs. Upgrading to a more expensive, feature-loaded version often involves quite a bit of reinstallation to take real advantage of the new features, so a bit of planning and forethought is advisable. Each version has a different price point, and Client Access Licenses are required for all clients.

Microsoft Exchange 2000 server

Basic server meets the messaging and collaboration needs of businesses that only need one server running Exchange. Exchange 2000 Server is limited to a single, 16-gigabyte (GB) database per server. It does not support Chat, clustering, or distributed configuration.

Microsoft Exchange 2000 Enterprise server

Enterprise server is designed for organizations that need multiple servers and features, such as unlimited message storage, active/active clustering, and the ability to host multiple stores per server.

Microsoft Exchange 2000 Conferencing server

Conferencing server meets the need to organize and manage data, voice, and video conferencing between people, regardless of location. Exchange 2000 Conferencing server can be used in conjunction with Exchange server or Enterprise server. Data Conferencing Service enables Administrators to prearrange an electronic conference, manage schedules, share applications, and use multicast video conferencing. Meeting invitees, using a T-120 compatible client application (such as Microsoft NetMeeting), participate in an interactive collaboration space where multimedia data is accessed and shared. Data Conferencing is available only in Exchange 2000 Conferencing server.

Overview of Exchange 2000 Features and Components

Exchange 2000 delivers several features that include clustering, distributed configuration architecture, fault-tolerant SMTP routing, support of multiple message databases, an enhanced programming platform, administrative and routing groups, and Windows 2000 integration. As we examine these components, we will highlight the features and utility they deliver.
Fault tolerance

Many of the components of Exchange 2000 can contribute to fault tolerance in your design if you utilize them properly. Having multiple connectors to the same site is a good example of using fault-tolerant options to create fault tolerance. If you do build multiple connectors, and one of your connectors or its underlying path is not working, Exchange’s link state routing will find the alternative path for your messages automatically. Unlike clustering, the sole purpose of fault tolerance is to provide high availability of system services.

An Exchange 2000 cluster, as shown in Figure 4-4, consists of two connected physical computers, or nodes, and a shared storage device, such as a Redundant Array of Independent Disks (RAID)-5 disk set channel. The cluster can provide a redundant hardware solution. Since Exchange 2000 services can run on one (or both) of the nodes in the cluster, clients can connect to either node in the same way that they would connect to a standalone server.

Figure 4-4: Clustering

To set up Exchange clustering, you need the Enterprise server and Windows 2000 Advanced server or Data Center versions.

You can use Windows Clustering to monitor the services on all nodes. In the event of a failure, Windows Clustering restarts or moves the services on the failed node to a functional node. In the case of planned outages, you or whoever is chartered with administrating the cluster can manually move the services to other nodes. The
ability to actively use all the servers in the cluster at all times reduces system costs while increasing reliability because you do not have to dedicate servers for disaster recovery.

With Exchange 2000 clustering, the client will experience an interruption of service during the time the virtual server is in a transition state.

Exchange 2000 server supports a type of clustering called active/active. Active/active clustering describes clustering when both members of the cluster are online and able to accept user service requests — unlike active/passive clustering, where only one member of a cluster provides service to users at a time. Active/active clustering provides a huge benefit to organizations that cannot justify the cost of having idle hardware (in a passive/active configuration) to provide high availability.

The Information Store service manages the sophisticated work of writing transactions of data to the various database structures used in Exchange 2000. Some of those database structures are improved versions of structures seen in earlier versions. The public folder and mailbox store are examples. Others (such as the Web Store) are entirely new and not seen in earlier versions of Exchange.

**Mailbox store and public folder store**

Both the mailbox and public folder stores are databases created by Exchange 2000. The former stores private messages for recipients and the latter stores the public folder data. Both are accessed through the Exchange 2000 Information Store service.

**Multiple message databases (Enterprise server only)**

Exchange 2000 Enterprise server supports multiple physical message databases. Multiple databases enable greater scalability, reduced database backup and restore time, more efficient management, and increased reliability.

**Stores**

A store is a logical database. A logical database in the Exchange 2000 server Information Store consists of two database files: the streaming file (.stm) and the MAPI store (.edb). The streaming file refers to a native (or Multipurpose Internet Mail Extensions [MIME]) streaming database file. The .edb file refers to a rich-text database file. These files provide access for both Internet (POP3 and IMAP4) and MAPI (Outlook) clients.

You can define multiple stores with Exchange 2000 Enterprise 2000. Because stores have no programmed size limit, you can use multiple stores to enhance the flexibility of backup and restore tasks.
Storage groups
Stores are contained in storage groups. Each storage group shares a set of transaction log files and is represented by a single ESE instance managed by the store process. Transaction log files provide detailed logging of every message sent and received, which provides recoverability capability. You should never delete these files, because they are crucial to a successful restoral. The number of storage groups you have will depend on what your hardware looks like, what you are trying to accomplish, how many backup units you have, and so on.

Note
Exchange 2000 Enterprise supports multiple storage groups and multiple databases in each storage group. Together, use of multiple storage groups and databases expand the storage capacity of a single server dramatically over earlier versions of Exchange while providing for faster transaction writes and restorals than when the databases were monolithic.

Web Storage System
The Web Storage System, as shown in Figure 4-5, is a storage platform that provides a single repository for managing multiple types of information within one infrastructure.

Because the Web Storage System combines the features and functionality of a file system, the Web, and Exchange 2000, you can access and manage information through a single Uniform Resource Locator (URL) location. There are five key components to the Web Storage System.
ExIFS: ExIFS stores streaming data, such as audio and video, and also enables clients, such as Office, to read and write documents. ExIFS uses a data model that supports both hierarchical collections (folders) and heterogeneous collections (folders that accept any file type).

Native Content Store: The native content store contains data from non-MAPI clients in native MIME format. No conversion is performed on the data unless a MAPI client accesses the data.

URL Addressing: You can access Exchange folders and mailbox objects by using URL addresses based on the folder name. For example,

http://servername/exchange/john/Inbox

will display John’s Inbox folder in your Web browser.

Web Distributed Authoring and Versioning (WebDAV): WebDAV is an extension of the HTTP protocol that allows you to write to Web sites and supports multiple authors. This ensures that Web pages are protected from multiple authors making changes at the same time.

WebDAV enables the implementation of Web folders so that files can be opened and saved using Port 80. This enables users, such as Office 2000 users, to store and retrieve Office documents directly to and from the Web Storage System using the standard File Save/As and File Open dialog boxes. This feature provides a consistent model and set of tools for managing both e-mail and documents.

System Attendant

The System Attendant is a service that runs on an Exchange server and keeps many of Exchange’s features functioning. For example, it maintains the link state tables for routing.

Routing Engine service

The Routing Engine service coordinates the transfer of messages. If the Routing Engine service stops, Aunt Emmy will probably not get the e-mail you just sent her.

Windows 2000 and Active Directory

Windows 2000 is not part of Exchange 2000, but it is part of Microsoft BackOffice. As with other BackOffice platform applications, Exchange 2000 is heavily integrated with Windows 2000. This integration is exemplified in multiple areas: integration with the Windows 2000 Active Directory, use of underlying transport protocols for connectors, integration with the Internet Information Service (IIS) for Internet protocol support, utilization of the Microsoft Management console, and integration with the Windows 2000 permissions and security features.
Integration with Active Directory

All Exchange 2000 directory information is stored in Active Directory. The Active Directory forest defines the boundaries of the Exchange organization. Because Active Directory defines boundaries, it is not possible to have a single Exchange organization span multiple non-trusted Active Directory domains.

Active Directory stores data for a large and customizable set of objects. Integration with Active Directory increases system performance and manageability while making directory management easier. Some of the benefits of the integration with Active Directory are listed below.

✦ Unified administration of Exchange 2000 and Windows 2000 directory objects enable an administrator to manage all user data in one place, with one custom MMC.
✦ Native Windows 2000 Discretionary Access Control lists (DACLs) are used in the Exchange 2000 Information Store.
✦ Security groups in Windows 2000 can automatically be used as distribution lists, eliminating the need to create a parallel set of distribution lists for each department or group.
✦ LDAP is the preferred access protocol used to query, add, update, and delete directory information.
✦ The Active Directory Connector (ADC) that is provided on the Exchange 2000 CD synchronizes directory information from legacy Exchange directories to Active Directory in Windows 2000.
✦ When Exchange 2000 is installed, user and group objects are extended.
✦ The Active Directory provides replication facilities to Exchange 2000. The Active Directory has the capability to replicate each changed or updated attribute rather than the entire object.

Exchange 2000 stores its information in the Active Directory. The information is partitioned into three categories: domain, configuration, and schema data.

The domain partition contains all the objects in the directory for a domain, including all recipient objects. Domain data in each domain is replicated to every domain controller in that domain, but not beyond its domain. Domain objects include recipient objects such as users, groups, and computers.


The schema partition contains all object types, and their attributes, which can be created in Active Directory. This data is common to all domains and domain trees in the forest and is replicated by Active Directory to all domain controllers in the forest.
Groups are one of the more important components in Exchange. Two types of groups in particular need your attention: administrative and routing.

**Administrative groups**
An administrative group is a collection of Exchange 2000 Active Directory objects that are grouped together for the purpose of managing permissions. The collection of administrative groups defines the administrative topology of an organization. An administrative group can contain zero or more routing groups, public folder trees, policies, monitors, servers, conferencing services, and chat networks.

If your organization contains two distinct sets of administrators that manage two distinct sets of servers running Exchange 2000, you can create two administrative groups containing these two sets of servers.

In order to establish permissions, you need to add the appropriate Windows 2000 users and/or groups to the security settings on the two administrative groups.

**Caution**

After you install a server into an administrative group, you cannot move it to another administrative group.

Exchange 2000 creates an administrative group by default when you install the first server in an Exchange 2000 organization. Alternatively, before you install the first server, you can install just the System Manager program. Then, you can create administrative groups, or you can rename the default administrative group. You can still create additional administrative groups after installation. Be aware, though, that you can only join an administrative group during the server installation process.

**Routing groups**
A routing group is another important group type. It is a collection of connected servers running Exchange. At least one server in each routing group must be running Exchange 2000. Messages sent between any two servers within a routing group are routed directly from source to destination. Point-to-point, 24-hour connectivity is required between servers running Exchange in the same routing group.

Routing groups are a separate concept from administrative groups. Routing groups define how mail is transferred, while administrative groups define access permissions and policies.

Exchange 2000 creates routing groups by default when you install the first server in an Exchange 2000 organization. During the installation process, you can rename the first default routing group.

You can also change and move servers between routing groups. You cannot create additional routing groups during the installation process. After you create the groups, you can install additional servers in those groups. You can send messages between routing groups by configuring a connector. If you define the connector as a routing group connector, the connector will identify a logical path between routing groups.
Exchange 2000 server uses SMTP as the primary messaging protocol between servers within a routing group. Using SMTP ensures better interoperability between Exchange 2000, the Internet, and other messaging systems.

**Connectors**
To send mail outside a routing group, Exchange 2000 sends the message, as shown in Figure 4-6, through a connector. If you send a message to another server in the same routing group, the message will not travel through a connector. The servers within a routing group are automatically configured to send mail directly to one another. There are several types of connectors, including the routing group connector, SMTP connector, and X.400 connector.

![Figure 4-6: Exchange 2000 connectors](image)

**Microsoft Management Console support**
The Exchange System Manager is a standard Windows 2000 MMC (Microsoft Management Console) snap-in that manages Exchange 2000 settings and tasks.

You can also add other snap-ins to manage specific Exchange 2000 objects. Using snap-ins enables you to customize the MMC to include only the Exchange 2000 functions to which you need access.

Table 4-1 details which snap-ins are available with Exchange 2000.
Table 4-1

<table>
<thead>
<tr>
<th>MMC Snap-Ins Available with Exchange 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Snap-in</strong></td>
</tr>
<tr>
<td>Exchange System Manager</td>
</tr>
<tr>
<td>Exchange Advanced Security</td>
</tr>
<tr>
<td>Exchange Message Tracking Center</td>
</tr>
<tr>
<td>Exchange Conferencing Services</td>
</tr>
<tr>
<td>Microsoft Exchange GroupWise connector</td>
</tr>
<tr>
<td>Protocols</td>
</tr>
<tr>
<td>Exchange SMTP</td>
</tr>
<tr>
<td>Exchange Servers</td>
</tr>
<tr>
<td>Exchange X.400</td>
</tr>
<tr>
<td>Exchange cc:Mail</td>
</tr>
</tbody>
</table>

Integration with IIS

In Exchange 2000, the Internet protocols run as part of the Internet Information Services (IIS) process. By incorporating the protocols into IIS, system architects may host protocols on a different server from that on which the Information Store service runs. This is the front-end/back-end configuration we spoke about earlier in the chapter.

Because Internet protocols now run as part of IIS, and Exchange 2000 is dependent on them, IIS is now required for Exchange 2000 to function.

Security integration

Exchange 2000 includes multiple types of security. You can secure the content of a message through signing and sealing. You can protect it from viruses with third-party products on the client side or on the server side. You can secure authentication and to a degree, transmission through connectors by configuring them properly. With Exchange 2000, you secure objects and access by leveraging Windows 2000 security through the use of authentication and permissions. Users log on to Exchange 2000 and, after they are authenticated, have access to resources based on their permissions.

In Chapter 22, we cover security in more detail.
Before users or processes can access Exchange 2000, they must log on to Windows 2000 Server by supplying a unique user name and password. The system must validate or authenticate this logon information for the user or process in order for the user or process to have access. For instance, when an Outlook user logs on, Windows 2000 Server identifies a security context. The security context determines the user’s access to system services, including group membership. That user only needs to log on once in order to gain access to Exchange 2000.

Within an Exchange organization, permissions control access to resources and provide specific authorization to perform an action. Permissions are a key component of Exchange administration.

**Programmatic platform components**

Exchange 2000 is considered a development platform. Not only can it be used to develop end-user collaboration and messaging applications, it can also be used to develop enterprise Exchange administration tools using some of the new features.

**Event sinks and triggers**

The Exchange 2000 server includes Event Sink and Trigger Support. This feature enables developers to design and integrate workflow processes into Exchange 2000 server processes by using Exchange 2000 as a programming platform.

An event is a system occurrence that you can use to initiate a defined system action. Examples of events include an SMTP command sent to a server, the generation of an SMTP error, or a page saved in the Information Store.

An event sink is a process that intercepts an event and adds an action or changes the action. You can use an event sink to extend the functionality of Exchange 2000. For example, you can write an event sink that initiates a process when an object is saved to a particular folder. Exchange 2000 provides the following event sinks:

- **Transport event sink**: Pass an incoming or outgoing message through a custom process before the message is stored or relayed to a recipient. Transport event sinks can alter the structure of the message.

- **Protocol event sinks**: Extend SMTP functionality. You can use protocol event sinks to filter messages. For example, you can add a protocol keyword to identify all incoming SMTP messages. That keyword would invoke a header search of all the incoming messages for specific senders. It would then reject the message if the sender name in the message is the same as a name in a rejection database.

- **Information Store event sinks**: Use whenever the Information Store receives, sends, deletes, or moves a message. As an example, if you need to export data posted from a custom form to a certain public folder to an external database, you can create an Information Store sink to initiate the export when the message is posted.
Forms
Forms, including Web forms, are a powerful component of Exchange 2000. Forms have been part of Exchange since its first release. Web forms are new to Exchange 2000. Forms provide a User Interface (UI) to view the elements of an item in an Exchange folder. Web forms allow the user to view the information contained within an Exchange item using a Web browser.

Utilities and monitoring
Exchange 2000 contains a number of utility programs that are used to remedy certain problems, including Inbox repair tool, ESEUtil, RPC Ping, MTA Check, ISInteg, Error.exe, filever.exe, and mdbvue.exe.

Monitoring utilities include Link Monitor for alerts and actions when a connector or link is down and Server Monitor for alerts and actions when a server service is down.

Other components
The vision of Exchange as a collaboration platform continues to grow as the industry finds new ways to take advantage of a connected universe. To facilitate collaboration, Exchange 2000 introduced new instant messaging and conferencing capabilities.

Instant Messaging service
The Instant Messaging service, which was designed primarily for individuals who want to have one-on-one conversations with each other or with multiple users, enables individuals to communicate with each other instantaneously, as if they were talking on a telephone. With Instant Messaging, individuals can immediately view messages written by all users online. It also enables them to determine if other users are online. Instant Messaging, for example, could be used for the online support of customers—customers could ask questions through Instant Messaging and receive an immediate response from anyone who is available online.

Data and voice conferencing
Sharing voice and data dynamically is the province of Microsoft Conferencing server. While sold as a separate product, it is considered a component of Microsoft Exchange 2000.

Full-text indexing and search
Exchange 2000, as shown in Figure 4-7, allows you to create indexes to provide high-speed and detailed searching of Exchange 2000 content. Administrators can enable full-text indexing by using the Exchange System Manager.

An index is basically a list of key words that point to objects, such as attachments, within the Exchange 2000 databases. The Indexing Service runs the processes to build that list. You can search e-mail messages, attachments, documents, and Web
content using full-text indexing. Without full-text indexing enabled, you can only search e-mail messages.

![Diagram showing full-text indexing and search]

**Figure 4-7:** Full-text indexing and search

**Address lists and the Off-Line Address Book**

Address lists, which are similar to function to an address book, are a feature of Exchange 2000. You can use address lists in Exchange 2000 server to limit the addresses viewed by clients. For example, an Internet Service Provider (ISP) could host several companies on one Exchange server and configure address lists to allow a company to view only their own company information.

The Off-Line Address Book is a text-based copy of all or part of the address list. It is stored in the system public folder. It is downloaded by remote users and used in conjunction with other local address books for addressing while not online.

**Recipients and contacts**

Often forgotten, the recipient is an important component of any messaging system — especially Exchange. Recipients are often referred to as Exchange users. The distinguishing feature is that they have an Exchange mailbox and are Windows 2000 Server “Users.” Contacts, on the other hand, are names and information for users who do not have mailboxes on your Exchange messaging environment.

**Summary**

In this chapter, we provided you with an overview of the features and components of Exchange 2000. As you can see, there are many enhancements and additions over earlier versions of Exchange. You learned about the major components and how they fit together to make a robust messaging and collaboration system, and how Exchange 2000 components can open opportunities for development. You were exposed to the differences in Exchange 2000 versions and how they correlate to the basic component architecture of Exchange 2000.
Exchange System Manager

This chapter introduces the Exchange System Manager, a Microsoft Management Console (MMC) snap-in that provides a framework for containing all other Exchange snap-ins, so that you can manage an entire Exchange enterprise from a single console. As a snap-in—that is, defining the MMC view you see and what you are capable of doing—the Exchange System Manager provides a consistent administrative experience for Administrators who deal with all facets of Exchange, including public folders, servers, routing, and policies.

About the Exchange System Manager

As an Administrator, you need to configure, maintain, and secure your Exchange organization. The Exchange System Manager snap-in provides the configuration options you need in one convenient window. Since you will primarily use the Exchange System Manager to administer the Exchange 2000 organization, this chapter focuses on this utility.

If you are unfamiliar with how the MMC is built, the lexicon that defines its parts, or how it is used, consult the help screen, Windows 2000 Server Bible, or the Microsoft Resource Kit (Microsoft Press 2000) for more information.

Starting Exchange System Manager

Assuming you are logged on to Windows 2000 under a domain user account that has administrative permissions for the Exchange server you are trying to administer, you can start the Exchange System Manager from the Microsoft Exchange program group, as shown in Figure 5-1.
If your privileges are adequate, the Exchange System Manager connects, by default, to a domain controller on the same subnet as the computer running the Exchange System Manager. The domain controller to which Exchange System Manager connects is determined by the Domain Name System (DNS) entries. If no domain controller exists on the same subnet, a domain controller will be chosen from within the same Windows 2000 site. After the Exchange System Manager connects to a domain controller, the Active Directory is queried to populate the console with data applicable to Exchange 2000.

**Tip**

If you want to direct the Exchange System Manager console to a specific domain controller, add the Exchange System Manager snap-in to an MMC console. Before adding the snap-in to the console, you will be prompted for the specific domain controller to administer. This domain controller information will be maintained in the saved console file.

In two cases you may want to override connecting to the default domain controller:

- If you need to bypass the Active Directory replication latency
- If you want to use the same Administrator computer to connect to multiple domain controllers in different Windows 2000 forests or to manage different companies or divisions
Exploring the Organization Object

The Organization object is the top-level container for all other Exchange 2000 system objects. You can access the properties of an Organization object using the Exchange System Manager. Table 5-1 describes the options in the Property dialog box of the Organization object.

<table>
<thead>
<tr>
<th>Tab</th>
<th>Option</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Display routing groups</td>
<td>Displays the organization's routing group information.</td>
</tr>
<tr>
<td></td>
<td>Display administrative groups</td>
<td>Displays the organization's administrative groups (collections of Exchange objects that are grouped together to simplify management of permissions); default is disabled.</td>
</tr>
<tr>
<td></td>
<td>Operation mode</td>
<td>Displays whether the organization is running in mixed mode (default) or native mode.</td>
</tr>
<tr>
<td></td>
<td>Change operation mode</td>
<td>Converts the organization to native mode. Select this option only when you are certain that you will no longer be coexisting with Microsoft Exchange Server 5.5; this action is not reversible.</td>
</tr>
<tr>
<td>Details</td>
<td>Creation Date</td>
<td>Displays when the Organization object was created in Active Directory.</td>
</tr>
<tr>
<td></td>
<td>Last modification</td>
<td>Displays the date and time of the last modification to the Organization object.</td>
</tr>
<tr>
<td></td>
<td>Administrative</td>
<td>Provides additional information (added by an Administrator) about the Exchange organization.</td>
</tr>
<tr>
<td>Security</td>
<td>Name</td>
<td>Displays the users and groups that currently have permissions on the Organization object (click Add or Remove to modify this listing).</td>
</tr>
<tr>
<td></td>
<td>Permissions</td>
<td>Displays access permissions for the object selected in the Name window (click Allow or Deny to modify the access rights of the selected object).</td>
</tr>
<tr>
<td></td>
<td>Advanced</td>
<td>Views or configures specific permissions, auditing, and object owner properties; allows inheritable permissions from parent to propagate to the Organization object. If cleared, prevents the Organization object from inheriting permissions from its parent.</td>
</tr>
</tbody>
</table>
Since the Security tab is not available by default on the Organization and Administrative Groups objects, enable the Security page on these objects by adding a new registry value:

```
HKEY_CURRENT_USER\Software\Microsoft\Exchange\EXAdmin\ShowSecurityPage=
dword:00000001 (enable) or 0 (disable)
```

The Top-Level Containers

The Organization object contains several top-level containers that hold Exchange 2000 system settings. The contents of these containers can change, depending on which display options you select in the Organization object properties. For example, if you display Administrative Groups, then the containers Servers, System Policies, and Connectors appear as child containers. Table 5-2 lists the top-level containers and their associated child containers.

<table>
<thead>
<tr>
<th>Container</th>
<th>Child Containers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Settings</td>
<td>Organization-wide Internet message formats, message delivery, and instant messaging properties</td>
</tr>
<tr>
<td>Recipients</td>
<td>Recipient policies, address lists, and address templates</td>
</tr>
<tr>
<td>Administrative Groups (not visible by default)</td>
<td>All administrative groups defined for the organization (each administrative group container displays containers representing the associated servers, policies, connectors, and folders)</td>
</tr>
<tr>
<td>Servers</td>
<td>All servers defined for the organization</td>
</tr>
<tr>
<td>System Policies</td>
<td>All defined mailbox store, public store, and server policies</td>
</tr>
<tr>
<td>Connectors</td>
<td>Simple Mail Transfer Protocol (SMTP), X.400, cc:Mail, MS Mail, Lotus Notes, GroupWise, and DirSynch connector objects (if you are viewing routing groups, you will also see connectors there)</td>
</tr>
<tr>
<td>Tools</td>
<td>Site replication services, message tracking, and server and connector monitoring</td>
</tr>
</tbody>
</table>

Table 5-2 refers to a recipient. A recipient is a user with a mailbox, a unique e-mail address, and the ability to receive e-mail.
Focusing on Administrative Groups

You can manage permissions easily by creating administrative groups. Figure 5-2 shows an organization with the administrative group First Administrative Group (organizations can have more than one administrative group). After creating an administrative group and setting its permissions, you can add objects to the group. These objects inherit the permissions you have set for the group.

![Figure 5-2: Administrative groups in the Exchange System Manager](image)

Design considerations

It is important to understand the design considerations for implementing administrative groups. In general, the administrative model and the size of the organization are the principal design considerations:

- **Single Administrative Group**: Exchange 2000 has a single administrative group called First Administrative Group. This administrative group is, by default, hidden from view in the Exchange System Manager. A single administrative group is typically sufficient for small to medium-sized companies.

- **Multiple Administrative Groups**: The main reason for creating additional administrative groups is to group servers for managing permissions.
The following two examples illustrate the use of administrative groups and the flexibility that they provide:

✦ A single central Information Technology (IT) group that maintains complete control over Exchange 2000 administration can use the single administrative group. This arrangement is typical of a small or medium-sized company. A large company that requires high-bandwidth connectivity to its regional offices can also employ this arrangement.

✦ Management of the Exchange 2000 organization can be distributed to company regions or divisions using multiple administrative groups. The central IT group manages standards and guidelines, but not the actual administration. In this arrangement, you should have at least one administrative group for every region or division. This is a very common arrangement in medium-sized and large companies with multiple divisions or geographically dispersed offices.

Creating additional administrative groups

The Administrative Group container is not displayed by default, so before you can create a new administrative group, you must first make the Administrative Groups container visible. To do this, access the Property dialog box of the Organization object, and select Display administrative groups.

Create a new administrative group by right-clicking on the Administrative Groups container, choosing New, and choosing Administrative Group.

If you have installed a new computer running Exchange 2000 and this is the first server, the Setup program automatically creates the First Administrative Group container, to which the server is added. If a single administrative group exists, then the server is automatically added to that administrative group. If multiple administrative groups exist, Setup will prompt you to select the administrative group to which the server should be added.

After you create an administrative group, you can add the following objects to it: system polices, routing groups, public folder trees, servers, conferencing services, and chat communities.

Using administrative groups

Administrative groups enable you to secure actions that users and groups can perform. You can grant permissions to allow or deny certain tasks to administrators. For example, you can specify which administrator can create and configure routing groups, servers, and polices.

Only some permissions can be granted through the Exchange System Manager. The remainder can be granted through the ADSI Edit utility. For example, granting the Create Policy Object permission can be done through ADSI Edit only.
Whether you grant permissions to administrators manually or by using the Exchange Administration Delegation Wizard at the organization level, the permissions automatically propagate to any new administrative groups.

You can also grant permissions on the Administrative Groups container; these permissions propagate to new child administrative groups. Use `Adsiedit.exe` to configure the Administrative Groups container under the Exchange 2000 Organization object in the configuration partition of the Active Directory.

**Creating System Policies**

A system policy, as shown in Figure 5-3, is a new object in Exchange 2000. It is a collection of configuration settings that can be applied to a mailbox store, a public store, or a server. You can define a single policy and apply it to a set of objects across the organization. When you need to modify an option setting, you can simply change the policy, affecting all servers that have that policy applied to them.

![Figure 5-3: System policies in the Exchange System Manager](image)

Using Exchange System Manager, you can create system policies in a System Policy container within an administrative group. When you select the type of policy to create, you are prompted to check the property pages that you want your policy to affect. The check boxes vary depending on the policy you are creating, but include the tabs that contain various configuration settings. Table 5-3 lists the tabs that you can add to the Property dialog box for each type of System Policy object.
### Table 5-3
**Tabs That Can Be Added to a System Policy Object Based on Its Type**

<table>
<thead>
<tr>
<th>Policy Object</th>
<th>Tab</th>
<th>Used To Configure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>General</td>
<td>Message tracking and log file maintenance</td>
</tr>
<tr>
<td>Public Store</td>
<td>General</td>
<td>General configuration settings</td>
</tr>
<tr>
<td></td>
<td>Database</td>
<td>Maintenance interval for the store</td>
</tr>
<tr>
<td></td>
<td>Replication</td>
<td>Replication interval and limits</td>
</tr>
<tr>
<td></td>
<td>Limits</td>
<td>Storage limits, age limits, and deletion settings</td>
</tr>
<tr>
<td></td>
<td>Full-Text Indexing</td>
<td>Update and rebuild intervals</td>
</tr>
<tr>
<td>Mailbox Store</td>
<td>General</td>
<td>General configuration settings</td>
</tr>
<tr>
<td></td>
<td>Database</td>
<td>Maintenance interval for the store</td>
</tr>
<tr>
<td></td>
<td>Limits</td>
<td>Storage limits and deletion settings</td>
</tr>
<tr>
<td></td>
<td>Full-Text Indexing</td>
<td>Update and rebuild intervals</td>
</tr>
</tbody>
</table>

### Applying a Policy

After creating a policy, you can apply it to relevant objects. To apply a policy, right-click on the Policy object, and add the required objects. Server policies can be applied to server objects, mailbox store policies can be applied to mailbox store objects, and public store policies can be applied to public stores. When you apply a policy, it takes effect immediately without requiring a logoff first.

**Note**
An object can have only one policy property sheet applied to it at a time.

### Managing Address Lists

Exchange 2000 users generally search for other users in their company in the Global Address List (GAL). The GAL, an aggregation of all messaging recipients, contains several address lists, each a subset of all users in the Exchange organization. Using custom address lists expedites the search for a particular user and enhances the security of user information by limiting access to an address list. Exchange 2000 uses three types of address lists.
Default address lists

Exchange 2000 includes several default address lists that you can use as they are or modify to suit the needs of your company. Figure 5-4 shows the GAL in the Exchange System Manager.

Table 5-4 lists the default address lists.

<table>
<thead>
<tr>
<th>Address Lists</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Contacts</td>
<td>This list comprises all mail-enabled contacts in the organization. Mail-enabled contacts appear in address lists but do not have an Exchange mailbox in the associated Exchange organization.</td>
</tr>
<tr>
<td>All Groups</td>
<td>This list comprises all mail-enabled groups in the organization. Mail-enabled groups appear in address lists and can be either distribution groups or security groups. A distribution group is a group of recipients that receive e-mail. A Security group is a group of items that have the same security profile.</td>
</tr>
</tbody>
</table>
Table 5-4 (continued)

<table>
<thead>
<tr>
<th>Address Lists</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Users</td>
<td>This list comprises all mailbox-enabled and mail-enabled users in the organization. Mailbox-enabled users have Exchange mailboxes. Mail-enabled users appear in address lists but do not have Exchange mailboxes.</td>
</tr>
<tr>
<td>Public Folders</td>
<td>This list comprises all mail-enabled public folders in the organization that are not hidden from address lists. Mail-enabled public folders appear in address lists and can receive messages.</td>
</tr>
<tr>
<td>Default Global Address List</td>
<td>This list comprises all recipients in the organization. In other words, this list comprises mailbox-enabled users, mail-enabled users, groups, contacts, and public folders.</td>
</tr>
</tbody>
</table>

Custom address lists

In Exchange 2000, you can create custom address lists based on fields of Recipient objects. (The Recipient object is displayed in the To: field when you create an e-mail message.) For example, using the Department field you can create an address list that contains only users in the Accounting department.

Offline address lists

You can also create an offline, local copy of address lists. The local offline address book (whose files must have an .oab extension) enables your users to resolve many names as they compose messages while not connected to the Exchange server. They can also search the content of the offline address lists and address the messages. Next time they are online, the messages created earlier are sent automatically.

Since the GAL is the most comprehensive of all address lists, you should use it as the source for an offline address list (unless the GAL is very large). All other address lists are typically subsets of the GAL. Use subsets only when the GAL contains the following:

- Names that are not useful (for example, if a certain division never communicates with another division)
- Names that are not appropriate for inclusion in the Offline Address List (that is, names that should not be available to everyone)
- A quantity of names too cumbersome for offline use (that is, such a large number of names that downloading the offline address book would take too long over slow links)
Creating an address list

You can create custom address lists in the All Address Lists container or All Global Address Lists container in Exchange System Manager. Right-clicking the container object and choosing the New option enables you to create a new custom address, as shown in Figure 5-5. After you specify a name for the new list, click the Filter Rules button. You can restrict membership to the address list by using the General tab. You can specify the server on which the recipient’s mailbox may be stored in the Storage tab. To restrict membership, use the Advanced tab to create field-based filter rules.

![Figure 5-5: Creating a new address list in the Exchange System Manager](image)

You can select the membership of an address list from the following list:

- **Users with Exchange Mailbox**: All mailbox-enabled users
- **Users with external e-mail addresses**: All mail-enabled users
- **Groups**: All mail-enabled groups
- **Contacts**: All mail-enabled contacts
- **Public folders**: All mail-enabled public folders not hidden from address lists

You can limit address-list membership based on storage location:

- **Mailboxes on any server**: Recipients on all servers
- **Mailboxes on this server**: Recipients with mailboxes on the specified server
✦ Mailboxes in this mailbox store: Recipients with mailboxes on the specified store

Defining a filter rule

A filter rule helps you fine-tune membership to an address list. Defining multiple filter rules for an address list defines specific membership requirements for that address list. All the filter rules are combined together to form a single Lightweight Directory Access Protocol (LDAP) query; all the conditions of the query must be met in order for a recipient to be a member of the address list.

You can define a filter rule by creating a new address list and choosing filter rules, as shown in Figure 5-6.

![Figure 5-6: Creating a filter rule](image)

Defining a field, a condition, and a value creates a filter rule. For example, to create an address that contains all users in the Accounting department, select the Department field for users, select the Is exactly (the word exactly is optional) condition, and specify the value as Accounting. Table 5-5 lists the conditions that are available for building filter rules.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starts with</td>
<td>The value for the selected field must start with the characters specified in the Value box.</td>
</tr>
<tr>
<td>Ends with</td>
<td>The value for the selected field must end with the characters specified in the Value box.</td>
</tr>
<tr>
<td>Is (exactly)</td>
<td>The value for the selected field must exactly match the contents of the Value box. The match is case-insensitive.</td>
</tr>
</tbody>
</table>
### Condition | Description
--- | ---
Is not | The value for the selected field must not match the contents of the Value box.
Present | The selected field must contain a certain value.
Not present | The selected field must not contain a certain value.

The value you specify in the Find Exchange Recipients dialog box is compared against the value in the Active Directory. Valid recipients are added to the address list.

#### Using a custom search
A custom search is invoked by directly typing the LDAP query code, instead of building a query by choosing fields and specifying conditions and values. The search results are added to the custom address list. Custom searches make it easier to determine address-list membership.

With custom searches, you can include logical operators in the query. For example, in the query

```
(&((&(firstname="Bob")(|(department=Accounting)
(department=Finance))))
```

the custom search is querying for a recipient that has a first name of “Bob,” and the department field may contain either the value Marketing or the value Finance.

To perform a custom search, open the Property dialog box for the address list on which you want to perform the search; enter the LDAP query in the Find Custom Search dialog box.

#### Securing address lists
You can secure an address list to control which recipients users or groups can view. Using the Exchange System Manager, configure the Open Address List permission on the Security page of the address list you need to secure.

You can specify the permission for each address list individually; alternatively, you can specify the permission for all address lists in the container by applying permissions on the All Global Address Lists or All Address Lists containers.

#### Updating an address list
Exchange 2000 includes a Recipient Update Service (RUS), the part of the System Attendant service responsible for building and maintaining address lists.
RUS runs as a thread of the System Attendant and polls the Active Directory on a predetermined schedule (default is once a minute) for updated recipient information. If there are any new recipients, new address lists, or changes to existing address lists, RUS updates the address lists.

You can update an address list manually by right-clicking on the address list and choosing either Update Now or Rebuild. Update Now will update the address lists with changes in the recipient information. Rebuild will rebuild the entire membership of the address lists.

**Recipients**

Exchange 2000 recipients include Exchange users, contacts, and groups. You can administer Exchange 2000 recipients through the Active Directory by using the Microsoft Management Console (MMC) snap-in Active Directory Users and Computers.

**Creating a Recipient object**

When you create a user in Active Directory, you can enable the user to send or receive e-mail by making the user a recipient. Users, contacts, or groups are called Recipient objects.

There are three types of Recipient objects in Exchange 2000.

- **User:** A user created in Active Directory can send or receive mail by being mailbox-enabled or mail-enabled. A mailbox-enabled user has an Exchange 2000 mailbox and an e-mail address and thus can send and receive e-mail messages. A mail-enabled user has a Windows 2000 authentication account and an external e-mail address associated with it, but no Exchange mailbox. A mail-enabled user is listed in the address list, though, enabling other users to easily locate and send mail to the user, even if the user does not have a mailbox in the Exchange 2000 organization.

- **Contact:** A mail-enabled contact is a user that has neither a Windows 2000 authentication account nor an Exchange mailbox in the associated Exchange organization. Mail-enabled contacts are visible in the directory, but receive their mail from a foreign system. Internal users can easily send them messages by selecting them from the appropriate address list.

- **Group:** A mail-enabled group can be either a distribution group or a security group. Once a group is mail-enabled, it will appear in address lists and can receive messages. These messages will be sent to the members of the group that have an e-mail address.

To create a mailbox-enabled user, simply create a mailbox for the user. You can create a mailbox when you are creating a new user in the Active Directory, as shown in Figure 5-7.
You can also create a mailbox for a user later by using the Exchange Tasks Wizard, as shown in Figure 5-8. Invoke the Exchange Task Wizard by right-clicking on a user in the Active Directory Users and Computers snap-in and choosing Exchange Tasks... The Exchange Tasks Wizard prompts you to specify the server, storage group, and store. The wizard then creates the mailbox in the specified store. Using the Exchange Tasks Wizard, you can establish e-mail addresses for users, contacts, and groups.

When you create a mailbox for a user, several new tabs appear in the Properties dialog box for the User object. Use these tabs to configure various Exchange 2000 settings, such as delivery restrictions and delivery options. A more detailed discussion can be found in Chapter 6.
Configuring a Recipient object

Once you create a mailbox for a user or mail-enable users, contacts, or groups, several tabs appear in the Properties dialog box of the Recipient object. You can configure the recipient by selecting the various configuration options available in these tabs. The tabs with Exchange 2000–specific options are General, Organization, Exchange Advanced, Exchange General, E-mail Addresses, and Exchange Features.

Administering Public Folders

This section explains the features and architectural elements of public folders in Microsoft Exchange 2000. A public folder is a repository for many types of information that can be shared among users in an Exchange organization. You create and manage public folders by using either client software such as Outlook 2000 or the Exchange System Manager. Outlook 2000 provides for basic configuration, while the Exchange System Manager enables in-depth control over the behavior of the folder.

Public folders in Exchange 2000 include the following features:

✦ Multiple public folder trees, also known as hierarchies, enable you to store public folders in more than one tree.
✦ Implementation of the Information Store in Exchange 2000, coupled with the Active Directory, enables you to send messages to a public folder, using the mail object entries for the public folder stored in Active Directory. This method enables you to place an object in a public folder without posting it directly.
✦ Exchange Installable File System (EXIFS) enables you to secure items in public folders.
✦ Accessibility from the Web enables you to use a Web browser to gain access to public folders by specifying a URL to the folder.
✦ Accessibility from the file system enables you to use EXIFS to share public folders.
✦ Full-text indexing capabilities for public folders (provided by MSSearch) enables Microsoft Outlook clients to use this index automatically when performing a Find or Advanced Find.
✦ Public folder referrals enable clients to gain access to any folder in the organization. Referrals are enabled by default between routing groups.
✦ Multiple public folder trees are supported for multiple collaboration applications.
Creating a public folder

To create a public folder using the Exchange System Manager, go to Organization ➤ Administrative Groups ➤ First Administrative Group ➤ Folders, and right-click Public Folders. Choose New, then choose Public Folder, and enter a name in the Name text box, as shown in Figure 5-9.

![Image: Creating a public folder with the Exchange System Manager](image)

Figure 5-9: Creating a public folder with the Exchange System Manager

The Exchange System Manager enables you to administer public folders. You can view all available public folder trees within an administrative group and the folders contained in each tree. You can create and configure folders, and you can mail-enable a public folder, which adds the folder to the Active Directory. You can also configure the security settings for a public folder or public folder tree root and propagate them down the hierarchy.

Configuring a public folder tree

To configure a public folder tree, you must configure the public store on each server that hosts content from that tree.
You cannot split a public folder tree across multiple stores. Be sure the store you choose has sufficient space to hold the expected growth of the public folder.

The Public Store Properties dialog box enables you to configure the settings for a public store. These settings include such items as replication schedule, age, and storage limits. The public store provides sub-containers with information about users, such as who is logged on and using the public folders; and public folder details, such as the name, path, size, and number of items for all public folders in the store.

You can set up a public folder in a public store to appear as a mail recipient in the Active Directory. To accomplish this, mail-enable a public folder by right-clicking it, choosing All Tasks, and clicking Mail Enable. After you mail-enable a public folder, the System Attendant connects to Active Directory and creates an object for the public folder in the Microsoft Exchange System Objects container. You can also configure the folder to appear in the GAL for clients, such as in Outlook 2000.

**Public folder permissions**

When you create a public folder, Exchange 2000 assigns a set of permissions that specify individuals with the right to perform designated activity in that folder. You can assign permissions to folders, items, and properties.

Exchange 2000 relies on the Active Directory to enforce security on Exchange 2000 resources. The operating system manages and enforces permissions that are specific to Exchange 2000, such as the ability to create a top-level public folder.

The top-level folder in a public folder tree is referred to as a parent folder. You can assign permissions to a parent folder and then propagate the permissions to all folders in that tree. You can configure permissions for a public folder tree by opening the Properties dialog box for the tree and selecting the Security tab, as shown in Figure 5-10. The permissions for public folders in Exchange 2000 are divided into four categories:

- **Folder Rights:** Enables you to control the permissions of users accessing the folder
- **Message Rights:** Enables you to decide which users can gain access to messages sent to a mail-enabled public folder
- **Directory Rights:** Enables you to control which users can manipulate the object that mail-enabling a public folder creates in Active Directory
- **Administrators Rights:** Enables you to assign specific rights to specific administrators
Table 5-6 describes the permissions that are available in Exchange 2000.

<table>
<thead>
<tr>
<th>Permission</th>
<th>Folder</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Public Folder</td>
<td>Organization, administrative group,</td>
<td>Administrative permission to create a public folder (overrides other</td>
</tr>
<tr>
<td></td>
<td>public folder tree</td>
<td>permissions, such as Create Container)</td>
</tr>
<tr>
<td>Create Top Level</td>
<td>Organization, administrative group,</td>
<td>Specifies who can define top-level folders, which in turn define the tree</td>
</tr>
<tr>
<td>Public Folder</td>
<td>public folder tree</td>
<td>structure</td>
</tr>
<tr>
<td>Modify Public Folder ACL</td>
<td>Organization, administrative group,</td>
<td>Specifies who can configure administrator rights</td>
</tr>
<tr>
<td>Modify Public Folder Admin</td>
<td>public folder tree,</td>
<td></td>
</tr>
<tr>
<td>ACL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Continued
Table 5-6 (continued)

<table>
<thead>
<tr>
<th>Permission</th>
<th>Folder</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Named Properties in the information store</td>
<td>Create Named Properties in the information store</td>
<td>Specifies who can create named properties; (use this to prevent unauthorized properties from being created, possibly by a denial-of-service attempt)</td>
</tr>
<tr>
<td>Modify Public Folder Deleted Item Retention Public Folder Expiration Public Folder Quotas</td>
<td>Modify Public Folder Deleted Item Retention Public Folder Expiration Public Folder Quotas</td>
<td>Specifies who can change these configuration properties</td>
</tr>
<tr>
<td>Modify Public Folder Replica List</td>
<td>Modify Public Folder Replica List</td>
<td>Specifies who can configure where a public folder is replicated</td>
</tr>
<tr>
<td>Administrator Information Store</td>
<td>Administrator Information Store</td>
<td>Specifies who can manage the public information store</td>
</tr>
<tr>
<td>View Information Store Status</td>
<td>View Information Store Status</td>
<td>Specifies who can view status on the public information store</td>
</tr>
</tbody>
</table>

Public folder replication

The replication of public folder components is controlled by different services: the Active Directory controls the replication of public folder directory objects, the Exchange store controls the replication of public folder hierarchies, and the Exchange Administrator controls the replication of the content of public folders.

You can configure a public folder to have replicas on multiple public folder servers. Replicas are useful for distributing the user load on servers, distributing public folders geographically, and backing up public folder data. A replica copied from one server to another is a separate instance of a public folder and its contents. All replicas of a public folder are equal. There is no master replica.

You can create a public folder replica by configuring the properties of a public folder. You can use the Replication tab to add or remove from the list of public stores that receive a replica of the folder. Replicating to a list is called push replication.

Caution

When you push a replica on another server, be sure there is sufficient space. It is considered bad form to push replicas without agreement from the Administrators at the recipient sites.

You can also configure the public folder instances for a server. From the Public Folder Instances object under the public information store in MMC, choose All Tasks–Add Replica to send the replica of any public folder to your server. This is called pull replication.
Be sensitive to the effect on bandwidth of pushing and pulling replicas. You may want to schedule these activities at night, to avoid bogging down your lines, servers, and regular messaging traffic.

Public folder referrals
Public folder referrals enable you to route information and requests to specific folders. You can enable public folder referrals to servers in another routing group by implementing and configuring a routing group connector between two routing groups.

Using Routing Groups and Connectors
Many organizations need multiple servers running Exchange 2000. This arrangement means that when a user on one server wants to send a message to a user on another server, the message must be transferred between servers. Transferring messages between servers is called message routing.

When the first Exchange 2000 server is installed, a routing group is automatically created and the server is installed into this routing group. Unless you create a new routing group, all subsequent Exchange 2000 servers also are installed into this routing group. Simple Mail Transfer Protocol (SMTP) is used for transferring messages between two servers running Exchange 2000 in the same routing group.

Message transfer between servers in different routing groups is more complex. Connectors are used to transfer messages between routing groups. There can be multiple connectors between two routing groups. In organizations with more than two routing groups, there can be multiple hops between routing groups, depending on how the routing groups are connected.

Creating a routing group
Once a server has been installed into an administrative group and the default routing group has been created, additional routing groups can be created to suit your Exchange 2000 topology. Exchange 2000 servers can then be installed into or moved into their designated routing group.

Creating a connector
Creating multiple routing groups is only half of the task of implementing your Exchange 2000 routing topology. Once routing groups have been created, they need to be connected using connectors.

You can use any of the following connectors to connect two routing groups:
✦ Routing group connector
✦ SMTP connector
✦ X.400 connector

The Routing Group connector can connect routing groups only. The other connectors can also provide connections to foreign systems.

The SMTP connector should be used to connect routing groups when the other side of the connector is the Internet Mail Service (IMS) from a previous version of Exchange.

You can use an X.400 connector to establish an X.400 messaging route between two routing groups or between a routing group and an X.400 system. An X.400 messaging route defines the path that an X.400 message follows to reach its final destination.

**Chat Communities and Conferencing**

With the Exchange System Manager, as shown in Figure 5-11, you can also manage and configure chat communities and conferencing servers. These topics will be covered in detail in subsequent chapters.

![Figure 5-11: Configuring a Chat Community](image)
Managing Administrative Security

Using Exchange System Manager, you can grant administrative privileges in Exchange 2000 by giving Windows 2000 users and groups permissions to Exchange objects. Granting these permissions makes administration more secure, since you can specify who can access which Exchange 2000 objects.

Exchange 2000 uses the security model of Windows 2000 and the Active Directory to manage access to objects. All Exchange 2000 objects are secured with a discretionary access control list (DACL) and individual Access Control Entries (ACEs) that give users and groups specific permissions to control administrative access to an object.

Exchange 2000 allows two types of permissions: standard and extended. Standard permissions are part of the default permissions for the Active Directory. Extended permission, as shown in Figure 5-12, is added when Exchange 2000 is installed.

Figure 5-12: Extended permissions
Permission inheritance

A child object inherits permissions from its parent object by default. Permission inheritance eliminates the need to apply permissions to child objects manually. Permissions can be applied to all child objects by simply applying the permissions to the parent object, ensuring consistent permissions among all child objects.

Exchange Administration Delegation Wizard

Administrators must be granted the necessary permissions to perform their administrative tasks in Exchange 2000. You can grant these permissions using an automated wizard called the Exchange Administration Delegation Wizard.

The Exchange Administration Delegation Wizard enables you to select a user or a group and give them one of three specific administrative roles:

- **Exchange Full Administrator**: Can fully administer Exchange 2000 system information and modify permissions

  This role is extremely powerful. This access level should only be granted to people who fully understand Exchange 2000 and are aware of the inherent dangers involved in making incorrect modifications to the system.

- **Exchange Administrator**: Can fully administer Exchange 2000 system information (this role is for users or groups responsible for the day-to-day administration of Exchange 2000)

- **Exchange View Only Administrator**: Can view Exchange 2000 configuration information (this role is for Administrators who need to view organization information of other administrative groups that they are not administering)

Summary

This chapter provides an in-depth look at the Exchange System Manager — what it is, organization objects, administrative groups, system policies, address lists, recipients, public folders, and routing groups and connectors. The chapter also presents a list of available Exchange 2000 monitoring tools — some of which are accessible from the Exchange System Manager — as well as the administration capabilities of the Exchange System Manager for chat communities and conferencing servers. Finally, the chapter explains how to manage the administrative security of Exchange 2000.
In this chapter, we discuss some of the basics of the Windows 2000 Active Directory Service (ADS) and its role in providing the enterprise directory components necessary to support Exchange 2000. Except to provide some grounding, what we focus on are the parts of Windows 2000 that are germane to administrating, planning, and developing on Exchange 2000. The goal is to provide a sufficiently detailed overview of the Windows 2000 ADS and its relevance in the Exchange 2000 environment to provide that grounding. This material is complemented by much of the material on planning and architecture in Part IV of this book. We strongly encourage you to invest some time learning Windows 2000 beyond the information presented here, because as almost anyone knows, a good craftsman or builder has a sound preparation in foundations. Windows 2000, especially Active Directory, is the foundation of Exchange 2000, and both have changed quite a bit from NT 4 and Exchange 5.

For more information on structure, administration, development, and support of the Active Directory Service, pick up a copy of the Active Directory Bible, Windows 2000 Programming Bible, or Windows 2000 Server Bible, all published by Hungry Minds, Inc.

Windows 2000

Using Windows NT and Exchange 5.5, administrators are tasked with maintaining two user accounts: one in the Windows NT Security Accounts Manager (SAM) and the other in the Exchange Directory. All administrative work associated with planning and implementation, backups, maintenance utilities, data entry, and troubleshooting is doubled when you have two directories. Besides that, administrators have to implement and support two separate directory replication topologies, which can be quite a challenge.
Other issues that arise from the Windows NT world are the restrictions placed upon designers by the limitations of the Windows NT domain model. Windows NT Trust relationships are not transitive, so domain topologies are not very flexible and end up being relatively flat. Rigid domain structures make setting up and delegating administrative permissions extremely difficult if you want more than a simple model.

Windows 2000 was designed to address many of these shortcomings. It is the next step in the evolution of the Microsoft enterprise-level network operating system and provides the core directory services required to support Exchange 2000. Windows 2000 provides

- A complete Lightweight Directory Access Protocol (LDAP) compliant directory service called Active Directory Service (ADS)
- Windows Management tools
- Kerberos security
- Public-Key Infrastructure (PKI)
- COM+
- Integrated Terminal Services
- Enhanced Symmetric Multiple Processor (SMP) support
- Integrated TCP/IP Load Balancing and Clustering support
- Integration of common services, such as DHCP and DNS
- Other Internet services

All the features and services of Windows 2000 integrate at a much more fundamental level than they did in previous versions of the application. Many of these changes to Windows 2000 were made with other third-party and Microsoft BackOffice applications, such as Exchange, in mind.

With the advent of Windows 2000 and Active Directory, companies now have a chance to structure their Windows and messaging environments to match their organizational structure, as well as integrate Windows with other operating systems, networks, and applications. The opportunity to finally create an integrated network and computing platform for an entire organization is here. IT organizations can combine the many disparate systems that support their current networks and applications and integrate them to create networks and applications to support the organization now and into the future. We will now take a look at the technologies that make up Active Directory, as well as discuss the impact of these technologies on the Exchange 2000 environment.
Active Directory

In the past, Windows NT provided a directory service called SAM (Security Accounts Manager) that contained users, groups, and computer accounts. The SAM directory and its interface were limited in functionality and difficult to extend or modify. In order for applications such as Exchange 5.5 and SQL server to interoperate with the SAM, a separate directory service was required to support the application.

Windows 2000 overcomes these limitations by using Active Directory and incorporating into it some of the concepts you became accustomed to in the Exchange directory (back fill, GUIDs, LDAP access, and so on). Active Directory is a directory service that is built on and surpasses the flexibility, scalability, and utility built into the Exchange 4 and 5 Directory. Active Directory is a Lightweight Directory Access Protocol (LDAP) and Message API (MAPI) compliant directory service. It provides distributed access to objects for users and services from anywhere in the organization via Global Catalog servers. Global Catalog servers and their importance will be discussed in detail later in this chapter.

Active Directory is, in simple terms, a distributed multi-mastered replicated database system. To provide a database engine for Active Directory, Microsoft uses an implementation of the Joint Engine Technology (JET) database engine, which it calls the Extensible Storage Engine (ESE). The ESE is a transaction-based database engine that commits changes using log files to provide greater reliability and recoverability in the event of systems failure.

You may recall that Exchange 5 had its directory built on the ESE design, but ADS is far more sophisticated—even though its origin, and certain aspects of its functionality, are the same as what you are familiar with in earlier versions of Exchange.

For example, size matters in a directory. Size limitations on the Exchange Directory were perceived as a drawback by some large organizations. The ESE database engine enables the Active Directory database to grow to an enormous size with a limit of 70 Terabytes, thus being capable of storing millions of objects. These objects are any and every object that is contained within the Windows environment, such as user, computers, groups, and domains. Each object consists of a set of attributes that defines the object in the directory service—for example, a user object that has attributes such as first name, last name, user name, and password.

The ability to store countless objects and attributes, coupled with the ability to extend the directory with new objects and attributes, gives Active Directory extreme versatility. This versatility is intentional and enables Active Directory integrated applications, such as Exchange 2000, to leverage the directory service as well as the replication mechanisms that are utilized to replicate objects, attributes, and changes amongst all domain controllers. Unlike Exchange 4 and 5, this directory replicates at an attribute level.
One way that Exchange 2000 leverages the ADS is that objects, such as the user object, are extended with attributes that support Exchange 2000, such as multiple e-mail addresses. By default, the user object can support one e-mail address per user, but with the extensions to the directory by Exchange 2000, many addresses can now be supported. In total, Exchange 2000 extends the Active Directory with approximately 155 additional object classes and 818 additional attributes. This effectively doubles the number of objects that exist by default, thus enforcing the need for a sound design and implementation of the underlying Windows 2000 infrastructure.

Chapter 3 discusses two standards, LDAP and X.500. These two standards are essential to the existence of Exchange and enable the use of the Directory Information Tree (DIT) architecture to provide directory support to Exchange. Exchange 5.5 was based upon proven and reliable DIT architecture that Microsoft calls the Directory Store. The Directory Store is an X.500-compliant directory schema that utilizes the LDAP protocol to provide both read and write access. Exchange 2000, as mentioned earlier, leverages the Active Directory and uses it to replace its own Directory. In a sense, ADS is the next iteration of Exchange’s Directory Store or DIT architecture—even though it is now owned by Windows 2000 and used by all of Microsoft BackOffice and many other third-party applications.

During the planning and development processes of Windows 2000 and Exchange 2000, Microsoft integrated many of the lessons learned from their past directory efforts and strove to provide products that were granular, robust, scalable, and flexible with a lightweight interface. By leveraging the AD, Exchange 2000 benefits from the enhanced replication architecture that is present in Windows 2000. Windows 2000 forest, trees, domains, and sites automatically define the actual replication architecture, but it is flexible so you can customize it if required. By customizing the configuration and placement of Active Directory services, Global Catalog servers, domain controllers, and DNS services, you can create a very efficient replication topology for the network that is right for your organizational and geographic topology. When done properly, it is able to recover from certain system or network failures.

Windows 2000 and Active Directory Components

To understand what makes Windows 2000 and ADS work together, you need to understand how certain industry standards were implemented and how they fit together to achieve design goals.

DNS

The topology and implementation of the Domain Name System (DNS) is one of the primary considerations in any Windows 2000 architecture. Exchange 2000 relies on DNS. DNS provides forward and reverse host to IP name resolution, as well as
service location services. When DNS is implemented incorrectly or malfunctioning, both Windows 2000 and Exchange 2000 functionality suffers. Proper planning and implementation of DNS is the key to providing namespace framework.

In the world of Windows NT 4.0, the internal DNS namespace was less important for several reasons. Instead of using DNS, Windows-based machines and most Windows-based applications relied heavily on NetBIOS to identify network names and services. In order to use the IP protocol, a service or mechanism was needed to provide resolution of NetBIOS names and services to IP. Some organizations used a Microsoft service called Windows Internet Name Service (WINS), and others used LMHOST files. Both introduced their own administrative issues and conundrums, especially in large environments. Since most internal applications used NetBIOS, DNS was used primarily for external resources such as Web sites or UNIX systems. NETBIOS had many drawbacks, and today, serious efforts are made to completely eliminate it in pure Windows 2000/Exchange 2000 environments.

For more information on DNS, see Chapter 3.

DNS is a client-server distributed database application. Since the database is distributed, administration of the namespace information can be delegated to the appropriate segments. To achieve the distributed database hierarchy or structure of the Domain Name System, it was modeled after the common Unix file system and is basically an inverted tree. Figure 6-1 shows the DNS database structure and how it relates at a low level to the UNIX file system.

![Figure 6-1: Unix and DNS database structures](image)
The DNS hierarchy and the Unix file system are similar in some ways. For example, they both use the concept of a root. The DNS hierarchy root is referred to using a single period: “.” The model resembles a tree. Limbs below the root of the DNS tree are called subtrees and are used to logically partition the database. As in most hierarchical structures, each junction or node acts as the root of yet another subtree that exists below it. In the world of DNS, these subtrees are called domains. Each domain or subtree can be further divided into more subtrees, commonly called subdomains.

The definition of each node in the database correlates to its position within that DNS structure. When all node names are concatenated together, you have a node’s Fully Qualified Domain Name (FQDN). In Figure 6-1, for example, the Sales node is the bottom of the hierarchy and thus its FQDN is “sales.ilmarin.com.” and can be either a device or the root of yet another subtree. In the event that this FQDN refers to the beginning of another subtree, a computer called “box” below it would have a FQDN of “box.sales.ilmarin.com.”

One of the most striking advantages of this type of structure is that it only requires unique node names at the node level. In Figure 6-1, for example, a system below the Sales node could be also be called “sales,” making the FQDN for this host “sales.ilmarin.com.” While a bit strange, this example is fully acceptable and unique.

Imagine how many Jian Chens there are in China, or how many John Smiths there are in the U.S. Duplicate user names are a problem in most large organizations. Telling users to alter their names is not an option.

One way to help avoid confusion and complexity from relying on FQDNs is to use shorthand. DNS can be thought of as an index of pointers. To make dealing with the pointers easier, it is possible to give machines aliases, usually called canonical names. In our example, imagine an alias created within DNS called “ebox.” “Ebox” is actually a pointer to the host system call “box.” This kind of flexibility helped solve major problems with the original HOSTS file (the problem revolved around name collisions within the file) and made usage of the system easier.

The subtree model allows for delegation of authority; it allows for management of the subtree by the party using or owning it. In the case of “ilmarin.com.” the subtree could be set up to be managed by Ilmarin employees. This has distinct advantages over a system where Ilmarin administrators have to submit changes and wait for the compiled HOSTS file to be republished, after which point they begin the task of updating all their internal systems.

DNS lexicon includes the term zones. Zones are equivalent to domains. Each zone and domain database contains a complete index of all known systems and subdomains within that level of the domain. Zone databases reside on whatever name servers act as authorities for that domain or zone. These names servers are said to “hold authority” over one or many zones. Since zones and domains are analogous, why refer to them differently? The usefulness of zones is easily understood when
you look at delegation of management and administrative responsibilities. By using zones or domains, they inherit the responsibilities delegated to them from the parent domain or zone, making propagation and management of the correct responsibilities easier.

As mentioned earlier, DNS can do more than locate nodes. It also provides the ability to locate services by using Resource Records (RR). Resource Records provide the ability to query a DNS server looking for the location of services and resources within a domain. For instance, one type of RR is the MX or Mail Exchanger record. MX records are regularly entered into DNS by domain Administrators to provide the location of domain e-mail servers that will accept incoming mail connections. An external mail server that is attempting to send mail to salesuser@Ilmarin.com would usually query its local DNS server for the address of this mail server. The local server will probably have to forward its request up the tree to resolve the address if it does not contain domain information for the “Ilmarin.com” domain (which it probably wouldn’t, unless it was part of the Ilmarin.com domain or was in the cache from a previous request). The request then works its way up the tree DNS server to DNS server and then down to the servers which handle Ilmarin’s domain. The Name Servers handling Ilmarin.com respond to the query with a return of the names and IP addresses for servers handling incoming mail for that domain. This information works its way back to the mail server originating the request, at which point it establishes a connection to the servers specified in the return of the query for the Ilmarin domain.

In the previous example, queries located a service within a domain. Queries can also be used to perform hostname to IP address resolution using either recursive or iterative models. Recursive queries occur when the Name Server

1. Receives the query
2. Checks its zones for which it is authoritative (and then)
3. Responds back to the query with the information if it has it

But what happens if the name server is not authoritative for the query and the server does not have the information? In the DNS model, if the name server is not authoritative for that zone, it will still query up the tree until at last it finds a server that is authoritative for the zone and can procure a response. Therefore, it needs to know about other servers that are farther up the tree and often will have more information than it has. Recursive queries provide a better method of resolving the IP address from the client perspective (because the client hands off all the work), but are a substantial burden on the name server in regards to performance and processing load (because the server has to do all the work).

Iterative queries are a bit easier on the local name servers. They respond with their best guess for a server that would have the information in situations where they are not authoritative for that zone. In other words, if a name server were asked for a host that it contained within its zone, it would respond with the answer to the
query or would provide the closest name server that it has knowledge of that might possibly know the answer. This method of operation can be further enhanced by the use of caching DNS servers. Caching means these servers keep a list of recently requested name servers in memory, which provides faster response for local name servers when they are attempting to procure information about non-zone systems and domains that happen to be in the cache already. Because people tend to work in the same patterns, it is highly likely that many hits will go to cache rather than out to the Internet for resolution. Caching servers may be quite sophisticated. They can contain an index of servers spoken to, domains they are authoritative for and domains they are not authoritative for. Each can improve speed. Employing all of them can improve speed even more.

Forests, trees, and domains

As mentioned previously in this chapter, Windows 2000 has many new concepts to reshape the way organizations structure their Window NT and 2000 environments. Two of these concepts (new to the Microsoft Windows platform, but not necessarily to the world of Network Operating Systems) will be discussed in this section: forest and trees. This section will also cover the role and definition of a domain in the Windows 2000 realm and how its changes will impact organizations.

Windows NT provided a rudimentary system of creating security boundaries using domains, but lacked a mechanism to help tie it all together to mimic the needs of a large organization that requires a complex security boundary structure. Many organizations were forced to endure the frustrating reality of managing the many domains that organically existed within their company while attempting to maintain a logical hierarchy and management structure. This often proved to be a daunting task and translated to greater levels of difficulty when considering applications such as Exchange 5.5, Exchange 2000, and upgrades to Windows 2000.

To solve the inherent shortcomings of the Windows NT domain model, Microsoft introduced forests, trees, and a slightly different version of domains. A forest provides a method of defining an organizational boundary at the highest level but does not require a one-to-one relationship with the DNS namespace. A forest is a collection of trees that facilitates replication between them. In fact, a forest is merely a container for domains. All domains in a forest share a single schema, global catalog object/attribute set, and configuration. All domains have transitive Kerberos trusts between one another.

Caution

Since the forest is the largest organizational structure in Windows 2000, containing a single schema and global catalog, only one Exchange organization can be hosted per forest. This does not mean that you cannot host Exchange services for more than one company. With the additional flexibility that Active Directory and Exchange
Windows 2000 allows, you can easily create multiple Global Address Lists, multiple public folder trees, and give users completely customized e-mail addresses. In fact, one of the design goals for Exchange was to make it flexible enough to be used by application service providers for e-mail and collaboration service hosting of many companies.

One major benefit of a forest is that it provides a system for allowing the automatic creation of bi-directional trusts between all of the domains with it. This trust model is automatic and establishes a hierarchy within an organizational boundary to serve the purpose of object access and authentication throughout the enterprise. This is what allows for the global access of user, group, and computer objects from any domain within it. This automation also significantly reduces the overall administration required to maintain an infrastructure at a corporate level, thus reducing the overall Total Cost of Ownership (TCO).

A forest is automatically created during the creation of the first Windows 2000 active directory domain in the enterprise. This first domain is called a “Forest Root Domain” and contains several properties that are unique to the forest. For this reason, a Forest Root Domain can never be removed from the enterprise, and it is not feasible to rename a forest. Forest Root Domains are the foundation for all other domains created subsequently. The Forest Root Domain contains two enterprise-wide Flexible, Single Master Operation (FSMO) roles — the Schema Master and the Infrastructure Master — which will be discussed in further detail in the FSMO Roles section. The root also contains the Schema Admins and Enterprise Admins groups. These two groups essentially have forest-wide control, including every object in every domain, so it is critical to make sure that the original Forest Root Domain is planned properly.

Caution

When planning a Windows NT migration to Windows 2000, it is imperative that the first Windows 2000 domain is the one that will be performing forest-wide operations and schema changes. Normally, this will be the corporate administrative domain.

Trees are a domain hierarchy that share a contiguous DNS namespace. In basic terms, a tree would be defined at the second level in a hierarchical structure. In the realm of Windows 2000, though, this is slightly different due to the fact that the first level in the hierarchy defines the forest. This means that even though the forest is defined at the first level, the namespace really begins at the second level. This enables organizations with many different divisions and corporate identities to create a single structure for all domains. That structure can include all domains below without being tied to a single namespace. All subsidiary parts of their organizations can readily be included. Figure 6-2 illustrates this issue.
Domains in Windows 2000 have a slightly more flexible definition than in the NT environment. In NT, domains provided a rigid security environment. It was also hard to achieve flexibility among many domains and to provide a security model that conformed to the needs of the organization, due to the way trusts worked. In Windows 2000, the definition has been reworked so that domains still provide rigid security boundaries yet act more like a container or component to the overall Active Directory security model.

Organizational units and groups

One of the main shortfalls of the Windows NT model is the way directory objects are logically grouped and administered. Users and groups are organized into other groups, and users, groups, and computers are arranged into domains. This logical organization works and still exists in Windows 2000. But what if I want to create departmental groups that can be administered by a department manager? I can
organize these users into a group, but there’s no way to give that department manager full control over the group without giving him control over other accounts in the domain. Microsoft addresses this issue by introducing the Organizational Unit (OU).

Essentially, an OU is a container for objects in the Active Directory. You can place user accounts, groups, contacts, and computers into this container. By themselves, OUs don’t do anything except logically organize other objects. However, when you add group policies and permissions delegation, they get much more interesting. Organizational Units are typically created for two reasons:

✦ **Delegation of administrative permissions:** Remember the department manager that needed full control over the people in his department? That scenario is easily accomplished by creating an OU and placing his users in it. Once the users are in the OU, simply run the Delegation wizard and assign full control to the OU manager. Think how difficult it was to delegate administration in Exchange 5.5. Almost all permissions had to be delegated at the site level due to the permissions inheritance model. The ability to create OUs is a big improvement.

✦ **Application of group policies:** Group policies are used for applying a variety of configurations and settings to users and computers. This is done in a “blanket” kind of way, making it very powerful and easy to manage. It is applied across the board to the object(s), and all retain the particular policy unless overwritten. Policies can be set at the domain, site, and OU level.

OUs can be nested into layers if necessary. This is effective if you want to apply multiple layers of group policies or administration to your environment. Again, we recommend that you keep things as simple as possible. If group policies are set at multiple OU levels, parent OU policies are inherited by child OUs. While this is the default, it can be disabled.

So far we have discussed various ways that OUs are used in Windows 2000. You might be asking yourself how OUs are used in the Exchange environment. Since OUs logically group directory objects, they can be used for administering Exchange.

✦ **OUs can be used for building address lists.** If you wish to create an Exchange address list based on OU membership, it is available as one of the criteria for address list.

✦ **OUs can be used to apply recipient policies.** If you want to apply different Exchange recipient policies for different OUs, you can use the OU as a filter for applying the policy.

✦ **OUs can be used to affect application settings.** If you want to create consistent application settings for your Outlook 2000 users, group policies can be set at the OU level. You can even advertise or publish an application like Outlook 2000 to an OU using group policy.
You should be aware that group policies are not applied for Windows for Workgroups, Windows 3.x, Windows 9x, and Windows NT workstation machines. In order to process group policies, the client operating system must be Active Directory aware.

**Windows 2000 server roles**

The next section will cover some of the different server roles that are important in the Windows 2000 enterprise—including those that are unique to the domain and server roles that span the enterprise.

One of the new Windows 2000 acronyms you should familiarize yourself with is FSMO. A FSMO is a Flexible, Single Master Operation. There are five FSMO roles in Windows 2000. The Schema Master and Domain Naming Master machines are forest-wide roles, and the RID Pool Master, PDC Emulator, and Infrastructure Master are domain-wide roles.

- **Schema Master (forest-level FSMO):** This is the only domain controller that can perform write operations to the directory schema. Those schema updates are replicated from the Schema Master to all other domain controllers in the forest. This server must be available during the time of the Exchange installation, since the Schema is extended when Exchange is installed.

- **Domain Naming Master (forest-level FSMO):** This is the only domain controller that can create new domains in the forest, remove domains from the forest, and add or remove cross-reference objects to directories outside the forest (trusts).

- **RID Pool Master (per domain FSMO):** This server hands out batches (pools) of valid security identifiers from a master list of IDs. The IDs are required for creating new objects in the directory. If a domain controller’s RID pool is empty, and the RID Master is unavailable, you cannot create new security principal objects on that domain controller. However, you could try another domain controller whose RID pool still contained IDs that had been allocated to it before the RID Master became unavailable.

- **PDC Emulator (per domain FSMO):** This server has many functions. It is the Time synchronization server for the domain. It accepts password changes and provides browsing services for down-level clients. It replicates the SAM database with Windows NT 4.0 domain controllers that have not been upgraded to Windows 2000. Password changes and account lockouts performed by other domain controllers in the domain are immediately sent to the primary domain controller for security purposes.

- **Infrastructure Master (per domain FSMO):** The domain controller holding the infrastructure master role for the domain is responsible for updating cross-domain group-to-user references. For example, if you rename a user object...
and then display the group membership of a group that user is a member of, you will instantly see the user’s new name if the user and group are objects in the same domain. When the user and group are in different domains, there is a time lag between when you rename a user object and when a group containing that user displays the user’s new name. The Infrastructure Master updates these references and replicates them with other domain controllers in its domain. If the Infrastructure Master is unavailable, these updates are delayed.

This FSMO role is fairly important to Exchange, since Exchange 5.5 distribution lists are converted to Universal groups during upgrade to Exchange 2000. Any Universal groups containing users and groups from different domains will require these services.

Another server role in Windows 2000 is the Domain Controller, or DC. Domain controllers in Windows 2000 act differently than they did in Windows NT. In Windows NT, the domain stored a writeable master copy of the SAM directory on the Primary Domain Controller (PDC). Any changes that were made to the directory were written to the instance of the directory stored on the PDC. Periodically, the PDC would update other domain controllers, called Backup Domain Controllers (BDC), with changes in its directory. Unfortunately, having a single writeable copy of the directory created a single point of failure, and also made replication of directory data more difficult to configure.

In Windows 2000, the DC infrastructure has changed to fix these issues. Every domain controller has a writeable copy of the domain directory. This means that all DCs in a domain replicate changes with every other DC in the domain. Once replication is complete, every DC contains a copy of every existing object and attribute in the domain. Replication traffic is uncompressed and occurs every five minutes by default among servers in the same site. Defining sites enables compression and allows the administrator to schedule replication times. This gives the administrator much more control over the replication topology than was available in the Exchange Directory service or Windows NT SAM. Sites will be discussed later in the chapter.

Another improvement in replication architecture is the granularity of replication. When you changed a single attribute of an account in Exchange 5.5 or Windows NT, the entire object with all its attributes replicated to every BDC. In Windows 2000, an object with all of its attributes is replicated to every DC at the time it is created. Subsequently, only the changed attributes are replicated. This saves considerable replication bandwidth on the network. Standard domain controllers do not contain copies of objects in other domains. In order to access resources and objects in other domains in the forest, users will contact a Global Catalog server.

When you change the schema in ADS, the entire directory replicates again everywhere.
Global Catalog servers are simply DCs that have the “Global Catalog” check box enabled. When this check box is enabled, the DC becomes a GC. This means it gets a copy of every object in the forest. Seems like a lot of replication doesn’t it? Actually, the GC only holds a subset of attributes for its directory objects. This helps to minimize the amount of replication that occurs between GCs. Global Catalog servers are very important to clients, especially in the logon process, but also any time they need to find resources outside their domain.

It is a good idea to have a GC in every physical location where there are many Exchange users or where access to a GC is limited by bandwidth. Exchange users cause access to GCs frequently.

Examining the concept of sites

In Windows 2000, Microsoft introduces another version of the site. A site in Windows 2000 is much different than an Exchange 5.5 site, an SMS site, or any other kind of site. In Windows 2000, a site is a collection of well-connected TCP/IP subnets—unlike the Exchange 5.5 site, which was defined by its server membership. In Windows 2000, servers and workstation are not placed into a site during installation. Computers are included in sites when they are placed on the network with an IP address that falls into one of the subnets defined in a site’s properties. There is also a default site that is assigned to any machine that comes onto the network in a subnet that is not defined in any other site. This site is called the “default_first_site” in the AD and should not be removed—even if you assign all of your active subnets to manually created sites.

Windows 2000 sites must be manually created and populated with the proper IP subnets by Administrators. You should place subnets with a robust (LAN or fat WAN), reliable network connection. Sites are created using the Sites and Services MMC snap-in. In the beginning of an AD rollout, all active subnets with machines that require AD services are placed into sites. When new subnets are added to the enterprise, they have to be manually added to existing sites in the AD, or new ones have to be created. This means that the Windows 2000 administration team has to work with the network infrastructure team to create a notification and change control process for adding new subnets to the network.

Since site modification carries forest-wide implications, only an Enterprise Administrator can make changes to the site infrastructure.

Servers in a site replicate AD data with one another according to an automatically generated replication topology. Anyone involved with Exchange 5.5 will remember the Knowledge Consistency Checker (KCC). One of the jobs of the KCC is to generate replication links between servers. As servers join and leave sites, the KCC continually generates replication connections between all servers in the forest. By
default, the first server in a site becomes the replication bridgehead. The replication bridgehead server is responsible for coordinating replication within the site. The KCC generates a replication topology that is bi-directional, and designed so that the maximum number of hops a change has to make is three. Keeping the total number of replication hops to a minimum reduces the time it takes for directory convergence, ensuring that it is “loosely consistent,” as Microsoft puts it. DCs or GCs in the same site replicate traffic using uncompressed RPC, and replication occurs at short intervals, as shown in Figure 6-3.

![Replication within a single site](image)

**Figure 6-3: Replication within a single site**

Replication is different when servers are located in different sites than when servers are located in the same site. When DC or GC servers are placed in different sites, it is assumed that they do not have robust, reliable network connectivity; thus, inter-site replication traffic is compressed (for anything larger than 50k), and replication times can be scheduled to optimize network utilization times. The compression ratio is very good, approximately 10:1.

Sites need to be logically connected by Site Links for replication. Site links can be created manually, or the default site link can be used. Figure 6-4 shows a site link.
You should plan your site links carefully and make sure that they are working as you expect when you have finished configuring them. Refer to the Windows 2000 Server Bible for more information on planning and implementing site links.

The replmon.exe utility allows network administrators to view the replication topology in a graphic form, which can be useful for troubleshooting purposes. Using replmon, the administrator can also force a replication cycle, much like forcing a synchronization using Server Manager in NT 4.

Besides replication, sites affect machines on the network in several ways:

- **Logon, DC & GC**: When Windows 2000 users log on to the network, they need to access a domain controller and global catalog server. The client is referred to servers in their site if they are available. This can dramatically decrease logon times. After logon, clients continue to receive referrals to the local site for different types of requested network services. It is a general recommendation to place at least one GC in every site if possible.

- **DFS**: When a user looks for a DFS root server, the AD returns a server in the local site if one is available.
Administrative groups

In Exchange 2000 as in Windows 2000, you can simplify the management of permissions by creating administrative groups. After you create an administrative group and set its permissions, you can add objects to the group. The objects you add to the group inherit the permissions you have set for the group.

When you install Exchange 2000, the installation process creates by default a single administrative group called the First Administrative Group. This administrative group is hidden from view in the Exchange System Manager. For small to medium companies, a single administrative group is sufficient.

You may want to create additional administrative groups to help you manage permissions. There may be several scenarios, as shown in Figure 6-5, where you may want to manage permissions by using additional administrative groups. For example, in a distributed environment, an organization with several divisions, you may want the Central IT Admin Group to manage the permissions over policies, while each division would manage the permissions over its own servers, routing groups, and folders.
Small to medium organizations will often have one administrative group that manages permissions on policies, servers, routing groups, and folders. For a complete discussion on the design considerations for creating administrative groups, refer to Chapter 23. To learn how to create administrative groups, refer to Chapter 5.

Summary

In this chapter, we discussed many of the important aspects of the Windows 2000 Active Directory as they relate to Exchange. If you already had Exchange 5.5 experience before you started using Windows 2000 and Exchange 2000, you probably know about 60 percent of what you need to know about Active Directory and DNS—it’s the other 40 percent that you need to absorb to be able to build your Exchange infrastructure on top of a Windows 2000 directory design appropriate for Exchange 2000. We introduced you to and clarified the importance of the changes from Exchange 5.5 Directory to Windows 2000 Active Directory Services and Exchange 2000. With this foundation, and the tips and cautions we provided, you have one of the most important core components of an Exchange 2000 messaging environment under your belt.
SMTP Routing and the Message Transfer Agent

Message transport components lie at the heart of any messaging system, which is true of Exchange as well. They pump the messages throughout the enterprise. When they are under stress—or when they fail—the messages stop flowing. Message transport was always one of Exchange’s strong suits. With Exchange 2000, this area has changed quite a bit and all for the positive. As a core component of an enterprise communication system, it received a lot of development attention as customers scaled their systems to levels that called out for faster, more robust, better self-healing message transport. The resultant changes are remarkable and add improvements in areas dear to any Administrator’s heart. In this chapter, we take a quick look at some of the changes and their implications.

Transport System Changes Since Exchange Version 5.5

One of the more noticeable changes is the routing protocol. In previous versions of Exchange, the Message Transport Agent (MTA) was built upon the X.400 protocol and the RPC interprocess communication mechanism. The MTA used those protocols for server-to-server message transport within a 4.x and 5.x Exchange organization and for transport to other X.400-compliant message transport systems. In Exchange 2000, Microsoft changes the routing architecture to take advantage
of the Simple Mail Transport Protocol (SMTP) for native server-to-server communication within the organization. The SMTP that Exchange uses is an enhanced version of the SMTP that is integrated into IIS version 5. So, the role of SMTP has increased from just the IMS in older versions of Exchange to the core transport protocol in Exchange 2000. Remote Procedure calls (RPCs) remain, but their role is diminished.

Another major change in the routing architecture is the way servers are logically grouped. In Exchange 5.5 and earlier, servers are logically grouped by site. Site boundaries in a 5.5 organization are determined by network connectivity or administrative control. Servers that have LAN or robust WAN-speed network connections are generally grouped into a site. Once placed into an Exchange 5.5 site, the servers make direct connections to one another as needed for message transfer and are administered as a logical entity. Permissions, size and aging limits, restrictions, the namespace and other important settings were all bound to the site in earlier versions, often making administration more rigid and difficult than necessary.

In Exchange 2000, servers are logically grouped for message transport into Exchange routing groups. Servers in an Exchange 2000 routing group make immediate and direct connections to one another for message transport, similar to Exchange sites in previous versions. Unlike the Exchange site of previous versions, routing groups are created solely for routing purposes, not necessarily linked with administrative boundaries. Once the organization is switched to Native mode (which happens once you are fully migrated from Exchange5/NT4 Servers environments to Exchange 2000/Windows 2000 Servers), routing groups and administrative groups are independent.

In earlier versions of Exchange, an individual server could try different paths based on a table to deliver a message, but had no way of communicating failing links to other servers. The routing table used for that least-cost routing was called the gateway Address Routing Table (GWART). It did not reflect real-time routing conditions, and servers often had difficulty dealing with communication failures. This meant that a message could get stuck in a ping pong or loop situation in larger environments as MTAs passed the message around trying to find alternate routes for delivery as the routes themselves were dynamically changing. Eventually the message would exceed its hop count and the Administrator would be notified of a possible routing error. These issues were often very difficult to track down and resolve in large, complex environments. To address this issue, Exchange 2000 uses link state information for maintaining its logical routing topology. If a link failure is detected, it is communicated to the other servers, making the overall transport system more intelligent and failure-tolerant.

Although much has changed in the routing architecture, the basic function is the same. Exchange 2000 is designed to transfer information within the organization and to external systems in an efficient, cost effective, and reliable manner “automagically” (“automagical” occurs when you combine automatic with a bit of wizardry). Let’s take a closer look at the various components that make up the transport system.
Protocols and Exchange Communication

As we mentioned previously, Exchange now uses SMTP as its transport in most circumstances. You may be wondering why Microsoft decided to use the SMTP protocol for native transport instead of the tried and true X.400 MTA. Besides the obvious fact that SMTP is the most standard worldwide enterprise protocol, there are several other good reasons for using it.

The enhanced SMTP protocol

The most compelling reason to switch to SMTP is probably to overcome some of the negative characteristics of using X.400 and RPC for communication in real-world environments where bandwidth is a regular challenge. SMTP is more resilient in high-latency environments and in many situations requires less absolute bandwidth than the X.400/RPC combination. This characteristic plays well in multinational environments and where cost of bandwidth is important (translation — everywhere).

The SMTP service that is installed with Internet Information Services is extended when Exchange is installed on a Windows 2000 machine. Listed below are some of the enhancements that are made to the SMTP service.

✦ To accommodate the passing of link state information (explained later in the chapter), SMTP is extended to include special commands. This provides a fast and reliable transport for maintaining link status throughout the organization.

✦ Most SMTP services have a relatively simple mechanism for managing message queues. The Exchange 2000 SMTP transport has an Advanced Queuing Engine (AQE) which controls the movement of a message throughout the Exchange transport system. Delivery Queues are dynamically created and maintained for both DNS domains and available communication links.

✦ One of the significant improvements in Exchange 2000 is the Categorizer. The base Windows 2000 SMTP service has a basic categorizer that is enhanced with the installation of Exchange 2000. The Categorizer is used for accessing the AD to resolve sender and recipient addresses. In addition, it expands group membership for messages sent to AD groups and helps to enforce message restrictions and limits. Custom events (script-driven processes) can be fired before or after message categorization.

✦ To enhance performance and component integration, a special driver is implemented for the SMTP service that allows it to read and write data directly to the Information Store.

Aside from the built-in SMTP services used for native Exchange communication, your organization will probably also use the SMTP connector that replaced the IMC and IMS in Exchange 4 and 5. This connector is used to connect to SMTP servers outside the Exchange organization. Multiple SMTP instances (virtual servers) can be used in Exchange 2000. By defining multiple servers, you can create SMTP servers with different port settings, authentication settings, routing restrictions,
and so on. This gives you a lot of flexibility to create connections customized for restricted access, specific TCP port use, relay capabilities, message size limits, and so on.

For more information on the use of the SMTP and X.400 connectors, consult Chapter 10 on enterprise connectors.

The X.400 protocol

X.400 is an excellent enterprise message routing protocol and served Exchange well through versions 4.x and 5.x. The X.400 specification was used to create the Exchange Message Transfer Agent (MTA), which was the heart of the message transport system in all versions of Exchange prior to Exchange 2000. The X.400 protocol can still be used in Exchange 2000, and in many companies it will remain an integral part of the Exchange transport architecture for some time.

In order for Exchange 2000 to communicate with previous versions of the MTA to transfer messages, the X.400-based MTA service is used. When using the MTA to transfer messages with legacy Exchange systems, an Exchange 2000 server looks and acts just like an Exchange 5.5 system. As long as legacy Exchange systems are in use, X.400 and the MTA will be used for sending messages to and from those systems.

The X.400 connector is still available and can be used in situations like connecting Exchange routing groups and connecting Exchange to other X.400 messaging systems. You can directly upgrade X.400 connectors from Exchange 5.5 to Exchange 2000.

Routing Groups

In Exchange 5.5, message routing topology and administrative boundaries are determined by placing servers into sites. These two aspects of the messaging system are separated in Exchange 2000. As mentioned earlier, servers are grouped into routing groups by network connectivity. At first glance, you may wonder why Windows 2000 sites aren’t used for grouping Exchange servers, since the requirements are similar. Namely, Exchange routing groups and Windows 2000 sites can be defined by relatively high speeds and permanently connected network segments.

Some situations may call for a one-to-one correspondence between routing groups and Windows 2000 sites. However, you should not be forced to keep these two logical entities in a one-to-one relationship. Rather, their topology should be created based on your organization’s requirements, and that will not often be a one-to-one alignment. Your use of routing groups should be based on a consideration of the following factors:

- Servers in a routing group should have reliable, robust, permanent network connectivity.
Servers in the same routing group connect to each other in a full mesh topology to transfer messages. This means that any server in a routing group can connect via SMTP to any other server in the routing group to exchange messages.

Routing groups are used to determine public folder access. A user will first try to connect to a local replica of a public folder. If one is not available, other servers in the routing group are tried. Servers in other routing groups are tried according to the connector costs set by an Exchange Administrator.

Routing group membership can easily be changed, so the routing group topology can fairly quickly be re-engineered.

Creating and administering a routing group

To create a routing group, you will use the Exchange System Manager MMC snap-in. The first task you will be required to do is enable routing groups. By default, a single routing group called First Routing Group is installed in a new Exchange 2000 organization. To enable routing groups, right-click on the Exchange organization object and go to Properties. In the properties of the organization object, check the “Display routing groups” check box. This will expose routing groups to the Exchange System Manager MMC snap-in. The properties are shown in Figure 7-1. After you have OK’d this change, you will see the routing groups container in your console.

Once routing groups are enabled, you can create them as needed. To create a new routing group, simply right-click on the routing groups container and choose New, then routing group. In the General properties screen, enter a display name for the routing group. Go to the Details tab if you want to enter administrative notes about
this routing group. Renaming the routing group is as easy as right-clicking on it and choosing Rename.

Each routing group has its own server membership and associated connectors, as shown in Figure 7-2. You can see that three routing groups have been created, called RGEast, RGWest, and RGSouth. Moving a server from one routing group to another is a drag-and-drop operation. Currently, the server named Lorien is a member of the RGSouth routing group. To move it to the RGEast routing group, you have to expand the RGEast container so you can see the Members container. Next, highlight the Members container of Lorien’s current routing group, RGSouth. In the right-hand panel, Lorien can be seen since it is a member of this routing group. To move Lorien from RGSouth to RGEast, simply drag and drop it from the Members container in RGSouth to the Members container in RGEast.

![Figure 7-2: Viewing the server membership and Connectors container of our routing groups](image)

Once the server is moved to the new routing group, all connectors installed on that server are also moved. You may have to manually refresh your console to reflect the changes. If multiple routing groups exist in the Active Directory during installation of Exchange, you will be prompted to choose one for server membership.

**Linking routing groups**

You might be wondering how messages are passed between routing groups at this point. Most medium and large organizations have multiple routing groups and link them so messages can be passed between the servers in those groups. There are different ways to connect routing groups, but they all use a connector and bridgehead servers. Depending on which connector is used, the connection can be configured in different ways.
The most common connector that can be used to link routing groups is the Routing Group Connector (RGC). The RGC is similar in some ways to the Site connector in previous versions of Exchange. Namely, it is very simple to set up and manage, and both sides of the connector can be set up at the same time. Unlike the Site connector, the RGC is schedulable, and uses SMTP for message transfer.

Although routing groups normally use SMTP, they are transport-independent. When a Site connector is upgraded, it becomes a routing group connector that uses RPC to transport messages.

To create a new routing group connector, you will use the Exchange System Manager. Expand the Routing Group object, and then expand the routing group that you will create a connector for. Right-click on the Connectors container and choose New and Routing Group Connector to create the connector for this routing group. Figure 7-3 shows the creation of the routing group connector.

![Figure 7-3: Creating a routing group connector](image)

The first property page you will see is the General property page. In the General property page, you will configure some of the basic properties of the connection, such as the display name, cost, and routing group name. Fill in the following information:

- **Routing Group Name:** A display name to reference the object. Use up to 64 characters, upper- and lowercase letters and numbers.

- **Connects this routing group with:** Is used for identifying the remote routing group to connect to this connector. You can only choose one remote routing group per connector, but you can create as many routing group connectors as you like.

- **Any local/These servers can send mail over this connection:** Specify the servers in your local routing group that can use the connector. You can configure any server in the routing group to use the connection. Alternatively, you can specify a single server if you wish to focus messaging activity to a single point. For redundancy, add a second server.
✦ **Cost:** A value that you assign to the connector. Least cost routes are used first in Exchange 2000. More on costs is available in the link state section later in the chapter.

✦ **Do not allow public folder referrals:** Restricts users from using this connection to access public folders in the remote routing group.

These options are shown in Figure 7-4.

![Figure 7-4: General property page of the routing group connector](image)

Once you’ve entered the General properties of the connection, go to the Remote Bridgehead property page. Here, you will enter the server(s) in the other routing group that can be used for message transfer from this connection. You can specify a single messaging bridgehead server for a more focused connection, or multiple bridgeheads. If you choose a single remote bridgehead, all messages transferred to the remote site will pass through that server and will then be passed on to the recipient’s home server. As you can see, every server in the local and remote routing groups can be included or excluded from using this connection, very similar to the Site connector in previous versions of the product. This gives you precise control over your routing topology when you use this type of connector.

After defining all of the local and remote servers that can use the connection, you’ll want to set up a messaging transfer schedule. Go to the Details tab and configure options for connection times and exception message scheduling. You can configure
the connector to act like a site connector if you choose the “Always run” option. In Figure 7-5, you can see how we configured ours. We have chosen a custom schedule for regular messages, which is configured for more frequent transfers during the weekdays. We also chose to send messages over 2MB every 2 hours.

![Figure 7-5: Setting a message transfer schedule for the routing group connector](image)

The last thing to do is define the overall message limit, and other message restrictions, by configuring options in the Content Restrictions property page. These options are self-explanatory, as shown in Figure 7-6.

Once you have finished configuring all the settings on the connector and clicked OK, you will be asked if you want to automatically create the other side of the connector. Since a routing group connector is unidirectional, two are required for bidirectional communication. If you choose yes, a connector is created in the remote routing group, which is essentially a mirror to the one in the local group. This is very similar to the way the “other side” of the site connector is created in Exchange 5.5. If you want, you can choose no and manually create the connector in the remote routing group at a later time.
If you prefer, X.400 or SMTP connectors can also be used for connecting routing groups — although we recommend routing group connectors in most cases. Since X.400 and SMTP connectors are designed to connect to non-Exchange systems, you have to configure them to identify the other side of the connection as an Exchange system. Identifying the other side as Exchange helps maximize the performance and utility of the connector by allowing it to treat the connection as an internal connection. In earlier versions of Exchange, you would use the “Connected Sites” property page to define adjacent Exchange sites that were serviced by a non-Exchange connector (X.400/IMS). In Exchange 2000, this functionality is accessed in the Connected Routing Groups property page, as shown in Figure 7-7.

When you enter a routing group name in this tab, you are essentially identifying the system that you are connecting to with this connector as an Exchange 2000 routing group in the same organization.

In most situations, it will make sense to use the routing group connector due to the increased functionality and performance and flexibility of that connector. However, organizations using X.400 connectors to connect their sites can upgrade them and continue to use them until they can deploy and test routing group connectors.
Routing group topologies

Routing groups can be connected in various topologies and will depend heavily on how network connections are implemented. There are two basic topologies you can modify to fit the needs of your organization. One is a hub and spoke (or star) topology, and the other is a mesh topology.

The hub and spoke topology is focused on a centralized location, where remote sites connect. Many networks are designed this way, and in these cases, usually the hub and spoke works well. The overall amount of routing group connectors is minimal, since you really only need a single routing group connector for each remote location, making the creation and maintenance of the connectors easier. This topology can be easily modified into a regional hub topology as well, in larger and more complex environments. In a regional hub topology, you simply create additional hub locations and connect them. Figure 7-8 shows a regional hub configuration.

The two major reasons not to use hub and spoke are possible routing inefficiency and lack of resiliency. In the hub and spoke topology, spoke sites have to go through at least one additional hop (the hub site), to transfer messages to other spoke sites, as opposed to connecting directly to one another. Depending on how your underlying network is configured, this may not be the most efficient routing topology. The other problem with the hub and spoke is that it creates a single point of failure at the hub. This issue can usually be dealt with by creating redundancy in the physical network, by using multiple hubs, and by specifying multiple bridgehead servers in your routing group connectors.
A mesh topology can be used in smaller environments with less routing groups. In a mesh topology, all sites connect to one another. While it takes more connectors to create a full mesh, messages traveling from one routing group to any other routing group in the organization only have to go through one connector. Another advantage to this topology is that if a connector goes down, the routing engine can dynamically reroute the message through another site to deliver the message. See Figure 7-9 for an illustration of the mesh routing group topology.

The disadvantage of the mesh topology is that it requires the creation and management of many more connectors, and the number of required connectors gets multiplied every time a new routing group is added. Of course, each of the above topologies can be easily modified to meet your organization’s needs.
One of the areas we have not discussed is the actual method used for determining the path of a message. In general, the Exchange 2000 routing engine attempts to find the “least cost” route, similar to the way it did in previous versions. Administrators assign costs by the “expense” of using a particular network link. The associated expense could mean the dollar cost (or euro, or yen, etc.), or the network bandwidth expense. For example, an Administrator might set the cost of a dial-on-demand ISDN connection higher than a Frame relay connection, due to the higher monetary cost of using ISDN in his location. All the connector costs are stored in the Active Directory and are used by the routing engine to calculate the most efficient route for the message. Sounds fairly straightforward right?

As long as the costs have been assigned properly and all the connectors run flawlessly, life is good. Unfortunately, things do not always run that smoothly. To handle communication outages, Exchange uses link state information. This is a new feature with Exchange 2000 and is the improvement over the Gateway Address Routing Table (GWART) that was used for calculating routing paths in previous versions of the product that we mentioned earlier. Although new to Exchange 2000, the algorithm used for building and maintaining the Link State table has in fact been around for decades and is the basis for the Open Shortest Path First (OSPF) method used
in many routers. The algorithm was developed by Edsger Dijkstra and is well-
documented on several Web sites. Dijkstra’s algorithm solves the problem of finding
the shortest path from a point in a graph (the source) to a destination. It turns
out that one can find the shortest paths from a given source to all points in a graph
at the same time.

As we mentioned earlier in the chapter, enhancements have been made to the
SMTP server in Windows 2000 to support the exchange of Link State Information.
As you might imagine, the exchange of Link State Information depends on the rout-
ing group topology in the organization. Within a routing group, a single server acts
as a Routing Group Master (RGM). The RGM is responsible for maintaining and
communicating an accurate picture of the available communication paths in its
routing group. It is also responsible for updating RGMs in other routing groups. The
link state of an individual communication path is a binary value; either it is up or it
is down.

The RGM periodically exchanges Link State information with other servers in its
routing group using TCP port 691. When a failure is detected by any of the servers
in the group, it is immediately sent to the RGM, which then notifies all the other
servers in the routing group. In the event of a failure, the RGM will immediately
notify RGMs of other routing groups to which it is connected. When Link State
Information is communicated between routing groups, TCP port 25 is used
(the standard SMTP port). If the RGMs in both routing groups are configured as
bridgehead servers in the properties of the messaging connector, then the Link
State Information is passed directly. If not, the RGMs use bridgehead servers to
exchange the information. See Figure 7-10 for an illustration of Link State
Information exchange between servers in a typical organization.

This infrastructure makes for very efficient link state communication. Essentially,
whenever a link is detected, it is very quickly passed to all other servers in a rout-
ing group via the RGM. Simultaneously, the failure is communicated to other routing
groups via the connector bridgehead servers. Link State Information takes priority
over standard user messages, so usually link failures are communicated before
errors in routing path determination occur.

For more information on this topic, please read the white paper on routing on the

Putting It All Together

Now that we’ve examined the major components of the Exchange transport system,
let’s run through some basic routing scenarios so you’ll get a feel for how it all goes
together. The basic routing situations involve routing in a single server, routing
between two servers in the same routing group, and routing between servers in dif-
derent routing groups.
Routing within a server

The first scenario we will examine is when Katrina sends a message to Ed. Both users’ mailboxes are located on the same Exchange server, named Lorellin. Figure 7-11 will help you follow the example.

1. Using Outlook 2000, Katrina submits the message to the Information Store on Lorellin. The message is addressed to Ed.

2. When the Information Store receives the message, the Information Store passes the message to the Categorizer, which applies any restrictions or limits set by the Administrator and queries the AD for the user’s home server.

3. Active Directory indicates that Ed’s mailbox is located on Lorellin and the message is placed by the transport system in the internal queue.

4. The Information Store writes the message into the appropriate database, sending a pointer to the client inbox.
Routing within a routing group

The second scenario involves transport of a message between Katrina and Joshua whose mailboxes are stored on different servers in the same routing group. Katrina’s mailbox is located on Lorellin, and Joshua’s is located on Lorien. Both servers are in the routing group called RGWest, as shown in Figure 7-12.

1. Using Outlook 2000, Katrina submits the message to the Information Store on Lorellin. The message is addressed to Joshua.

2. When the Information Store receives the message, the Information Store passes the message to the Categorizer, where restrictions, limits, and events are run against the message.
3. The message enters the routing system. Here, the routing engine queries the Active Directory and DNS to find the server name and IP address of Joshua’s home server.

4. A queue is dynamically created for the connection and the message is placed there, awaiting transfer to Joshua’s server.

5. The routing engine on Katrina’s server then uses SMTP to connect to Joshua’s server to send the message.

6. When the message arrives at Joshua’s server, it is placed in an NTFS queue, where it is picked up by the queuing engine and routed to the Information Store.

**Figure 7-12:** Message routing between two servers in a routing group
Routing to servers in other routing groups

The third scenario describes the message transport between two servers in different routing groups. This situation gets a little more complicated, since servers in different routing groups do not always connect directly to one another to exchange messages. Instead, they pass the message to bridgehead servers, which pass messages from one routing group to another.

For the example, we’ll be dealing with two routing groups: RGWest and RGEast. In the RGWest routing group, we have two servers named Lorellin and Lorien — the same as in the previous example. Katrina’s mailbox is located on the Lorellin server. In this case, the Lorien server is configured as the only local bridgehead server for the routing group connector to the RGEast routing group.

In RGEast, there are two servers that are named Gorinth and Elora. Our recipient, Eugene, is located on Gorinth. The routing group connector is configured to allow both servers in the routing group to act as bridgeheads. The following process occurs when Katrina sends a message to Eugene:

1. Using Outlook 2000, Katrina submits the message to the Information Store on Lorellin. The message is addressed to Eugene.
2. The Information Store then passes the message to the Categorizer, where restrictions, limits, and events are run against the message.
3. From there, the message enters the routing system. Here, the routing engine queries the Active Directory and finds out that Eugene’s home server, Gorinth, is in another routing group (RGEast).
4. A queue is dynamically created for the connection, and the message is placed there, awaiting transfer to Joshua’s server.
5. The routing engine examines its current Link State Information and determines the best route for the message. In this case, the best route for the message is the routing group connector that connects RGWest (Katrina’s routing group) to RGEast (Eugene’s routing group).
6. Since Lorellin is not configured as a bridgehead server, it uses SMTP to pass the message to the routing group connector local bridgehead, Lorien.
7. When the message arrives at Lorien, the server consults its Link State Information and the connector information in the Active Directory and decides that it can send the message directly to Gorinth, which is configured as a bridgehead server in RGEast.
8. When the message arrives at Gorinth, it is placed in an NTFS queue, where it is picked up by the queuing engine and routed into the Information Store.

See Figure 7-13 for an illustration of this routing process.
In summary, we have covered three of the common routing scenarios in the preceding section, but there are many other variations. You should carefully plan your transport topology to minimize network utilization and message delivery time and should make a continuing effort to manage and optimize it over time. Let’s examine some of the tools you can use for managing your message transport system.

Managing Message Transport

With the addition of Link State Information to the routing architecture, the routing system is more intelligent and efficient than ever. Despite this fact, you should get in the habit of monitoring your transport system to make sure it is functioning as you expect and to be proactive in failure detection.

One of the first things you can do is set up ongoing queue monitoring. Since queues are one of your best indicators of how the messaging system is functioning, you’ll want to set up some performance logging and charting that is focused on the message queues. Figure 7-14 shows us the system monitor, which has been configured to display the queue length for all outbound messages and for the number of messages received per second.
It will take some time and a little patience to get good at using the system monitor, but it is well worth the investment. Here are some tips for using performance logs and alerts:

1. Set up a counter log that captures key statistics such as message queues (MTA Connections object/queue length counter) on key messaging servers such as connector bridgeheads. Set it up to poll every 15 minutes and run the trace for a couple of weeks. This will establish a baseline for how large your queues typically get and will begin to show a traffic pattern in your organization. Comparing the queues on various servers over an extended time period will also give you an idea if messages are bottlenecking.

2. Once you have an idea how things are working, go back and make changes to routing groups, connectors, costs, and so on, to fix any problems that were revealed in the counter logging process. Usually it is best to make changes in a serial format—that is, one after another. If you make too many changes at once, you’ll never be able to tell what fixed or did not fix a problem.

3. Run the counter log again to see if your changes worked. Keep adjusting until you are satisfied that your message routing is working properly.

4. Now you can use the system monitor or alert features of the performance snap-in to notify you of unusual occurrences. Alerts can be set up to monitor message queues and alert you if the queue goes above a certain threshold. Also, the system monitor can be used for a real-time display chart of message queues. Figure 7-15 shows the properties of an alert we set up. If the message queue for SMTP goes over 200, an alert will kick off. We can have the system run a program, do a pop-up alert, log an event, and so on.
Figure 7-15: Setting up an alert for message queues

**Viewing the queues**

Once you have been alerted of a backed-up message queue, you might want to take a look at the messages in it. Viewing the message queues is very easy in Exchange 2000. Simply open up the Exchange System Manager and drill down into the server whose queues you wish to view. Open a protocol container, such as SMTP or X.400, and you will find a queue container for any connections associated with that protocol. Highlight the queue container and you can see statistics, such as the number of messages in the queue and the connection state. Open the individual queues and you can see who is sending the message, its priority, the message subject, and so on. Looking at messages in the queue can give you some clues why the queue has backed up.

**Message tracking**

Another useful tool in Exchange 2000 is the message-tracking center. If you suspect messages are getting stuck or routed improperly, you can turn on message tracking in one or more routing groups and track the routing of the message. To turn on message routing, go to the properties of the server object and mark the check box “Enable Message Tracking.” Message log files are stored for seven days before being automatically removed. You can choose not to remove log files automatically or configure them to be saved for up to 99 days. Log files can get quite large in busy organizations. Make sure you have a plan for removing them so your disks do not get filled up and run out of space.
Summary

This chapter contained a discussion of all the major components of the Exchange 2000 message transport and routing architecture. It covered the major changes from previous versions of Exchange and discussed the implications of those changes. After examining how all the components work together in some typical situations, we looked at some of the tools that can be used to manage and maintain a message transport system.
Data Repositories

One of the main issues facing Administrators of early versions of Exchange was the trade-off associated with the information stores as Exchange servers scaled to accommodate larger numbers of users. Exchange 2000 changes all of that in a dramatic way; it introduces several storage innovations to support multiple databases that can be distributed among several servers. The improvements in storage technology allow access to Exchange information anytime, anywhere, from any device and allow server storage systems to scale yet remain manageable.

There are several major differences between previous versions of Exchange and Exchange 2000. Previous versions of Exchange used the Joint Engine Technology (JET) databases and supported only one private messaging database (Priv.edb) and one public messaging database (Pub.edb) per server. Exchange 2000 introduces an information store based on the Extensible Storage Engine (ESE), which replaces the JET database engine and can support up to 90 separate databases per server. The new Information Store allows you to break up your mailbox and public folder stores into multiple databases and makes it easier to manipulate, back up, and restore the stores.

What does all this mean to you in terms of administration and design? A lot! Here is a brief overview of the advantages of using Exchange 2000:

✦ With the introduction of storage groups in Exchange 2000, the number of simultaneous users on a server can be increased dramatically.
✦ The speed of backup or restore jobs can be increased by combining multiple tape devices with multiple storage groups.
✦ The old adage that Exchange will only store one instance of the message changes.

In this chapter, we will help you understand the data storage options provided to you in Exchange 2000.
Storage Groups and Multiple Databases

Exchange 2000 allows you to create multiple databases on the same server. It also allows you to organize multiple databases into one or numerous storage groups. A storage group is a collection of mailbox stores and public folder stores that share a set of transaction log files. Exchange 2000 can host up to 15 storage groups on a single server. Each storage group can have up to six databases, but one is reserved for the system. The other five are available for mailboxes or public folders. That means a total of 75 databases per information store and per server. In Exchange 2000, all the databases run under a single process: STORE.EXE. Each storage group is an instance of the Extensible Storage Engine (ESE) and consists of up to six databases and the transaction logs for the databases in the group.

Creating storage groups

During the installation of Exchange 2000, a default storage group — First Storage Group — is created. You can find the storage groups under the server in the Exchange Management console, as shown in Figure 8-1.

![Figure 8-1: Storage group in an Exchange server](Image)

You can create additional storage groups by right-clicking on the name of the server and selecting Storage Group under the New menu item. Use the dialog box shown in Figure 8-2 to create a new storage group.
When you create a storage group, you can set two options. The first option allows the system to zero out deleted database pages, which allows the system to clear each 4KB page of data from the drive at deletion. This is a security requirement to avoid another application reading the content of an Exchange information store.

The second option allows the system to use circular logging. Without circular logging, each database transaction in a storage group is written to a log file and then to the databases. When a log file reaches 5MB, it is renamed and a new log file started. Multiple log files are created, and when a database fails, the transactions can be recovered by restoring the data from the log files. Circular logging allows the log file to be reused after its data has been written to the database. This reduces the need for storage, but enhances the risk that you will not be able to recover all the transactions since the last full backup. If you can’t recover transactions, messages can be lost.

Don’t use circular logging in a production environment.

Set circular logging off and ensure that your server does not run out of available hard disk space on the drives that contain your databases and your logs. Also, ensure that you have successful backups to clear the logs. See Chapter 28 for more information on backup and restore.

Creating databases

You can create data store databases in every storage group you create. The databases are either a mailbox store or a public folder store.
Create a public store database by right-clicking on a storage group and selecting Public Store under the New menu item. Use the dialog box shown in Figure 8-3 to create a new public store.

![Figure 8-3: Creating a new public store](image)

Similarly, you can create a mailbox store database by right-clicking on a storage group and selecting Mailbox Store under the New menu item. Use the dialog box shown in Figure 8-4 to create a new mailbox store.

![Figure 8-4: Creating a new mailbox store](image)
Building storage groups and database

You should ask yourself how many storage groups you need to create. Having multiple storage groups allows you to distribute your information store across multiple databases located in different storage groups. There are two advantages to be gained from a wider distribution of the information store. First, it affords you the flexibility to handle more concurrent users. The likelihood that all the users would be accessing the same database at the same time is very low. This means you could distribute the folders to different databases in different storage groups on different physical drive spindles in such a way as to distribute the load. Second, by distributing the information store into different databases in different storage groups, you can minimize the likelihood of a catastrophic failure. If a particular database or a particular storage group fails, all the other databases and storage groups will continue to function.

The downside of having multiple storage groups and databases is the additional administration burden for maintenance and backup. Instead of backing up just two files, you need to back up each individual database. Further, Exchange servers were always noted for their “single instance storage.” This means that when a server received a message for a group of users listed on the to: and the cc: lines who had accounts on its message store it only stored one copy of the message for everyone to use. When people in the group deleted the message, Exchange would not delete it, but kept track of who in the group had asked for it to be deleted. When the last person in the group asked for the deletion, Exchange would finally actually delete the only instance on that server. This made for efficient storage and was far superior to having an instance stored for each individual on the to:, cc:, or bcc: list. Now, Exchange stores a single instance for each storage group that has any of the individuals on the to:, cc:, or bcc: list. So if you use multiple storage groups, you will have multiple copies of the message if the addressees reside on different storage groups.

Thinking about instance storage when deciding who should be in what storage group? Should you put all the VPs in one group? If you do, messages to all of them will have only one instance. However, if that storage group fails, all the VPs will be mad at you at the same time.

Now, what rules should you use to distribute the information store among different storage groups and the database in them? If your organization is running an Exchange service for several sub-organizations, such as an Internet Service Provider (ISP) or a multi-division corporation, then each sub-organization would have its own storage location and data retention policy. Within the sub-organization, you should distribute the information store between different databases to keep each sub-organization data completely separate from the other sub-organizations. It also makes it possible to back up and restore the different databases independently. By distributing data in different databases within a sub-organization, you enhance the performance of Exchange for that sub-organization.
Database content

In earlier versions of Exchange, all messages were converted into Exchange’s internal format—the Message Database Encapsulated Format (MDBEF). In Exchange 2000, information is stored in its native format and not immediately converted. Therefore, it is important to know what applications users are probably going to use so that you can minimize downstream conversion of data type, since conversion takes time and creates overhead.

Some organizations will take advantage of the cache in Exchange and use it to build their Web servers. Ensure that the users creating the WWW are not adding content from a MAPI client or Exchange will store the content with a MAPI wrapper, and that will slow down access from an HTML client with no corresponding advantage.

Exchange 2000 also supports storing streaming media files. When you create a new Exchange database, two files are created: one (.edb) that supports rich-text (as in earlier versions of Exchange) and the other (.stm) that supports streaming Internet content. Figure 8-5 shows the properties of a mailbox store with the two files.

![Figure 8-5: A mailbox store with .edb and .stm files](image)
Streaming Internet content is stored in a Multipurpose Internet Mail Extensions (MIME) format directly into an .stm file. Consequently, clients using standard Internet protocols such as HTTP, Simple Mail Transfer Protocol (SMTP), Internet Message Access Protocol version 4 (IMAP4), and Post Office Protocol (POP) can store audio, video, voice, or other multimedia formats as streams of MIME data without conversion. Whenever a MAPI client attempts to read a message in an .stm file, the file is converted (using document property promotion) and sent to the client.

When Exchange receives an Internet mail message, it stores the body of the message in an .stm file, and the header information (From, To, CC, Time Sent) is converted into Rich Text Format (RTF) and stored in an .edb file.

Since the mailbox store consists of both an .edb file and an .stm file, they should be moved together whenever you want to move a mailbox store.

Replication

You can configure a public folder to have replicas on multiple servers. Replicas are useful for distributing the user load on servers, distributing public folders geographically, bringing the data closer to the users, overcoming the effect of slow speed WAN links, and backing up public folder data. All replicas of a public folder are equal; changes can be made on any instance. In other words, there is no master replica.

Exchange 2000 supports the ability to host multiple public folder trees. Each public folder tree corresponds to a Top Level Hierarchy (TLH). With this extension, it is now possible to refine replication to a single tree within a single public folder per server.

When public folders are replicated, the public folder hierarchy is replicated to every server, but the contents are replicated only to servers on which an administrator has set up replicas.

After you create a store on an alternate server, you need to identify the folders to replicate in that store. Although the storage location on the alternate server is associated with a specific folder hierarchy, the folders in the store are not replicated to the alternate server by default. You must access the folder and specify that it should be replicated to the alternate server.

You can use the Exchange System Manager to add a replica by right-clicking on the folder you want to replicate and selecting Properties. In the Properties dialog box, select the Replication tab, as shown in Figure 8-6.
To create a replica, click on the Add button and then select a public store on an alternate server. When you set up replication, you can set up the schedule for replication by clicking on the Customize button. The replication Schedule dialog box will then appear, as shown in Figure 8-7.

Once replication begins, you can verify replication status, as shown in Figure 8-8, by clicking on the Details button.
When you no longer want a public folder replica, you can delete it from its database.

As mentioned earlier, replicas enhance accessibility and bring the public folder data closer to the user. When a public folder has more than one replica, a user making a connection to it can be connected to the closest replica that exists, and that replica will be the same as all others except for any latency in the system. When it comes to user access, a user who wishes to connect to a public folder will first get to the replica on their server. If that replica doesn’t exist, or the replica on their own server is down, they will then be directed to another replica (determined by the routing group connectors).

When a user creates a public folder, its location in the hierarchy is replicated to every server. (This could take a while in a large organization.) If the new public folder is a TLH, the contents of the new folder are on the user’s public folder server. If the new public folder is not a TLH, the contents are located on every server on which the TLH contents reside.

**Installable file system**

Exchange 2000 introduces the Installable File System (IFS), which can be thought of as a repository of data that can be shared by many applications. Exchange 2000 creates a logical drive M: that can be shared as a network drive and that contains all the resources that users want to share. Drive M: contains one folder that corresponds to the domain name (e.g., SG01.TECNOWIZ.COM) and consists of three additional folders, \PUBLIC FOLDERS, \TEST TREE, and \MBX for mailboxes, as shown in Figure 8-9.
While the public folders can be both listed and viewed, the mailboxes are individually invisible.

**Clustering**

Clustering enhances the availability of an Exchange 2000 system. To run clustering, you must have either Windows 2000 Advanced Server or Data Center version installed on the machines to be clustered. Windows 2000 Advanced Server allows you to define a set of servers as a cluster. Exchange 2000 services can run simultaneously on all the servers in a cluster. If one of the servers fails, another server can take on the failed server’s responsibilities.

A cluster consists of two to four nodes that form a clump. The servers in a clump are connected to each other and to a shared storage device. To a client, the clump appears as a virtual server with a unique name and a unique IP address.

Storage groups can be assigned to a cluster as resource groups. These storage groups reside on the shared storage medium. This configuration maintains the high availability of Exchange services in an organization.

Now that we have seen how the information store is organized, we can turn our attention to the way data stored in an information store is made accessible from any place at any time using the Web Store.
Web Store

With the continued evolvement of a mobile workforce, having access to information at any time and at any place should be a required maxim for doing business. We envision the day when people will use their wireless telephones as their own personal information store.

The Web Store embodies the integration of Internet technologies into Exchange 2000. Changes to the information store structure allow the Web Store to access information that can include e-mail, documents, URLs, voice-mail, streaming media, business applications, workflow applications, and calendaring from anywhere at any time. The Web Store utilizes the information store service to store and retrieve rich text formats and multimedia content. This allows you to take advantage of the cache in Exchange.

The Web Store is designed to provide a single repository for knowledge items that can be indexed and searched.

Indexing and searching

Web Store indexing facilitates searches by key message fields such as recipient, by document properties such as author, and by document content that includes searching through attached text.

Exchange 2000 creates indexes on searchable text on all the items that are stored in an .edb file. It also uses an Extensible Search Engine (ESE) index for all the properties of an .stm file. Whenever a client requests a search, Exchange 2000 combines the information from both indexes to present a result.

HTTP, WebDAV, HTML, and XML

As you will learn in more detail in Chapter 24, the Web Store supports a number of Internet protocols that facilitate access to items within an information store. The technologies that are supported include HTTP, WebDAV, HTML, and XML.

- The Hypertext Transfer Protocol (HTTP) is a standard for requesting and returning information on the Internet. Through its support for HTTP, the Web Store can access information store components as URL resources over the Internet.
- Web Distributed Authoring and Versioning (WebDAV) is an extension of HTTP version 1.1 that allows file copy and move operations over the Internet. By supporting WebDAV, the Web Store allows Exchange clients to manipulate files across the Internet.
The Hypertext Markup Language (HTML) is a standard for delivering content across the Internet. By supporting HTML, the Web Store facilitates the transfer of the content of Exchange items across the Internet.

The Extensible Markup Language (XML) is considered by many a standard for allowing self-describing data. With its accompanying XML Style Sheets (XSL) standard, XML content can be transmitted and uniformly presented to a client across the Internet.

Web Store administration

Administering a messaging environment by taking full advantage of the Web Store provides three primary benefits: lowered cost of administration, enhanced scalability and reliability, and increased security.

Lowered cost of administration

The Web Store is a single virtual location for storing information. With the Web Store, it is possible to administer all information from one place. For many, it will lower operating costs because management, backup, and restore of databases used by many applications will now be centralized in the Web Store. The Web Store can also simplify the training needs of an organization by limiting the number of information resources it needs to learn about.

Enhanced scalability and reliability

The Web Store takes advantage of the scalability and reliability built into the information store to provide the many benefits mentioned earlier in this chapter. Your messaging environment will benefit if you take advantage of the features. For example, distributing content among databases and across physical spindles enhances scalability and availability, but the trade-off is you end up storing more copies of messages sent to groups with members in different storage groups.

However you implement your storage, the fact that all transactions are written to a write-ahead transaction log means that transactions cannot get lost—as long as the transaction log is available to recover the data after a system failure. Logging is a key ingredient in making the Web Store reliable. Write-ahead logging means that Exchange 2000 doesn’t write changes directly to a database, but rather to a log file. This file is subsequently processed and the changes are made to the database. The result is a system that processes changes quickly and reliably.

If you combine logging with two other options for enhancing reliability mentioned earlier—replication and clustering—you have a very scalable and reliable framework for storing your information.
Increased security

The Web Store integrates with Windows 2000 security. The Web Store allows control over permissions at the item and field level. Permissions are shared between the Web Store and Windows 2000 file system.

In a nutshell, the Web Store allows a knowledge-worker to access information anywhere at any time using a multitude of applications including browsers, e-mail clients (Outlook), and Office 2000.

Summary

This chapter provided the information necessary to make intelligent decisions regarding where and how to store your Exchange information so that its availability and accessibility is enhanced. The chapter covered ways you can utilize storage groups to store Exchange information in multiple databases. You learned about reliability and scalability. You also saw how Exchange information can consist of common text, rich text formats, and newer streaming media formats. Finally, we discussed ways to store information using WebDAV so that the information is easily accessed across the Internet or intranet for users with Web browsers.
System Attendant, Monitoring, and Message Tracking

The System Attendant (SA) is a core Exchange 2000 service. When it is not running, your Exchange messaging server is crippled. SA runs as a background service performing duties such as server and link monitoring, maintaining message delivery link state tables, and feeding information to other monitoring systems (like the Event Viewer). Because many of the System Attendant’s functions are performed in the background, they are not normally manipulated directly by the Administrator. Other activities of the SA may be somewhat controlled by Administrators through property pages. While all activities of the System Attendant are important, Administrators tend to focus more on those they can interface with directly, such as the configuration of Server Monitors and Link Monitors and the maintenance of message tracking logs. For that reason, we focus this chapter on the SA roles you can administrate directly.

Monitoring

The System Attendant can be configured to monitor local or remote Exchange servers. It does this by monitoring specified services installed on an Exchange server monitored by a Server Monitor. What users care about is the business functionality that Exchange delivers. It doesn’t do an Administrator a great
deal of good to know that the Exchange box is running if the services that deliver User mail are not functional. Administrators need to monitor services in order to know that a server is healthy from a business perspective. This is the functionality that Server Monitors provide.

Server Monitors work very differently than in previous versions of Exchange. One is created automatically (though notifications are not) and continues running even when the Exchange System Manager is shut down. Others are created and configured manually.

All Exchange servers capable of permanent RPC connectivity can be monitored from a single administrative console. This capability leverages the centralized administrative model built into Microsoft Exchange. Some of the more common services to monitor would be Internet Mail Connector, Exchange Message Transfer agent, and Exchange Information Store.

Creating a New Monitor

To view existing Server Monitors, you open the Exchange System Manager and navigate to the Tools/Monitoring and Status/Status object. There you will see all the existing Monitors for the Exchange server that you are connected to. If you wish to connect to a different server, right-click on Status and select the “connect to” option; this brings you to the screen, shown in Figure 9-1, where you will specify which server you want to connect to. In our screen shot there was only one server to select. By default a monitor is automatically created for every new server.

![Figure 9-1: Choosing a server](image)

To create a new monitor, connect to the server that you wish the new monitor to run on. Right-click on the server in the Status object and select Properties. This brings up the Monitoring panel you see in Figure 9-2. Even if no new monitors have
been created, at least one will appear: the default Exchange Server Services monitor, which is configured to change its status state to Critical if one of the server services it is configured to monitor stops. By default, those services are

- Microsoft Exchange Information Store
- Microsoft Exchange MTA Stacks
- Microsoft Exchange Routing Engine
- Microsoft Exchange System Attendant
- Simple Mail Transport Protocol (SMTP)
- World Wide Web Publishing

You can add other Exchange services to the list by clicking on the Add button in the lower left of the default Microsoft Exchange Services screen and choosing which service you want to add. Similarly, you can add services to new monitors you create. You can alter state changes in the default or new monitors as well. For example, to change to Warning from Critical use the drop-down box in the lower right after you select the resource.

It’s good practice not to add services to monitor to the Service Group in the default monitor, but rather to create a new group and add them there. This allows more flexibility in monitoring, as you can then have one group activating a Warning and another a Critical state.

Click on Cancel to return to the Monitoring page.

There is an optional check box in the lower left of the Monitoring property page, which you can use to stop all monitoring without deleting the monitors. Check this box when your server is overburdened to remove overhead. The System manager page will now show the server in maintenance mode.

Clicking on the Add button in the lower left-hand corner brings up a dialog box, as shown in Figure 9-2, with a list of resources:

- Available virtual memory
- CPU utilization
- Free disk space
- SMTP queue growth
- Windows 2000 service
- X.400 queue growth
Select “Available virtual memory” by highlighting it and clicking OK. Now you are asked to set the period of time that the virtual memory must fall below a threshold to change state. In other words, how long is it acceptable for the server to have less than your minimum threshold before taking an action? You set this in minutes. It is not unreasonable for you to select a value high enough so that you are not bothered by alerts coming from printer spooling (though it’s preferable to avoid having printers on your Exchange server in moderately busy or higher environments).

Also, choose whether the thresholds are warnings or critical state or both and then set the percentages for them. Click OK when you are done.

The default percentage is 5 percent for Warning and 5 percent for Critical, but you cannot leave them both at 5 percent. The Warning state must be larger than the Critical state. The options increment in units of 5.

To revisit this screen, double-click on the monitor. Once it is running, this screen will also show you the current available virtual memory.

You can create a CPU Utilization monitor the same way. The screen, shown in Figure 9-3, looks almost exactly like the one you just saw in the Virtual Memory monitor.
Chapter 9  ♦ System Attendant, Monitoring, and Message Tracking

Figure 9-3: Creating a CPU Utilization monitor

The CPU Utilization and Virtual Memory monitors are very important to use in order to access the ability of your hardware to keep up with the load as configured. Use these regularly as patterns of usage change and as you alter your hardware. For example, you just added 500 users who all use inbox rules and other server services. You see CPU utilization remaining high over a long period. It may be time to add another processor, increase the speed of the current processor, move this server to a faster box, or off load server roles that require CPU bandwidth to other servers.

You create a Free Disk Space monitor by hitting the Add button and selecting Free Disk Space. Specify the drive to be monitored. Set the minimum drive space thresholds in megabytes for the Warning state and/or the Critical state. When the monitor is running, this screen will also tell you the amount of disk space available on the selected drive. From the Monitoring screen, the Free Disk Space monitor is called Free Space Threshold.

You will want to create a Free Disk Space monitor for each drive to track the utilization. When a drive fills up, it fragments more. Also, when a drive holding an Information Store database totally fills, it can cause services to stop.

To create an SMTP queue threshold monitor, select it from the list of possible monitors. In the property page that results, specify the number of minutes before a Warning and/or a Critical state is caused. This means that the system has noticed that the queue is growing faster than it is being cleared. Obviously, if this continues indefinitely, you will run out of disk space. Set your monitoring parameters realistically, but do set them to help you preempt such a failure.

You may wish to select Add a Windows 2000 service. Here, you may enter a name for the monitor. In Figure 9-4, one of the monitors we added was a TCP/IP simple service, and we called it TCP/IP. We set it up for a Warning state of 30 minutes and a Critical State of 60 minutes.
Finally, you may create an X.400 queue growth monitor by selecting the X.400 queue growth resource from the Add Resource box. Set thresholds here for Warning and/or Critical states. Set them in minutes. When activated, this page will show the current queue growth in minutes.

If you have added many resources to monitor, your server monitor screen may look something like that shown in Figure 9-4. Note that the icons on the screen tell you something about the status of the monitor. The E drive has a warning designated by the exclamation mark.

![Figure 9-4: Many monitors added](image)

### Creating Notifications

Exchange 2000 has the capability to generate notifications. Open the Exchange System Manager. Go to Tools ➤ Monitoring and Status ➤ Notifications. Right-click on Notifications. Select New. Your choices are E-mail notification or Script notification. Let’s create an e-mail notification.

Specify the monitoring server. This is the server whose SA service is monitoring Critical and Warning states.

It is not always a good idea to have a server monitor itself. If there is a failure, the server may be unable to generate a notification. Better for servers to monitor each other.
Select the monitoring server. If the one you want is not on the list, try entering its name in the box provided. Specify the server and connectors to monitor. Choices include:

- This server
- All servers
- Any server in this routing group
- Any connectors
- Any connectors in this routing group
- Custom list of servers
- Custom list of connectors

Next, specify the state that causes the notification. Choose Critical or Warning using the drop-down box. If the notification is for a connector, it will only trigger when Critical and cause a notification to be generated. Specify the e-mail address that the notification goes to and the cc: list from the Active Directory list.

Choose the e-mail server from the list or enter the Fully Qualified Domain Name (FQDN) of the SMTP server that will send the notification.

The SMTP server you select must allow the Exchange server to send e-mail using anonymous relay, or e-mail notifications will not be delivered.

If you are interested, study the subject and script. If you choose to alter the subject, use Windows Management Instrumentation (WMI) placeholders for customization. Then, hit Apply. Figure 9-5 shows the properties page with entries and selections.

It takes a while to stop and restart an Exchange server. Some time is taken up by database clean-up, services starting, MTA Check, etc. The cumulative time can be significant. Clock one of your large servers to get a feel. Then set Warning and Critical states with that knowledge.

**Who Should Receive the Notifications?**

Choose the notification recipient carefully. If the person is away on vacation, what happens? If the round trip for the e-mail to get to the person from the server is an hour, that is an additional hour before anyone takes action. Does the e-mail go to someone who does not get e-mail on weekends? Should the e-mail go to a distribution list? Do the recipients know how to escalate a problem within your organization? Do you want a more permanent record of the notification?
Figure 9-5: Creating a notification

Script Notifications

What if you want a notification, but don’t want the notification method to be an e-mail? If that is the case, you want to turn to the provision for using script notifications. In a script notification, an executable is run instead of an e-mail being created. That executable could generate a page or ring a bell or almost anything you could imagine and program. An example of how this feature could be used (or possibly abused) is for you to write a program that copies all files older than 180 days to an archive area on another server and then deletes them from the Exchange server under space duress. Then, you set a monitor on disk space. If disk space becomes critical, this program runs and automatically creates space, avoiding a shutdown of the Information Store. Other practical examples might be sending a notification to a pager or cell phone from a non-Exchange server when e-mail is not working (obviously sending it through Exchange would be hard if the Exchange server itself was down).

If you do take advantage of the script notifications, you should thoroughly test your programs. If a critical event takes place and a notification is triggered and your program fails, the result could be that no one would ever know about it. Often a combination of script and e-mail notification is used to help ensure someone will be notified.
Monitoring the State of Links

The System Attendant also operates Link Monitors. Link Monitors measure the condition of the link (physical circuit, network and transport layer devices, and Exchange Connectors) between two points in an Exchange messaging system. With a bit of intellectual dexterity, it can be used to monitor connectivity to a foreign messaging system as well. The dexterity is needed to make the correct inferences from the displayed status. The System Attendant accomplishes this task by sending ping messages, waiting for their response, and providing an available or unavailable status.

While you can readily monitor a link or a server across any connector, be cautious with slow speed connections or heavily taxed lines. The Link and Server monitors create e-mail messages. The messages look like traffic to the connector. They cause the SA at the recipient server to take an action. It is not helpful to have someone set a fast polling interval on a Link or Server Monitor and hammer your server and the link in between servers. You can flush all SA messages from the MTA queue to rectify this problem.

New Link Monitors are created automatically when the connector is created. You will find the monitors in the Monitoring and Status/Status object in the Exchange System Manager. There are no property pages you are allowed to configure for the actual monitor, but you can create a notification to go with your connector link monitor.

You can create a new Link Monitor Notification in much the same way you created a New Server Monitor Notification. Using the Exchange System Manager, navigate to Tools ▶ Monitoring and Status ▶ Notifications and right-click. Select New E-mail notification (or script notification). Look familiar? It should, but instead of choosing This server (which is the default) in the “Servers and connectors to monitor” box, select the Custom list of connectors, as shown in Figure 9-6. You can also see the results of clicking the Customize button and then clicking the Add button. The system will search for connectors and display a list. Highlight the connector you are interested in and click Add. Fill in the remaining blanks, such as who the e-mail will go to. Then click OK, OK, OK to complete adding your Link Monitor Notification.

If you missed filling in any required boxes while trying to build your monitor notification, the system will beep and alert you to the ones that still need to be filled in.

Also note that when assigning a display name, it is always useful to create a name that describes the type of monitor and connection points, such as “Delhi to Ottawa Link Monitor.”
Message Tracking

Message tracking is an incredibly useful tool, but it consumes resources on your server and takes time before it can be used gainfully. Consequently, it is not turned on by default. To turn on message tracking, go to the server general properties page, as shown in Figure 9-7.

Once message tracking is turned on, your SA will begin reading the P1 envelope information on messages and placing it in a log, which can be viewed by starting the Exchange System Manager. Navigate to Tools ➪ Site Replication Services ➪ Message Tracking Center, and highlight it. Right-click and select Track Message to bring up the General tab on the Message Tracking Center properties page, as shown in Figure 9-8. Use the Browse button to select a From and Sent To entry. You may select more than one entry. Select the servers whose logs will be interrogated. Then hit the Find Now button.

If you have not enabled message tracking on the server, there will be no log file and you will receive an error from the system telling you it cannot find the log.

Once you have successfully found messages that meet your search criteria, you can display the basic history of the message by selecting the message and clicking on the Message History dialog box, as shown in Figure 9-9. Note that the message ID number is displayed in the message tracking.
Figure 9-7: Turning on message tracking on your server

Figure 9-8: Using message tracking

Tip
Don’t forget to turn message tracking off when it is not being used. We usually turn it on when we need to troubleshoot. When you do turn it on, turn it on for all the servers that might be involved in the problem to collect more comprehensive information.
Other SA Activities

As we discussed earlier, the System Attendant is active as a background process in areas that an Administrator would not directly interface with. For troubleshooting purposes, it is good to gain a sense of the type of work the SA is doing in the background. Here are a few examples of the types of activities that rely on the SA:

✦ When the SA starts, it scans for the best Active Directory Server ("Best," in this case, means closest, fastest response, etc.). Once it comes to a conclusion, that name is passed to the DSProxy.dll.

✦ The SA is directly involved in maintaining address lists. The process works like this: The SA service makes a call to WLDAP32.dll to update an address list. That call causes contact with the local domain controller, then a search of the Active Directory for the results that satisfy the rule in the build list, and finally creation of a new address list using those results.

✦ The SA helps maintain administratively imposed Mailbox limits. The SA generates the notification to users that they have exceeded their limits.

If you are using a cluster, you need to create an SA resource for that environment. Before you do that you must ensure the disk, network name, and IP Address are already in the resource group. Once they are successfully moved, you can install the SA resource and list both nodes in the cluster as allowed owners.
Summary

The System Attendant is your best friend. Spending some time learning about its hidden roles will pay dividends when your server is acting squirrely. In this chapter, we discuss topics such as how to build and manage monitors, how to create different types of notifications, and the usefulness of message tracking. This chapter, however, is only an introduction; now it is up to you to really develop a relationship with the System Attendant.
Enterprise Connectors

If your Exchange servers need to communicate with each other across Wide Area Network (WAN) links or the Internet, you need enterprise connectors. The enterprise connectors in Exchange 2000 are the Routing Group Connector, SMTP, and X.400. The SMTP and X.400 enterprise connectors connect your Exchange environment to many standards-based, Exchange messaging systems other than your company’s, as well as to other portions of your own Exchange messaging environment. The Routing Group Connector is for Exchange-to-Exchange only.

Specialty connectors for Exchange 5.x—cc:Mail, Notes, and GroupWise—are discussed in Chapter 18.

This chapter covers the SMTP Connector and the X.400 Connector and includes illustrations of all the property pages. Since some property pages are the same for all connectors, the illustrations do not show all connectors on all property pages, but instead focus mostly on meaningful differences. Even if you focus only on a certain connector, you may still be able to find a relevant illustration in some other part of the chapter.

Basic Connector Concepts

Although enterprise connectors vary, they share a few concepts in common: cost, addressing, and bridgehead servers.

Cost

On the property pages for various connectors, you will see the term cost. For example, on the SMTP Address Space property page, the column for cost appears on the right. Each address space entry has a value for cost. The default is 1. But what is cost?
To provide fault tolerance, load balancing, and redundancy, we often create multiple message paths. The system uses the concept of cost to determine which path to use. We use the term cost in the world of Exchange because, although it is not necessarily the actual monetary cost, it often seems to translate to monetary cost. An administrator wants a higher-cost connector used less than a lower-cost connector, usually because the higher-cost connector is slower, resulting in lower performance — and higher monetary cost. For example, one of the slowest connectors is the asynchronous telephone line; for that reason, dialing long-distance is extremely expensive compared to most other types of connector lines.

Cost values range from 1 (lowest) to 100 (highest). They are defined through property pages for certain objects. The Message Transfer Agent (MTA) selects a lower-cost route over a higher-cost route for message transfer.

The issue for the MTA is not the delta between the values set for the costs, but whether one cost is higher.

When assigning cost values, look at the costs of all connectors in your organization in relation to each other. If the path to the end destination requires that the message pass through multiple Exchange connectors during its routing, each pass through a connector is considered a hop. For multiple hop routes, the MTA views the cost of each portion of the path and adds all of their costs together. Then it compares the total cost for this route to the total cost of alternative routes. It selects the lowest-cost route to the final destination.

For more information on costs and link state routing, see Chapter 7.

**Addressing and routing**

Each time you create a connector, you establish one or many logical paths, or Address Spaces, to the foreign site or system. The Address Space helps control the flow of messages across the connector.

Be sure you understand Address Spaces. Creatively and accurately applying knowledge of Address Space to control the movement of messages is one of the most important skills of Exchange architects and administrators.

For each connector, you can configure multiple Address Spaces that identify a certain type or group of messages and their route to the remote site. The Address Space entries are domain names, parts of domain names, or wild cards. After selecting the lowest-cost route, the MTA looks at each message and determines whether it matches the specification in the connector Address Space property page. If so, it sends the message through that connector.

The Address Space is stored in the connector object, but used in Link State Routing Tables. (In Exchange 2000, the Link State Routing algorithm replaces the Gateway Address Routing Table [GWART]).
In earlier versions of Exchange, both a null and an asterisk (*) were allowed as Address Space wild cards. Many people thought they were the same, but they were not. In Exchange 2000, only the asterisk is allowed, to eliminate confusion. For more information about routing and selection, see Chapter 7.

## Bridgehead servers

Bridgehead servers focus traffic from all Exchange servers in a single location (geographical or topological) through a single server. That server communicates across a connector to another bridgehead server, which focuses communication for all the Exchange servers on the other side. Both of them are called a bridgehead server. Bridgehead servers enable you to monitor network traffic closely and to limit the number of servers you have to deal with in troubleshooting. By focusing the traffic, however, you are increasing the load on a particular server, and perhaps on a particular LAN segment or other potential choke point.

## Making connectors work

There is little magic in making connectors work; the trick is to ensure that all the parameters are set properly on both sides of the connector. All too often, one side is set up differently from the other — for example, enabling TLS security on one side with no local certificate, with the result that all mail bounces back to users. There are many other possible out-of-synch configurations, such as incompatible X.400 transport settings.

The way to eliminate such problems is to focus on them, detail the setting you will use on both sides, understand what each does and how the connector works, and test as you iterate. Finally, make sure that what you build is what you want and need. It is not good enough to have mail passing with no security if you need a secure connector.

## Building a Routing Group Connector

In most large Exchange 2000 environments, the Routing Group Connector is the predominant connector where you have 64 Kbps or higher bandwidth available (the Microsoft recommendation), because

- It is easy to build and administrate.
- It uses the SMTP protocol, so it works even when you have slow links.
- It recovers from many interruptions in link availability.
When you upgrade an Exchange 5.5 environment to Exchange 2000, site connectors are converted to Routing Group Connectors. RPC-over-TCP communication is replaced with SMTP communication, which is more tolerant of slow links and brief interruptions.

To see the Routing Groups, open the Exchange System Manager and right-click on the Organization name. On the General Property page, select the box for display routing groups, as shown in Figure 10-1. After enabling the display, close the System Manager and reopen it for the display change to take effect.

![Note]

Routing Group Connectors are unidirectional, but in most cases, you will want messages to pass both ways. To accomplish that, you need to create two connectors, one on each side. You will be asked when you create the connector if you want the connector on the other side to be created. If you choose to have the remote connector created, it will inherit many of the settings from the local connector you are building. In this way it is similar to the Site connector in Exchange 4.x and 5.x.

After choosing to create the remote connector, navigate to Administrative Groups → Routing Groups → [Any routing group] → Connectors. Right-click and select New, and choose Routing Group Connector from the drop-down box. If you do not have a virtual server as a member of the routing group, you will get an error message and have to add one. If you do not get the error message, you will see the General property page, as shown in Figure 10-2.
On the General property page, fill in a name for the connector. (Choose a name that shows what is being connected.) Select the routing group the connector will connect to. Set the cost. Define whether only specific servers or any local server may use the connector.

One of the primary ways to control costs of transmission and bandwidth utilization is by controlling which servers use which connectors. It may be imprudent to have servers in Osaka, Japan using a connector in Manchester, UK.

Determine whether the connector will allow public folder referrals and select the box appropriately. Next, open the Delivery Restrictions property page, where you can specify a list of addresses to accept exclusively or reject exclusively. The list you choose from is the directory list, so you may need to add custom recipients to the directory, in order to make the entries you want in these boxes.

In the Details property page, you can add notes and see the creation date and the last modified date. In the Delivery Options property page, you specify the connection time, choosing from the following options:

- Always run
- Run daily at 11:00 p.m., midnight, 1:00 a.m., or 2:00 a.m.
- Run every two or four hours
- Never run
- Use custom schedule
If you choose to use custom schedule, click the Customize button, fill in the schedule bars, and set either 1-hour or 15-minute increments. You may also choose to use different delivery options for oversize messages. You can set the definition of oversize (the default is 2000K) and specify the time, just as you did for connection time.

On the Content Restrictions property page, select which message priorities (High, Normal, or Low) will be allowed to pass through this connector.

**Tip**

Priorities for message content should be consistent with corporate policy. Educate your users to set priorities on messages responsibly. Doing so helps your messaging organization move high-priority messages faster, while controlling bandwidth costs.

You can specify to allow system messages and non-system messages, or just one or the other. If you choose to disallow system messages, you may inhibit link monitor and server monitor messages, and therefore the utility of the monitor. You can also specify the maximum size of the messages allowed through the connector.

On the Remote Bridgehead property page, as shown in Figure 10-3, you can specify the name of the remote server. You can also override connection credentials for Exchange 5.x on this page, just as you did in the Override property page in Exchange 5.5 connectors.

**Figure 10-3:** Specifying the name of the remote server
Routing Group Connector does not allow encryption on the bridgehead server. X.400 Connectors provide greater security.

Creating an SMTP Connector

To create a new SMTP Connector, open the Exchange System Manager. Choose the connector object, right-click, and select New. The drop-down box will show you the following:

✦ SMTP Connector
✦ TCP X.400 Connector
✦ X.25 TCP Connector
✦ DirSynch Requestor
✦ DirSynch Server

DirSynch Requestor and DirSynch Server are used exclusively when connecting to a Microsoft for PC Mail (MS Mail) legacy system.

For more information on connecting to MS Mail, see Chapter 18.

Select SMTP Connector. The General property page is displayed, as shown in Figure 10-4. Enter a name for the connector.

Choose a name that illustrates what the connector connects. That makes it easier to identify the connector for all subsequent administrative efforts in an environment with many connectors of the same type.

Designate whether you wish to use DNS to route to each Address Space on this connection (this is the default) or whether you wish to forward all mail through this connector to specific smart hosts. If it’s the latter, specify the smart hosts.

Now add local bridgeheads. The list will have at least one entry — the default virtual server. In addition, other virtual servers you have created will appear. Choose the servers that you need.

The last option on the General property page is to prohibit public folder referrals.
Now select the Content Restrictions property page. It is exactly the same as the content restriction page on the Routing Group Connector.

The Delivery Options property page is a bit different. The top, as on other property pages, allows you to specify when messages are sent through this connection and to use a different, specified time for oversize messages. But the bottom allows you to queue mail for remote-triggered delivery. You could use this option for users who connect periodically and download their mail. These users’ mail clients would issue an ATRN or TURN command, and the SMTP service would then send messages to their domain.

On the Advanced property page, as shown in Figure 10-5, you specify that your SMTP service would send a HELO, rather than the EHLO that older clients might require. Specify the outbound security by clicking on the button and selecting anonymous access, basic authentication, integrated Windows authentication, or Transport Layer Security (TLS) encryption (a security protocol that uses smart cards or certificates on the remote computer).

Default is anonymous. This means not only that you have disabled authentication, but that you have no authentication security.
On the Details property page, you can place administrative notes and see the creation date and the date of last modifications. Use this page to communicate your intent with other administrators. If you created this connector to be temporary, say so here. Also say when you expect the connector to be deleted, and who you are.

On the Delivery Restrictions property page, you can specify a list of addresses to accept exclusively or reject exclusively, just as you did in the Routing Group Connector. (This page is basically the same for all connectors.)

On the Address Space property page, shown in Figure 10-6, you will find multiple address space entries. These entries take wild cards, which manage the flow of messages through this connector. You can build address space entries for many different types of addresses. For example, if you wanted to move all the MS Mail through this connector, you could create a wild card entry of the MS Mail type *. The setup in Figure 10-6 moves all SMTP mail this server sees, as well as anything that ends with HQ.llmarin.com, through this connector.
On the Connected Routing Groups property page, place values if you did not place them in the Address Space page. Otherwise, the system will not know what messages to send through this connector. In a small organization with only one routing group and one administrative group, you should probably use the Address Space and leave this blank. In larger organizations, use either or both to control the flow of messages in more sophisticated and intricate ways. If you use the Connected Routing Groups property page, enter the adjacent routing groups and the administrative routing group.

**Building X.400 Connectors**

The X.400 Connector is commonly used to connect Exchange Servers or to connect Exchange to a foreign X.400-based system. It is often used to connect Exchange servers across the Internet with more security than the SMTP Connector (which transmits in plain text). The X.400 Connector uses bridgehead servers to focus messaging traffic.

The X.400 Connector requires a transport protocol. Exchange supports three transport protocols: TCP/IP, TP4, and TP0 (X.25).
Many pieces of literature refer to the X.25 X.400 Connector, but the correct parallel construction and terminology is TCP, TP4, and TP0 (X.25).

**MTA transport protocols**

Before you can configure the X.400 Connector, you must configure an MTA transport protocol for each server that will have an X.400 Connector. The MTA Transport stack configuration is the abstraction layer that ties the network transport to the connector and allows you to configure the network transport for Exchange.

The available network transports are TCP/IP, TP0/(X.25), and Remote Access Server (RAS). (In Exchange 5.5, TP4 was also provided.)

- **TCP/IP:** Used for an Open System Interface (OSI) such as X.400 over the Windows 2000 TCP/IP services.
- **TP0 (X.25):** Requires an Eicon X.25 port adapter for dial-up or dedicated-line connections.
- **RRAS:** Allows use of Windows 2000 Routing and Remote Access Server for dial-up line communication via async modem, ISDN, and so on. RAS is often used with third-party boards to lower overhead on server and increase the number of ports available.

**TCP/IP**

Transport Control Protocol/Internet Protocol (TCP/IP) is standard on many networks; it is the default protocol of Windows 2000/Exchange 2000 and is defined by RFC 791 and RFC 793. You can configure the MTA transport through the Exchange System Manager by following these steps:

1. Go to Servers ➪ (The Server that will hold the X.400 Bridgehead Connector) ➪ protocols ➪ X.400.
2. Right-click.
3. Choose New TCP/IP. The property page appears, as shown in Figure 10-7. The MTA Transport allows you to configure specific OSI information for the Session (S), Transport (T), and Presentation (P) service access point (SAP) addresses. Your entries can be either in text or in hexadecimal.
4. Name the transport stack.
The Connector property page is empty at this point, but later will show any connectors using this stack. The Details property page is the same as on the Routing Group Connector.

**X.25**

The TP0/X.25 protocol provides communication in accordance with the OSI X.25 recommendation. Shut down your server and install the Eicon X.25 card. Install the MTA Transport stack on the same Exchange Server that has the Eicon hardware. Unlike the old MS Mail, do not install the Eicon card on a different machine (producing a separate gateway machine). Select New X.25 X.400 Service Transport Stack and you will see the property page, as shown in Figure 10-8.
Figure 10-8: X.25 X.400 Service Transport Stack general property page

Note that there are additional entries for Call user data, facilities data, and the X.121 address.

**X.400 TCP/IP Connector**

X.400 remains a popular connector, because it is robust and secure. It is common when connecting to other companies and in national or multinational enterprise environments. In order to create an X.400 Exchange 2000 Connector:

1. Open the Exchange System Manager.
2. Navigate to Routing groups ➤ [the routing group you wish the connector to be part of] ➤ connectors.
3. Right-click.
4. Choose New TCP X.400 Connector. That brings up the General property page, as shown in Figure 10-9. Specify the name for the connector. You must make an entry for the remote X.400 server name and password.
5. Select Modify. The remote credentials box appears, as shown in Figure 10-9.
6. Enter the name and password carefully and exactly. All X.400 connections are secure. Any error here will cause the X.400 Connector on the opposite side to reject connection.
If you select Message text word-wrap, indicate in which column you want wrap to occur. Next, select whether remote clients support the Microsoft Mail Messaging Application Programming Interface (MAPI). If they do, they will get rich text e-mail. If they do not, unselect it.

Figure 10-9: X.400 TCP General property page and remote credentials box

Foreign systems may or may not include text wrap; find out what they are looking for and match it. For Exchange 2000 connections to each other, the default is Never.

Specify whether to allow public folder referrals.

Select the Schedule page and specify the schedule for connection. Default is Always. The schedule options are a bit different from the ones for the SMTP Connector:

✦ **Never**: Really means *never*; stops traffic through the connector during maintenance.

✦ **Always**: Enables the server to make connections whenever it can.

✦ **Selected Times**: Follows the schedule you dictate using the graph.

✦ **Remote Initiated**: Prohibits the local server from initiating a connection, but enables the remote server to make a connection; commonly used in the X.400 RAS Connector, or when you care less about immediate transferring local mail than about keeping the connector open for incoming mail.
On the Stack property page, you can specify the remote host name or the IP address, as shown in Figure 10-10. You can also specify the OSI address points if you click on the OSI button.

Select the Override page. As in Exchange 4.x and 5.x, the Override property page allows you to override the local MTA values for a specific server. Generally, the only change on this page is the name and password—unless you are connecting to a foreign system. In that case, find out what entries are needed in the connection retry area and additional values area.

**RTS values**
The MTA uses a sliding window protocol to transfer data. The MTA sends data and then inserts a checkpoint to verify successful transmission. Inserting checkpoints creates some overhead, slightly reducing transmission speed (unless this value is set too high). If an error occurs, transmission resumes from the most recent checkpoint.

If you have an unreliable network, you may want to decrease the checkpoint size to reduce data retransmission before a checkpoint is inserted. A smaller value reduces retransmission from line problems, but creates more checkpoint overhead, resulting in a lower data-to-checkpoint ratio. For the sake of efficiency and cost, getting the right balance of data and checkpoints is important.

**Connection retry values**
Connection retry values control the number of times the MTA should try to establish a connection and try and transfer a message over an open connection.
The 144 value for Max Open Retries represents once every 10 minutes for 24 hours.

**Association parameters**

MTA Association parameters control the open RPC session between MTAs. MTA associations stay open for five minutes by default, so that the MTAs are not constantly establishing connections. Tune this value up or down, depending on message traffic. Two MTAs can have multiple associations open between them.

**Transfer time-outs (sec/K)**

Transfer time-outs configure the number of times the MTA will try to transfer messages. You are allowed to configure different values for Urgent, Normal and Non-urgent messages.

Do not change transfer time-out values randomly. To reset these values, use the reset Default button.

Connected Routing Groups, Address Space, Content Restrictions, Detail, and Delivery Restrictions property pages are the same as you have seen in the SMTP and Routing Group connectors.

Select the Advanced property page, as shown in Figure 10-11. The Advanced property page is technically complex; if you do not understand the items on this page, do not alter them.

![Figure 10-11: X.400 TCP Advanced property page](image-url)
Chapter 10  ♦  Enterprise Connectors

The first check box allows BP-15 in addition to BP-14. Default is selected. The references are to body parts. BP-15 allows attachments to go as file transfer body parts (FTBP). If you uncheck it, they all go as BP-14, simple binary attachments.

Leave the BP-15 and 14 option checked (the default) if you want Exchange to transmit in Microsoft Database Encapsulated Format (MDBEF, the old Exchange format) when possible. If you clear that box, all messages will be converted to standard P2/22 X.400 body parts.

The two-way alternate button is selected by default. This button causes the MTAs to take turns sending. Even if the link was remotely initiated, both sides get to send. In some cases, you do not want one side to interrupt the other at all, because you want all the mail in one direction to pass as quickly as possible. In that case, you can unselect this option and probably set the recipient connector to remote initiate. Together they would speed up the flow in one direction, and prohibit it over these connectors in the other direction. A second messaging path would allow e-mail to flow in the other direction.

The body part box allows you to specify the alphabet and characteristics of the body part. IA5 is the international set. This affects outgoing mail and needs to be set in accordance with the recipient’s capabilities. Most modern English-based systems work with IA5. This setting commonly would change with foreign systems that use different alphabets.

Set the specification this connector should adhere to in the X.400 conformance area. The dates correspond to different versions of the X.400 specification. If you choose 1984, then you must configure the Global Domain Identifier. Default is 1988 normal mode.

The c, a, and p components of the X.400 address specify the remote MTA and are referred to as the GDI. Use the Global Domain Identifier fields to alter the GDI from the default. This information is used to determine whether the message was relayed, rerouted, or a Distribution List (DL) expanded. A loop is detected and the message returned when the message enters the same MTA GDI without a rerouting or a DL expansion. The GDI setting is used to make that identification possible.

To finish, make entries for either the Address Space or the Routing Group property pages. When you click OK, you will receive a warning that you must configure both sides of the connector before messages will pass. If you have not configured the other side, do so now.

**RAS X.400**

RAS X.400 configures similarly to the TCP X.400. Before building your connector, ensure that you have your modems properly installed and working. If you are using multiple modems, you may want to establish a hunt sequence with your local telephony supplier or on your local PBX to allow one phone number to be called and various lines tried until an open line is found.
Use the remote initiate option to control who pays the long-distance charge. Use the schedule page to aggregate messages and pass them through windows in groups only, to minimize line charges.

Other Connectors

There are other important connectors in Exchange 2000. They include Microsoft Mail for PC Networks (MSMail), IBM cc:Mail, and Novell GroupWise.

For more information about these connectors, see Chapter 18.

Summary

Making connectors work at all takes attention to detail, but making them deliver the service levels your organization expects, while controlling costs of transmission and bandwidth, is both an art and a science. This chapter gives you all the fundamentals: basic concepts as well as explanations of specific connectors. Beyond this introduction, experience will give you the insights and wisdom to fine-tune your system.
Administrating Protocols

In Chapter 3, we covered the basics about many of these protocols — what they are and where they came from. In this chapter, we explain how these protocols are administered on the server, covering issues of interest to administrators, such as setting server limits, managing security settings, creating virtual servers, and more.

Using NNTP

Network News Transfer Protocol (NNTP) is becoming more and more popular as it addresses the inefficiencies of its predecessors. It is regularly displacing Internet mail lists (in which a client subscribes and a list server sends a copy of each message to each subscriber) and the Usenet news system (in which messages are posted for subscribers to retrieve all or some), especially in companies committed to Exchange.

The Windows 2000/Exchange 2000 NNTP implementation is streamlined to require fewer server resources and runs as a background task, but is fundamentally based on Usenet RFC 850. It allows hosts to communicate with each other over LANs and over the Internet. A simple overview of how it works for a user connecting to an NNTP server follows:

1. Interested user becomes a subscriber.
2. Subscriber user application issues a Newsgroups command.
3. If there are newsgroups on the server and the NNTP service is able to respond, the server responds with a list of the new newsgroups and allows the subscriber to subscribe to them.
4. The subscriber connects to a newsgroup and uses the Newnews command to find new articles.
5. The server responds, this time with a list of new articles since the last connection from this subscriber.

6. The subscriber requests some or all of the articles.

7. The server feeds the articles to the subscriber.

NNTP is part of Windows 2000 and Internet Information Services 5. When Exchange 2000 is loaded, NNTP is extended. The additional capability to access other news servers for newsfeeds becomes available. This feature means that the Exchange server becomes a conduit for information from other servers transmitting articles to its subscribers originally contained in newsgroups on other servers. Because NNTP is now used for access to newsgroups, it replaces the earlier Exchange 5.5 Internet News Service (INS).

The Internet News Group folder in Exchange 5.5 INS could remain in the local language. In the Exchange 2000 NNTP implementation, it must appear in English. If you are using a language version other than English, you must reset the language on the folder to English.

Exchange 2000 NNTP enables construction of multiple NNTP servers arranged in sophisticated master-slave topologies. Multiple server and master-slave capabilities are important to Administrators who are intent on building systems that are scalable and redundant. Even with more sophisticated designs, users can maintain their views of newsgroup content while connected to multiple servers. To the users, the sophisticated architecture becomes somewhat transparent.

To build an NNTP master-slave configuration, first determine how many servers are needed, where they will be located, and which will be the master.

In your design, try to keep newsgroup server information frequently accessed close to users, and limit calls for it across slower-speed WAN links. Balance this rule against the storage space required for multiple replicas.

Keep in mind that in a master-slave configuration, all users connect to the slave NNTP servers, never the master. To achieve load balancing, DNS helps distribute the user load across all appropriate NNTP slave servers. One potential problem with a master-slave topology is that you can run into problems with latency. The reason this can be a problem is that each posting is made to a slave server, which in turn posts to the master server. The master server then sends the posting to all the slave servers at the same time. This takes time. There are many factors—including network and WAN bandwidth, server horsepower, and other activity in the NNTP queue—that can affect how much time there will be between posting and actual appearance. So if you have slow-speed links or slow hardware, you may find the latency is too high for master-slave configurations without upgrading one or both.
Once you have configured the servers for your master-slave design, create the newsgroups on the master server:

1. Go to the Exchange System Microsoft Management Console (MMC) snap-in and find the server you want to turn into an NNTP server.
2. Under it find the Protocols container.
3. Open the NNTP container.
4. Now right-click on the default virtual server. The General property tab appears, as shown in Figure 11-1.

![Figure 11-1: Default NNTP Virtual Server property tab](image)

The virtual server uses TCP port 119 for communication unless you specify otherwise or choose Secure Sockets Layer (SSL), in which case the default will be port 563. You must assign unique IP and TCP/SSL port combinations for each virtual server. The Advanced button on the General property tab shows the default. The defaults are TCP 119 and SSL 563.

To avoid an addressing conflict, you must assign unique IP and TCP/SSL port combinations.

On the General property tab, configure the number of connections from other NNTP hosts allowed. The default is 5000. If you expect fewer connections, reduce this number to avoid unnecessary use of server resources. Set the Connection timeout (default is 10 minutes). The default for the path header is the Fully Qualified Domain Name (FQDN). You can alter it if you choose, but your entry will be appended to the NNTP path header, which clients use to determine the path of the message through various servers.
The Settings property tab, as shown in Figure 11-2, enables you to control limits on articles posted. You can also specify moderation and set whether you want other news servers to be able to retrieve articles from this server (the default setting for retrieval is to allow). Setting a moderated group server causes all postings to be sent via SMTP to an SMTP address for review before posting.

![Figure 11-2: The Settings property tab](image)

The Control Message check box is checked by default. Control messages are used by NNTP servers to communicate with each other about items such as new newsfeeds. The server receiving a message about a new newsfeed will determine if the newsfeed should be added to a newsgroup. If you have disabled the control messages, your server will not know about the new newsfeed, will not know to cancel messages, and will not know to remove newsgroups.

The Administrator e-mail account text box specifies the e-mail address that will receive Non-Delivery Reports (NDRs) whenever messages are not delivered to the newsgroup moderator. If you want to enable your system to send NDRs, you must make a registry entry.

- Do not make registry entries on your Exchange server unless you are experienced with registry editing. Erroneous entries can cause your server or a service to act erratically or even refuse to start.

You can cause your system to send NDRs by adding a DWRD value to HKEY_LOCAL_MACHINE\SYSTEM\Current\ControlSet\Services\NntpSsvc\ Parameters. Add MailFromHeader with a value of 1.
The last property tab is the Settings property tab. Default is for every check box on this tab to be checked. You specify the following:

✦ Whether to allow client posting, limit the post size (default is 1000K), and limit the connection size (default is 20MB)
✦ Whether to allow feed posting (once again defaults limiting the size to 15000K and the connection to 40MB)
✦ Whether to allow servers to pull news articles from this server
✦ Whether to allow control messages
✦ The name of the server for moderated groups
✦ The default moderator domain
✦ The Administrator e-mail account

The Access property tab has an area to set access control methods. Those methods include authentication. The default is to allow the first three types of authentication, as shown in Figure 11-3.

For more information about security, see Chapter 22.
The Access property tab allows four methods of authentication:

- **Anonymous Access:** No user name or password is required. If you choose anonymous access, take a look at the Anonymous account being used. The default is IUSR_[Yourserver name]. In this case, it was IUSR_LORELLIN. You may also specify the password and choose to enable automatic password synchronization on this tab.

  If you choose anonymous access, it is difficult to protect your system from hackers—or to track them easily if they get in.

- **Basic Authentication:** The password is sent over the Internet in clear text using standard commands. Because it is easy to capture and display, basic authentication is not much more secure than anonymous access. For the SMTP protocol covered later in this chapter you are allowed to select the TLS encryption option to improve the security.

- **Integrated Windows Authentication:** The client and server negotiate using the Windows Security provider Interface. (In earlier versions this was called NT Challenge/response.) Users must have valid Windows 2000 user names and passwords. Users are authenticated and then allowed access within their security context.

  For both basic authentication and integrated Windows authentication, user name and password are required only when anonymous access is disabled. Access is restricted by using the NTFS access control lists. This means that for security reasons, you should not use FAT partitions.

- **Enable SSL Client Authentication:** This method requires a server certificate.

The next step in a master-slave configuration is to create a newsgroup on the slave servers:

1. Go to the Exchange System MMC snap-in and find the server you want to build the newsgroup on.
2. Find the Protocols container.
3. Open the NNTP container.
4. Right-click on the default virtual server (or the virtual server you want to build the newsgroup on).
5. Choose the Newsgroup container, as shown in Figure 11-4. Right-click.
6. Choose New Newsgroup; the Wizard appears. Supply a name.
7. On the next page, supply a description and a “pretty” name.
Next, create *complementary newsfeeds* (newsfeeds from the master to each slave server, and from each slave server to the master). To do this, use the Wizard, as shown in Figure 11-5, or follow these steps:

1. Go to the Exchange System MMC snap-in and find the server you want to build the newsgroup on.
2. Find the Protocols container.
3. Open the NNTP container.
4. Right-click on the default virtual server (or the virtual server you want to build the newsgroup on).
5. Choose the Feeds container. It shows all the feeds set up for this server. To set up a feed, activate the Wizard as you did for the newsgroup.
6. Highlight Feeds, right-click, choose New, and fill in the Remote server name or IP address. If the server name or address cannot be resolved, you are asked if you want to still use this server name. If you say yes, the next screen enables you to specify Peer, Master, or Slave.
7. On the next screen, specify Inbound or Outbound. If Inbound, pull articles from the remote server (default) or accept push feed. If you leave pull, you will be asked on the next screen to specify the date and time after which to pull all articles.
8. On the next screen, specify which newsgroups will be affected by this feed; default is a wildcard (*), meaning all.
In a non-master-slave design, you would configure the newsgroups and news servers similarly. Expiration Policies and Virtual Directories can also be configured in the Virtual Server container. Current Sessions enables you to monitor and manage active users; you can see who is active and force selected users to disconnect by terminating their session.

You can disconnect all users by choosing *terminate all* after selecting any user. Doing so can be important for maintenance, but irritates end-users, so proceed with caution. If you must disconnect all users, give them plenty of advance warning.

The Expiration policies help you manage space on your server; use them to specify when objects are deleted. To evoke the Wizard to accomplish this task, right-click Expiration Policies, point to and highlight New, and choose Expiration Policy. Expiration is set in hours with a maximum of 9999 hours, or about 13.5 months.

Ask the business unit what Expiration period they need. Set expiration at a reasonable number for type of storage, patterns of use, and size of budget. The more you store, the more time it takes to recover from catastrophe, and the more server resources you need.

The Virtual Directories tab allows you to establish a virtual root and map it to a file system (local or remote share) or an Exchange Public Folder. Right-click to start the Wizard.

### Administering SMTP

Simple Mail Transport Protocol (SMTP) is gaining importance in the expanding Internet-connected world. In Exchange 2000, it is considered the core transport service.

### SMTP message format

Exchange 2000 allows administration of SMTP message format from the Exchange System snap-in to the Microsoft Management Console (MMC). Setting changes for the elements of SMTP we are about to discuss are organization-wide settings. Because of that, they are found in the Global Settings container. Open the Internet Message Formats object. On the General property tab, you will see most message types that are commonly used, as shown in Figure 11-6. You have the option of adding more. Entries for text, audio, and video show their type in more specificity and show the associated extension. You can move a particular type up or down depending on use in your organization. The defaults should suffice for most organizations.
The other object in the Global Settings container is the Message delivery object. Use this object to control the size of outgoing and incoming messages and the maximum number of recipients allowed on a server.

Be cautious about setting the maximum size of outgoing and incoming messages too low. Low settings inhibit real work for many of us. For example, it is a constant source of frustration when mail systems reject transmission of large, perfectly legitimate objects that contain numerous embedded objects—such as the chapters in this book—because they tripped over a low threshold for maximum size.

**SMTP virtual servers**

An SMTP basic service is installed automatically as part of Internet Explorer 5. Exchange 2000’s installation extends that installation. The extension includes the following:

- Link State Information command support and Extended SMTP (ESMTP) commands (such as x-link2state and xexch50)
- An advanced queuing engine
- An advanced message categorization agent
- The Installable File System (IFS) store driver

The extended service manages one or many virtual servers. Each server can be started and stopped independently of the others, although there is only one SMTP service. Of course, if you stop the SMTP service itself, you will affect all of the virtual servers that depend on it. While the service is running, it enables the virtual servers to accept connections. When it is stopped, SMTP both ceases accepting new connections and terminates all existing connections.
Your organization may not require virtual servers; the default single-instance server may suffice. Typically, organizations create virtual servers for the following reasons:

- They are hosting multiple domain names.
- They wish to have more than one default domain.
- They wish to have different authentication requirements for different constituencies.

You can design and configure multiple virtual servers, have them host multiple domain names, and set up different authentication on each of them. You can assign IP addresses, ports, and authentication on an individual virtual server basis to meet those design requirements. Don’t worry about port 25: a virtual server will be listening on port 25 for all IP addresses on all Exchange 2000 boxes.

Caution

All virtual servers on any particular physical server must be part of the same routing group.

Setting up virtual servers is easy:

1. Start the Exchange System MMC snap-in.
2. Navigate to the Protocols container of the server you want to create the virtual server on.
3. Select New, then select New SMTP Virtual Server to start the New SMTP Virtual Server Wizard, which prompts you for a name and an IP address, as shown in Figure 11-7. Remember, the address must be different than that of the default virtual server.

Caution

You can select only an IP address that is already bound to this server. Ensure that you have made appropriate MX and A record entries in DNS for the virtual server.
To configure a virtual server, highlight it, right-click, and choose Properties. On the General tab you are allowed to enable logging in one of four formats:

✦ **W3C extended log format:** This selection (the default) affords you an ASCII text file. You can control the size of the file and what is written to it.

✦ **Microsoft IIS Log File Format:** This selection provides a comma-delimited ASCII text file. No change in the log format is allowed. Many transactions take multiple records.

✦ **NCSA Common Log File Format:** This selection creates a National Center for Computing (NCSA) format. Once again, the format is not configurable. Many transactions take multiple records.

✦ **ODBC Logging:** This selection creates an Open Database Connectivity (ODBC) standard database format.

On the General tab, you can enable logging, although doing so consumes resources. You can limit the number of connections and specify the connection timeout (default 10 minutes). You can also alter the IP address and, using the Advanced button, specify ports and enable filtering.

If you enable filtering, you must add the character @ for valid entries—for example, @wizard.net or @carpediem.com.

Two IP and port configuration issues are of note. First, choosing All Unassigned causes the server to reply to all requests not handled by other virtual servers. Second, you may assign multiple IP and port combinations to a single virtual server. The General property tab is shown in Figure 11-8.
The Secure Communication button under the Advanced button allows you to view and set the secure communications used when the server is accessed. Two buttons appear in this area. The first is for the Certificate Wizard. Use this Wizard to create and administer server certificates.

If your server does not have a certificate loaded, you will be notified and asked whether you wish to create a new one with the help of the Wizard, or attach to an existing certificate.

The Access property tab has an area to set access control methods. Those methods include authentication.

For more information about security, see Chapter 22.

The default is to allow all three types of authentication. Note that SSL is not an option as it was with NNTP and are covered in the NNTP section in detail.

The second area is for communication; you can specify that once a key certificate is installed, access to it must take place on a secure channel.

The third area on this tab is for connection control. Here, you grant or deny access to the resource. You can use either IP addresses or domain names. You may choose to allow only the list you specify — or do the reverse, and allow all but the list you specify.

The last area on the Access property tab is for relay restrictions. This area prevents hackers from hijacking your SMTP server to send spam e-mails that look like they came from your server. Again, specify a list using IP addresses or domain names and specify only the list, or all except the list. The button at the bottom of the property tab for relay restrictions provides the option (default is checked) to allow all computers with authenticated certificates to relay regardless of the list.

On the Messages property tab, you can do the following:

- Alter limits for message size (default is not checked, but contains value of 2048K, or 2MB)
- Limit the session size (default is not checked, but contains a value of 10240KB)
- Limit the number of messages per connection (20)
- Limit the number of recipients per message (default is 6400)

You can also specify who will receive a copy of non-delivery reports. You can specify a badmail directory, but you must do it through the metabase directory properties, not just by retyping it in this location. The default is c:\Exchsrvr\Mailroot\vsi#\badmail, where vsi# is the designation for the virtual server (for example, vsi1 is a virtual server).
Do not use drive M for a badmail location; it causes problems with message flow, Exchange administration, and transport services.

The last text box is for Forward all mail with unresolved recipients to hosts. You would choose this option in a mixed Unix environment, placing the name of the Unix mail server in the box to have it resolve mail that your Exchange server cannot resolve, but which is appropriately addressed to your organization. The entry causes Exchange to forward the unresolved e-mail to the Unix host.

The Delivery property tab allows you to specify outbound e-mail retry, expiration, connection values and security. In the outbound area, the default first retry takes place after 10 minutes. The second and third retry intervals are also 10 minutes. The default subsequent retry is 15 minutes. Default delay notification is 12 hours, and the expiration timeout is 2 days.

Make sure that outbound e-mail retry, expiration, connection values, and security are appropriately set for the service levels you commit to.

The Outbound security button brings up a page where you may set Anonymous, Basic, or Integrated Windows authentication. You may also select TLS encryption to increase security. If TLS encryption is selected and the remote server does not support TLS, the name and password are sent in clear text affording no security.

If you are trying to connect two servers, ensure that the outbound security settings match the inbound security settings on both servers. If you use TLS, establish a policy to select the TLS encryption box everywhere in your organization.

The Outbound connections button brings up a page that allows you to limit the connections and the connections-per-domain. By default, both are checked. The default value for the former is 1000 with a 10-minute timeout. The default value for the latter is 100. The default TCP port is 25.

The Delivery property tab also has a button for advanced settings, which brings up the Advanced delivery property page. Here, you specify the maximum hop count for your messages (default is 15). Unless you are familiar with routing or have a particular problem to address, stick with 15. That number allows 15 lines of routing hops in the header — any more, and a Non-Delivery Report (NDR) is sent.

The Advanced delivery property page includes a text box for Masquerade domain. Enter a domain name here if you wish mail to appear as if it came from the domain you specify. That domain name is placed in the Mail from: field (found in the header) and From: field (found in the message body) of all outgoing messages. The From: field reverts to the real entry after the first hop, while the Mail from: field remains the same throughout the routing.
The Advanced delivery property page also includes a text box for Fully Qualified Domain Name, a button to check DNS, and a button for Smart Host. The Smart Host designation is used with the check box to attempt direct delivery before sending to Smart Host. This causes Exchange to send the mail to the destination SMTP host first. If that is unsuccessful, then it goes to Smart Host.

There is also a check box (default is unselected) for performing reverse DNS on incoming messages, to find out if the domain name is valid (from a DNS perspective). Exchange tries to determine if the domain a message is coming from matches that of the domain in the `EHelo / HELO` command. If it does, Exchange makes no change to the Received: header; if it does not match, Exchange will stamp `unverified` in the header.

The last item on the Advanced delivery property page, Configure external DNS, is self-explanatory.

**Other SMTP implementation concerns**

SMTP is used for the addressing in Exchange 2000 instant messaging. To make it easy for users in other domains to guess your users’ instant-messaging addresses, we suggest you subscribe to the convention that is growing in popularity. That convention is to set up DNS so the host name of your routing servers is `im.domain-name.com`, where `im` stands for instant messaging and domain name is your domain. If you do that, then a user outside your domain could send an instant message to Steve Depoian at Ilmarin.com as `SteveDepoian@im.Ilmarin.com`.

Another caveat regarding SMTP concerns deleting a Mailbox store. Before deleting a Mailbox store, clear the SMTP queues of all messages. When you tell the system to delete the store, you are warned if there are residual messages in the outbound SMTP queue. If you continue anyway, you are given the option of identifying a new store inbound queue for the SMTP messages.

Changes made to a virtual server do not take effect immediately, but only after the IIS metabase update service has propagated the changes to the metabase. How quickly that happens in your environment depends on the load on your server. Even on snappy servers, it may take a few minutes.

**Administering IMAP4**

Like NNTP and SMTP, IMAP4 and POP3 appear on the Exchange System snap-in under [server name], Protocols container. By highlighting, right-clicking, and choosing properties, you bring up the HTTP General property tab, as shown in Figure 11-9. Select the IP address from those bound to the server, and use the Advanced button to designate the TCP port and SSL port (the defaults are 143 and 993). You are allowed multiple entries. On the General tab, you may also limit the number of connections, specify the connection timeout in minutes (default is 30) to enable fast
message retrieval (default is unselected), and include all public folders when a folder list is requested (default is include).

The Access property tab includes a button for authentication. Two options are allowed:

✦ Basic Authentication
✦ Integrated Windows Authentication

These two types of authentication have the same definitions and caveats used earlier in this chapter in the SMTP and NNTP sections. The secure communication and Connection areas also have the same options and explanations as in the section on SMTP.

The Message Format property tab offers three options:

✦ Provide message body as plain text
✦ Provide message body as HTML (default)
✦ Provide message body as both plain text and HTML

You can also specify Exchange Rich Text Format.

The plain text and HTML is not selected as a default; if selected, it increases overhead for your system to convert the messages.

As with the other protocols, you can create a new virtual server using the Wizard by highlighting the IMAP4 protocol, right-clicking, and choosing New IMAP4 virtual server.

Figure 11-9: IMAP4 General property tab
Administering POP3

POP3 appears on the Exchange System snap-in under [server name], Protocols container. Highlighting, right-clicking, and choosing properties brings up the POP3 General property tab. It looks similar to the IMAP4 property tab, as shown in Figure 11-9. There you select the IP address from those bound to the server, and use the Advanced button to designate the TCP port and SSL port (the defaults are 110 and 995). You are allowed multiple entries for this virtual server. On the General tab, you may also limit the number of connections and specify the connection time in minutes (default is 10). The Access property tab and its buttons are the same. For more information, see an earlier section in this chapter, Administering SMTP.

The Message Encoding property tab offers three options for MIME:

✦ Provide message body as plain text
✦ Provide message body as HTML (default)
✦ Provide message body as both plain text and HTML

You can also specify the character set and the alternative of UUEncode. If you select UUEncode, you are allowed to further specify that you want BinHex used for Macintosh compatibility.

Use BinHex in environments heavy with Macintosh computers.

As with the other protocols, you can create a new virtual server using the Wizard by highlighting the POP3 protocol, right-clicking, and choosing New POP3 virtual server.

Controlling HTTP Access

HTTP appears on the Exchange System snap-in under [server name], Protocols container. A sample case shows the default virtual server (named LORELLIN) and the three folder objects it created (Exadmin, Exchange, and public), as shown in Figure 11-10. By highlighting, right-clicking, and choosing Properties, you do not bring up the HTTP General property tab, but instead get a warning that you must use IIS to manage the virtual server’s settings.
If you created another virtual server, you can right-click on that server (Moria in the sample case, as shown in Figure 11-10) and bring up a General property tab, as shown in Figure 11-11. (The second virtual server was created with the Wizard: highlight the HTTP protocol, right-click, and choose New HTTP virtual server.)
On the General property tab, select the IP address from those bound to the server and use the Advanced button to designate the TCP port and SSL port (defaults are 80 and 442). You are allowed multiple entries for this virtual server, but they must be unique. You may also limit the number of connections and specify the connection time in minutes (default is 900), and you select mailboxes for the Public Folder. If you select the latter, you may wish to modify the SMTP domain, which in the sample case is Ilmarin.com.

You are instructed to go to the IIS snap-in to configure the other settings. At the snap-in, you can administer other IIS virtual servers. The view from the IIS snap-in is shown in Figure 11-12. As you can see, both the default server and the virtual server we created called Moria are available.

For more information, consult the *Windows 2000 Server Bible*, published by Hungry Minds, Inc.

**Figure 11-12:** Non-default HTTP Server property box

The Access property tab has control buttons (all are selected by default) for the following activities:

- Reading
- Writing
- Script source addressing
- Directory browsing
You can specify which type of permission will execute: None (default), Scripts, or Scripts and Executables.

The Authentication area brings you to the Authentication Methods property tab. Anonymous is allowed, but Basic Authentication is the default. You can alter the default domain name by typing in the text box. The last method allowed is Integrated Windows Authentication, and it is one of the defaults as well.

## Managing RPC

Remote Procedure Calls (RPCs) play a critical role in Exchange 2000, although that role has diminished from its apex in earlier versions of Exchange. Older versions of the Exchange and Outlook clients, and many other MAPI programs, made calls to the directory service using RPC, used RPC for server-to-server site connectors, for client-to-server Outlook and Exchange user applications, and for intra-site directory replication. Today, TCP and other transport mechanisms are more popular than RPC. However, RPC still remains important and requires a basic understanding of what it is and how it works to ensure it does not break on your Exchange system.

RPCs, used in client-server programming, enable certain procedures to run locally and others to run remotely. Even when all code appears to be running locally to the application, some of it may be running remotely. Application libraries are not linked statically. This dynamic is accomplished by using programmatic stubs. The client calls the stub to cause a remote procedure to run at the server instead of locally. The model relies on a process view of the world rather than a transport view; it is easier to program because it relies on the underlying network for transport.

Handshakes are required for every RPC. This requirement has two serious implications:

- It drives up network utilization.
- It does not recover well from interruption.

RPCs require robust, high-speed connectivity and suffer from unreliable or slow links. They do not handle timeouts or disappearing links well. Other transport protocols, like TCP/IP, are built differently in order to address these problems.

In Exchange 2000, RPCs are used for special situations, such as certificate requests (users can also use HTTP), the Site Connector (TCP is used to initiate the connection, but RPCs take over after that), and communication to Exchange 5.5 servers.

For information about testing and tuning RPC connectivity using the RPCPing tool, see Chapter 27.
Summary

In Exchange 2000, Microsoft organized all the important protocols — NNTP, SMTP, IMAP4, POP3, HTTP, and RPC — in one place, made their property pages similar, and added wizards. This chapter explains where and how to administer the protocols section of the Exchange system MMC. You should be able to use the important messaging and collaboration protocols covered in this chapter to deliver real business utility to users on your Exchange messaging organization.
Exchange 2000 was built with client-server architecture. The client-side applications are many and varied. Microsoft built their own clients—Outlook 2000, Outlook Express, and Outlook Web Access—which are covered in detail in this part. Other manufacturers and organizations built clients that support popular protocols, such as POP3 and IMAP4. Those standards are also supported, providing access for many popular clients. In this part of the book, we walk you through installations, differences between the various client applications, configuration issues, and crucial foci, such as security. Because your users will be interacting with Exchange through their client applications, this part of the book covers important topics for any Administrator.
The Exchange 2000 default client is Outlook 2000. Although Outlook can be used as a standalone product, this chapter will treat it as the full-feature client. Exchange works well with a variety of client applications, but Outlook 2000 is probably the most popular choice, due to its tight integration with the Microsoft Office suite, extensive feature set, and the ongoing development of the server-side components. Therefore, you will want to give serious consideration to Outlook 2000.

Introduction to Outlook 2000

Messaging and collaboration applications have matured considerably over the last few years, and Outlook is no exception. Since its introduction in 1997, Outlook has been the flagship Exchange client application. Older generation Microsoft Exchange clients, such as the standard Exchange 4.0 and Exchange 5.0 clients, had a limited feature set and, simply put, not enough flexibility for today’s users or administrators. The look and feel of these applications was unsuitable for manipulating a variety of data types, and collaboration options were minimal.

Through the different releases of Outlook, Microsoft continues to improve the application. In its present form, Outlook is a premier information manager that provides a single point of access to messages, calendar and task data, contacts, custom applications, and more. However, any good Administrator knows that additional business utility and application complexity can result in an increased need for support. Microsoft has tried to minimize that need in Outlook 2000. In fact, one of the design points for Outlook 2000 was increased utility without increased administrative burden.
In our experience, users who can greatly benefit from the utility of many Outlook features tend not to utilize those features. Even though Outlook has been in use for years, it is unexplored territory for many. Great benefits can be achieved when administration and support personnel make a purposeful effort to understand their users’ application and collaboration needs. With that knowledge, they can advise and educate on useful features that end-users are not familiar with, as well as respond effectively when support issues arise.

**Components of Outlook 2000**

Outlook is available as a standalone product or with any version of the Office 2000 suite. The base product has everything you need to connect to Exchange (any version), some legacy mail systems, and a variety of Internet mail services.

Outlook consists of the base application, as well as several other components that interact with the application. Some of these components are described in previous chapters and include the following:

- Message store providers
- Address providers (directory services)
- Personal folders
- Fax, data synchronization, and other services

**Message store providers (POP3/IMAP/Exchange/MAPI)**

Outlook as a messaging client depends on message store providers. Store providers come in many flavors and have vastly different capabilities. In general, a user will want to have different stores for business and personal e-mail. It is important to note that a user is not limited to a single store provider and has the ability to use multiple server-based and local stores. In addition, Outlook has excellent support for customizing how multiple store providers interact with each other so they don’t “step on each other.” Figure 1 shows an implementation of Outlook where the user has connections to multiple store providers.
Figure 12-1: A user connects to Exchange 2000 server for corporate e-mail, a customer’s IMAP4 server for collaboration on a project, and their ISP’s POP3 server for personal e-mail.

Exchange 5.5 Information Store

Outlook (any version) can access the Exchange 5.5 Store. When an Outlook user contacts Exchange 5.5 to send or receive mail, use public folders, or send meeting requests, Outlook is utilizing the store. Also, because it has extensive protocol support, that store can service a variety of other e-mail clients as well, such as Eudora, Netscape Mail, and Outlook Express. Clients can access the store by using MAPI, IMAP4, POP3, SMTP, or HTTP (through Outlook Web Access). The store has two basic functions: as a repository for client mailbox data and as a repository for public folder data. It also functions as a management service for the same data.
Exchange 2000 Information Store
Like the 5.5 Store, the Exchange 2000 Information Store is responsible for data storage and management. While it continues to be a terrific platform for messaging, the Store has taken on a much greater role in Exchange 2000, as it encompasses additional protocol support, native storage for XML (Extensible Markup Language) and MIME (Multipurpose Internet Mail), and new APIs to access data in the Store. Not only does it manifest greater flexibility in the way the databases are physically and logically organized and managed, it contains full-text indexing on common attributes and a more efficient data file system. For well-connected Outlook clients, this is the store provider of choice.

POP3
Many modern messaging stores — including those found in Exchange 5.x and Exchange 2000 — provide POP3 support. POP3 (Post Office Protocol version 3) is a very common messaging protocol on the Internet. Most Internet Service providers offer some level of free POP3 support, which makes it a popular choice for home users. POP3 store capabilities are limited, since the protocol command set is small and the underlying architecture is relatively unsophisticated. When Outlook is configured in “Internet Only” mode, it is assumed that the store provider will support either the POP3 or IMAP protocol.

When you use POP3, all the mail has to be downloaded to a store on your client. A copy can be maintained on the server after the download if your client and the server are configured for that.

IMAP4
The Internet Message Access Protocol version 4, or IMAP4, is similar to POP3 with the exception of certain enhanced capabilities. An IMAP4 store has support for access to multiple mailboxes, searches, read and unread message flags. In addition, there are some advanced features, such as synchronization and manipulation of data on the server. With an IMAP4 client and store, mail can either be downloaded or maintained on the server.

cc: Mail
Outlook provides an information service that can be used to access to a mailbox and shared folders on a legacy cc: Mail post office. In migration situations, this information service can be useful. It allows users that have migrated to Exchange to access legacy e-mail data that was not migrated. Except in migration situations, most designs wean users off their cc: Mail information store and onto Exchange.

MS Mail
The MS Mail service provides access to a mailbox and shared folders on a legacy MS Mail post office. This service is also useful in migration situations. Except in migration situations, most designs wean users off their MS Mail post office information stores and onto Exchange quickly.
Address providers

One of the most important components in Outlook 2000 is the Outlook address provider. Think of address providers as directories, server directory services, personal address books, and personal contact lists. All of them are databases or services that own databases that contain the information. A directory service, like the Windows 2000 Active Directory Service (ADS), integrates the directory database and directory information management into a single entity. All calls from the clients go to the service that controls all access to the database. Like store providers, there are a variety of address providers with various levels of functionality available to Outlook users. Typically, users need multiple address providers in order to access and organize information about the different communities with which they interact.

Exchange 5.5

The Exchange 5.5 address book provider is called the Directory Service (DS). It is a database built on enhanced JET 3.0 technology similar to the 5.5 Information Store. The DS stores and manages Exchange directory objects and their attributes, and it replicates changes to other 5.5 servers. When a client that is online communicates to an Exchange 5.5 server looking for an address in the Global Address Book (GAL) or looks up the properties of a co-worker and finds that they’re in a legacy Exchange 5.5 environment, it is the Exchange Directory Service that provides the information.

MAPI clients released prior to Outlook 2000 search for a Directory Store on all Exchange servers. When these clients connect to an Exchange 2000 server, the Directory Store proxy service relays the RPC requests to the nearest Windows 2000 Global Catalog (GC) server. This happens every time the client connects to an Exchange 2000 server for directory access.

Outlook 2000 can query Exchange 2000 for the name of a GC to use and record its name in the registry for future use. Once Outlook 2000 knows where the GC is, it no longer needs an explicit referral to a GC.

Windows 2000 Active Directory Service

The Windows 2000 Active Directory Service (ADS) is the address provider for Exchange 2000 clients and services. ADS is a quantum leap in sophistication, granularity, flexibility, and industrial strength from Exchange Server 5.5. For example, the information that can be accessed from Outlook is no longer limited to Exchange-related items; anything in the directory that you have permissions to is available. Because the directory schema is extensible, anything your company organizes to populate the directory with and provide you permission to access can be available to you in the directory.

Microsoft Lightweight Directory Access Protocol (LDAP) service

An LDAP service is a directory access protocol service for X.500-compliant directories. In conjunction with a directory store, it can act as an address book and an
authentication mechanism for applications such as Outlook. Many Outlook clients will use the Windows 2000 Active Directory that has native support for LDAP. The Microsoft LDAP service provides an LDAP service that allows connection to additional LDAP providers for directory services. LDAP has gained a tremendous amount of momentum recently and will likely continue to evolve into the staple of access protocols for corporate and global directories.

**Offline Address Book**
Assuming your Outlook client has downloaded a copy, the Offline Address Book (OAB) is available when the client is not on the network or in a location where a server-based directory is not accessible. The OAB allows the user to access directory information that is usually accessed from the server, even though the user cannot access the server at this particular time. The OAB is a text file image of some portion of the Global Address List (GAL)/Windows 2000 Catalog Servers. Administrators can control what portion of the GAL is copied and how often it is updated. When the client has a copy with the right information in it, they are able to prepare and address messages off-line. The messages will be sent when the user is again connected to the Exchange environment. In Exchange 2000, the OAB is based on one or more address books provided by the Windows 2000 Active Directory.

**Personal Address Book**
The Personal Address Book (PAB) is a file with the .pab extension that is usually stored on the local machine in the %windir%\Application Data\Microsoft\Outlook directory. The PAB provides the ability to create, store, and manage addresses that can be used on the local machine. Think of it as a private directory.

The Contacts module in Outlook is supplanting much of the functionality of the Personal Address Book. In fact, many organizations now discourage use of the PAB due to support and maintenance issues. Most Administrators and users have moved (or plan to move) names to their Outlook Contacts folder. Once in the Contacts folder, names are available for many other third-party applications and add-ons to Outlook, such as WindowsCE devices or cell phone downloads.

**Outlook Address Book**
The Outlook Address Book is essentially an Outlook contacts list that is shown in the standard address book format. The Contacts folder supports personal contacts and personal distribution lists. In previous versions of Outlook, distribution lists were very difficult to configure in Contacts folders, which is why some users preferred PAB. Outlook 2000 users should be encouraged to use Contacts as their personal directory instead of the standard Personal Address Book for a number of reasons:
✦ Contacts have much better security, are more easily filtered and searched, and are better integrated with the rest of Outlook and third-party applications.

✦ The object attribute list is more extensive in contacts, and you can add custom attributes, categories, and views.

✦ The Contacts list is much more likely to be included in the systems backup routine. Of course, it is probably the Administrator’s responsibility to ensure that that happens.

**Personal folders**

Personal folders are a common component in many implementations of Outlook. Sometimes referred to as a *PST* or *personal store*, these folders are stored in a PST file (a file with the .pst extension). They provide a way to take information out of a server-based store and have it delivered or saved locally or on a network file server that you are connected to actively. Personal folders support all the default data types (message, calendar, task, contact, journal), and you can access multiple PST’s concurrently. PSTs can be encrypted and password-protected when data protection is required. Rules applied to a PST will only fire when the PST is available.

One feature not commonly known about PSTs is that they store two copies of the message: one in ASCII text and the second in MDBEF (Exchange native) format.

**FAX and other services**

Microsoft provides a FAX service with Outlook that can be used in combination with FAX addresses in Contacts and will utilize local TAPI settings for dialing. Additional services are also written for Outlook by third-party vendors. Popular services include WindowsCE devices, Windows PC, cellular phones, and RIM Blackberry Devices.

Visit [www.slipstick.com](http://www.slipstick.com) for useful information about Outlook, third-party tools, and development.

**Using Outlook or Common Outlook Configurations**

Outlook was built to be a multipurpose collaboration and messaging client application. While its design affords many combinations of information stores, service providers, contact access and protocols, Outlook can be configured in three main ways.
Internet client

The first configuration is as an Internet mail client. In Internet client configuration mode, you can connect to an Exchange server or any other server running appropriate Internet Mail services—such as POP3, IMAP4, and SMTP—as long as you have physical connectivity and permissions.

Note

If you connect to your messaging world without an Exchange server, certain information may be stored differently by default. For instance, if you have an Exchange server mailbox, your calendar information is probably stored on the Exchange Server. If you use Outlook and have only a POP3 message store, your calendar information is stored in your local folders as the default.

Standalone application

A second configuration is to use Outlook as a standalone information manager for contacts, calendar data, tasks, and so on. Used this way, Outlook is not a messaging application and does not depend on the network or server-side components. Users do not exchange data in this configuration and miss much of the utility built into Outlook.

Exchange client

Outlook 2000 can be configured for use with either Exchange Server 5.5 or Exchange 2000 server or in a mixed environment of both. When a client’s server has been upgraded from Exchange 5.5 to Exchange 2000, they can begin using the additional functionality of the new store immediately. A mixed mode environment has more effect on administration than it does on client operations.

This chapter will focus on Outlook as an Exchange client. From a configuration standpoint, this means that you chose the Corporate or Workgroup option the first time you set up Outlook. You can change this option if you go into Tools/Options, switch to the Mail Services tab, and click on the Reconfigure Mail Support button.

Outlook as an Internet client will be covered in Chapter 24.

Configuration Options

Unless your Outlook deployment includes the automatic creation of profiles, the first post-install task to be completed is to create an Outlook profile. Because this is so important and so few people take advantage of it, we will examine it.
If you configure your client in the Corporate or Workgroup mode, many features require connection to the Exchange server the first time you set them up. To avoid being frustrated, set up and configure all the features that might be used later while you are on the road while you are still in the office connected to the server. If you cannot accomplish it while in the office, try to set up when you are dialed in and authenticated. This is especially true of remote features, rules, and folder sharing and similar features.

Profiles

As we have already discussed, Outlook can be configured in a number of different ways depending on a user’s needs. A specific configuration of Outlook is stored in what’s called a profile. The profile consists of a number of information services and related configurations. An information service is used for connecting Outlook to another information or management system. Using combinations of different information services and multiple profiles allows for an incredible amount of flexibility with the application.

Why use a profile? The profile sets up the configuration settings for using the application successfully in a specific environment. In many cases, a user will have different environmental needs depending on whether they happen to be at work, at home, or working on the train in the morning commute. The type of information services available will vary, network access will vary, etc. Profiles are useful in computer sharing scenarios, too. For example, users may share a PC in the reception area of a hotel, in which case it makes sense to have profiles for each individual using Outlook on the machine. Each user can then customize Outlook to look and behave according to their personal preferences. Profiles can also be used by Administrators to “clone” an environment for multiple users. Using profiles in this way ensures consistency in the environment and makes client support easier.

Creating a profile

You can create a profile in a number of different ways. If you’re upgrading from a previous version of Outlook, your profile will be carried over, and you shouldn’t have to do anything except modify it as appropriate.

Creating a profile after a new installation

If you’re performing a new installation or you’ve never used Outlook before, you will be prompted to create a profile the first time you start the application. A wizard will step you through the process of creating the profile. This is also where you get to choose if Outlook is going to be an Internet client (POP3/IMAP) or a Corporate or Workgroup (Exchange server/POP3/IMAP/etc.) client. Some of the other important items that should be configured are listed below:
Corporate or workgroup: Unless you plan on using only the POP3 or IMAP4 features of Exchange, you should choose Corporate or Workgroup. Once you have chosen to use Outlook with Internet client support or as a Corporate or Workgroup client, all profiles will use that setting. If you choose Internet mode, see Chapter 13 for POP3/IMAP4 service configuration information.

Exchange Server service: Assuming you have configured the client in Corporate or Workgroup mode, you will want to specify the server where your mailbox resides (in the server field) and the name of the user’s Exchange alias (in the mailbox field).

Outlook Address Book: Most organizations will have the Outlook Address Book as a standard service. No configuration is necessary.

Mobile option: When asked “Do you travel with this computer?,” what you are really being asked is whether or not to enable offline folders for this profile. If you choose “Yes,” offline folders will be enabled.

If a service does not show up in your information service options, you can add it after finishing the profile creation.

Creating an additional profile or reconfiguring an existing one
To add profiles or access existing profiles, go to the control panel and choose Mail, or Mail and Fax, depending on the operating system you are using. Choose Add to create additional profiles, or Properties to reconfigure an existing profile, as shown in Figure 12-2.

Figure 12-2: Profile dialog box

Automatic creation of profiles
In addition to a manual process, there are automated tools — such as Profgen.exe — for creating profiles based on the user’s account name.
Using the profile

Once you’ve created your Outlook 2000 profile, simply start Outlook to start using it. By default, Outlook will use the profile you’ve created. If you want to use multiple profiles, you can have Outlook prompt you to specify what profile you want to use — just start Outlook, go to Tools/Options from the menu bar, and then go to the Mail Services tab. From there, hit the “Prompt for a profile to be used” radio button. Figure 12-3 shows the dialog box you will use to enable this option.

![Figure 12-3: Profile property page](image)

Data storage

Users can store information in the Exchange store, personal folders, public folders, and offline folders. The data storage location will depend on several decision criteria, such as security requirements, accessibility, and backup requirements.

Exchange Store (Private and Public)

To store data in the Exchange Information Store, a profile needs to be created that contains Microsoft Exchange Server information service. You can only access this service if you have configured the client to support “Corporate or Workgroup” services. If this is not one of the information services available, you must start the application and choose Tools, then Options from the menu. Next, choose the Mail Services tab, click on the Reconfigure Mail Support button, and choose the Corporate or Workgroup option, as illustrated in Figure 12-4.
Users should always be encouraged to store their information in the Exchange store when possible. If personal folders are used as the default storage location, many of the collaboration options will not work, backups will be more difficult, and security options are limited. In addition, by choosing PST as the default storage, many server-provided features such as indexing and single-instance storage are rendered ineffective.

**Personal folders**

Users can choose to store their data in personal folders. Here are some situations where PSTs are very useful:

- **Archive**: Some use the PST as an archive. Drag and drop, the Export Wizard, Auto Archive, and individual folder aging can all be used to move historical information to a PST file.

- **Offline use**: Instead of using offline folders and synchronizing, some remote users prefer to download all messages to local personal folders and then work the e-mail later offline.

- **Sending or moving folders with the data**: Personal folders are a feasible way of moving large amounts of mailbox data while preserving the folder structure. There are server-based techniques that can be used as well, but sometimes it is more efficient to export to PST, move the PST, and access it on the destination computer. Folders with custom design elements can be sent this way. For example, a developer could send folders with custom event scripting in a PST and they could be imported to a public folder at the client location.

When deciding whether or not to use personal folders, consider these facts:

- **Collaboration is limited**: By definition, information in personal folders cannot be accessed by more than one person at a time. As a result, online folder sharing is not possible.
✦ **Security is limited:** The only security the PST offers is password protection and encryption. Password protecting a PST file is problematic because there are programs on the Internet that are easily downloaded and designed to discover the password.

There is a program called pst19upgd.exe that has been bumping around the Internet. Ostensibly, it was built to upgrade and fix a PST from one version to another, but in the process, it removes the password and makes a copy. We are not clear what the origin was or who, if anyone is supporting this program.

✦ **Backups are more difficult to manage:** Since most PST files are stored on the client computer, backups of PST files usually depend on the diligence of the user. IT administrative staff should coordinate backups of this data on a periodic basis.

✦ **Limited access to mail:** If your users use more than one client machine, they will not have the mail that has been downloaded into the PST if it was downloaded to the other client machine.

To add a set of personal folders, simply go to the properties of an existing profile, choose Add, and then choose the Personal Folders information service. Figure 12-5 depicts a profile that includes personal folders.

![Figure 12-5: Personal folders](image)

**Tip**

You may want to have multiple personal folders. When you create an additional personal folder, make sure you rename it so you do not have three folders called personal folders in your side panel. We tend to use the filename. Also, as we’ve mentioned previously, make sure you control where it is being saved and remember to back it up frequently.
Offline folders
Offline folders can be used to access Outlook data when there is no network connection to the Exchange server. Essentially, the user decides which server-based folders should be available offline, specifies the filename to use, and then specifies the synchronization behavior. It is important to note that whatever happens to the offline folders, the data is still available on the Exchange server due to the fact that information is copied to the offline store rather than moved, as in the case of personal folders. The offline store has an .OST extension.

Caution
Offline folders are useful, but can be challenging to recover from in the case of a catastrophic client failure. Recovery is difficult because there are links between the Outlook profile, the Offline folder file, and the Exchange mailbox that are difficult to recreate. Offline folders can also be difficult when the user has more than one client machine, since offline data is being stored in more than one place.

When users can’t access information in their offline folders, it’s possible that the folders have become corrupt. The user might see persistent errors in the synchronization log or a message indicating that the offline file is not valid. If so, the errors can be fixed using these techniques:

- Running a scandisk to look for problems in the file system and then using the inbox repair tool (scanpst.exe) to attempt to fix the problem. The utilities can be run multiple times and return better results in some cases.
- Using a new tool called the “OST integrity check tool” that was built for Outlook 2000. This tool will attempt to fix offline data integrity problems and persistent synchronization issues.

Another issue arises when the profile is deleted or becomes corrupt due to a modification. OST files are tied to a person’s Windows profile, which is stored in the registry. If the Outlook profile that was configured to use the offline folder is deleted, you have to restore it in order to access the information. If you are running Outlook on a Windows 98 machine, it is possible to start in DOS mode and use the scanreg.exe utility to find and restore a system registry previous to the change or corruption. On an NT machine, you would need to restore a valid backup of the registry prior to the issue. Obviously, this may cause other problems, so make sure you have backed everything up before attempting this procedure. If you have to recover from scratch, create a new profile and set up a new offline folder. All of your previous customization—including rules, favorites, synchronization settings, and so on—will be lost.

Additional Tools menu options
Outlook has an amazing array of options available for customizing the application, enabling you to control the way information is organized and managed, how the application responds to various events, and how the data is presented. Since there are so many options, it is not possible to cover every check box and radio button on every property page. We will cover the most likely configurations and useful customizations.
Auto signature

The auto signature is a great way to tag your messages with a personalized signature. The signature can include rich text, special characters, and even a virtual card file (vcf). The profile can be configured to attach a specific auto signature to new messages only or to every outbound message—it can even be configured to store one or more signatures to be used at the user’s discretion.

Choose Tools➪Options/Mail Format and then click on the Signature Picker button to create and edit signatures. Figure 12-6 shows the dialog box for editing a signature.

If you send messages in HTML or rich-text format (an option in Tools/Options/Mail Format), you can format the signature with rich text to make the signature more appealing.

You may want to include an Internet business card (vCard) in the signature. The vCard is a specially formatted contact that is stored in personal information devices such as Palm Pilots. You should be aware that vCards present certain compatibility issues. In Exchange 5.5, the VCF created an X.500 address by default that was Internet-unfriendly. Also, some non-Outlook e-mail clients will display the vCard as text, which can look awkward.

To create a vCard, first create and save a contact that contains all the information you want. Make sure you fill out the e-mail address field manually instead of choosing from an address book. From within the contact, go to File and then Export to vCard File. By default, it will save your vCard to the signatures directory, where it will be available to attach to a signature.
Views

Outlook views enable users to customize the presentation of Outlook data. Many useful views are already provided with Outlook. Views can be created for use on personal folders, Exchange store folders, and public folders. It is easy to configure views for different data types such as contacts, messages, or calendars. Once the view is created, it can be used in a number of different ways.

Creating and customizing views

Views can be accessed through the View/Current View menu in Outlook. You can choose one of the views provided, or choose Define Views from the Current View menu. From the dialog box that appears, choose New to create a new view.

When creating a new view, you can define the type of view, fields to be included, the sort order, grouping (by category, date, importance, etc.), and who the view is available to. The table view is the most common type of view, but your choice will depend on what type of information is in the folder and how it is best presented. You can restrict the use of a view to a single folder, or make it available for all folders. Figure 12-7 shows a view summary.

![Figure 12-7: View summary showing the configuration of a specific view](image)

Views can be useful when custom applications or special folder design is in use. Many custom applications use custom-defined fields, which means that the Outlook-supplied views are insufficient.

Once you’ve defined the view, you can set it as the initial view for the folder. To set a view as the initial view, highlight a folder, right-click, go to Properties, and choose the Administration tab. From there, choose the initial view from the drop-down list.

If you choose to customize a default view, you may want to reset it later. If so, go to View ➤ Current View ➤ Define View from the menu and highlight the view you have changed. To reset the view, simply choose the Reset button.
AutoArchive

AutoArchive is a feature that will take historical information and move it to personal folders of your choice. The feature is accessed by choosing Tools  Options, moving to the Other tab, and choosing the AutoArchive button. The AutoArchive dialog box will appear, as shown in Figure 12-8.

![AutoArchive Dialog Box](image)

**Figure 12-8:** Configuring AutoArchive to assist in managing Outlook data

AutoArchive can be a strain on e-mail admins. Oftentimes, a user accidentally clicks Yes to the “do you want to autoarchive now” without actually reading the message. This often results in immediate calls to the help desk. Putting a policy in place and training end-users on appropriate use is a smart way to preempt some of those calls.

The AutoArchive dialog box is fairly simple to understand. Just set the frequency (in days) of autoarchiving, choose whether or not to be prompted, decide whether or not to delete expired mail, and choose the PST file to use.

Administrators may want to have the PST files stored on a file server for backup purposes and for access from roving users. If you choose to do this, also consider that additional network traffic will be generated as the AutoArchive process is in progress.

The expired mail option will only delete mail that has the expiration date defined. Most user-generated mail does not have an expiration date. To override the default AutoArchive time on an individual folder, right-click on the folder, go to properties, and switch to the AutoArchive tab. You’ll notice that the Contacts folder does not have an AutoArchive tab, since AutoArchive does not work with the Contacts folder by design.

Working remotely — What are my configuration options?

There are a variety of options for configuring Outlook to be used remotely. One of the main goals of any remote solution is ease of use, and Outlook does a great job in this area.
Offline and synchronizing
The most common way to configure Outlook to be used offline is to enable offline folders. Using offline folders is advantageous for many reasons. Since data is copied from the Information Store instead of moved, the mailbox data is backed up on a regular basis. Also, offline folders can be synchronized with public folders, which is an option not available with personal folders. But, offline folders can have only one synchronization partner. Consequently, if you use multiple physical machines, you may not be a candidate for offline folders.

There are two ways to enable offline folders.

✦ First time you start Outlook: If you are starting Outlook for the first time, the Profile Wizard will ask if the user travels with the computer. To enable offline folders, choose Yes. This action will enable offline folders in the profile.

✦ When the profile has already been created: If a profile exists, start Outlook. Once in the application, go to Tools, then Options from the menu, and then go to the Mail Services tab. Next, check the “Enable offline access” check box and optionally choose synchronization options for all folders, as shown in Figure 12-9.

![Figure 12-9: Configuring synchronization options for offline folders. Additional configuration can be done on an individual folder basis.](image)

The offline folders can be set to synchronize periodically in this dialog box. This works well with dial-up networking, and is a good option for telecommuters. From the home office, Outlook will periodically open a dial-up networking connection or will use a connection that you have manually started. These options can be

Once you’ve enabled offline access, you have to decide which folders to synchronize, how often to synchronize them, and how to synchronize the Offline Address Book. Synchronization options other than those in Figure 12-9 are accessed through the Tools ➪ Synchronize menu option.

First, download the Address Book by choosing Tools ➪ Synchronize, then Download Address Book. You will be asked which address book you wish to download and whether you would like the full detail, as shown in Figure 12-10.

Choosing the “No Details” option will cut the download time considerably and produce a file that is smaller but will not provide much information about the users in the Address Book.

The server rebuilds the OAB periodically according to a schedule set by the Administrator. The default schedule is set to rebuild the list once daily. Also, a user may not choose to download the OAB every time they connect to the Exchange server. The combination of these factors can introduce a significant amount of latency to the address book synchronization process. This can be a support issue if the server-based address list has frequent or significant changes, such as in a directory migration scenario. In these cases, clients should be encouraged to download the OAB more frequently. Sometimes Administrators address this issue by scheduling address changes so many are made at once followed by client downloads of the OAB.

Once you have the Offline Address Book built, it is time to configure folder synchronization behavior. Once again, go to Tools ➪ Synchronize and choose Offline Folder Settings. From here you can access all of the synchronizations options for all folders or for folders configured for quick synchronization (by default, Mail and Calendar). Figure 12-11 shows the Offline Folder Settings dialog box.
The options available for synchronization in Outlook 2000 are much more extensive than previous versions of Outlook. Filters, quick synchronization, and incremental OAB synchronization are just some of the new features added for more convenient remote use.

The Settings button in the Offline Folder Settings dialog box allows you to configure synchronization options for the Offline Address Book. Aside from the address book and mailbox data, public folders can be configured for synchronization as well. Once offline folders are configured, the Tools ➤ Synchronize menu options can kick off a manual synchronization.

In general, the first synchronization should be done when connected to the LAN.

**Offline and using Remote Mail**

Remote Mail is another option for working remotely. Remote Mail is simpler than working offline and synchronizing but does not offer as much control over downloading mail. It works in conjunction with dial-up networking to allow remote connection using any supported dial-up networking method such as VPN, analog modem, or ISDN.

Remote Mail is particularly effective when there is only a low bandwidth connection available. It allows you to download the message headers, mark only the items you want to take off the server, and then download the marked items.
Several of Remote Mail’s components need to be configured before it can be used. Here’s a short checklist:

✦ **Configure offline folders for use:** Please refer to the beginning of the *Working Remotely — What are my configuration options?* section of this chapter.

✦ **Synchronize all folders:** Go to Tools ➪ Synchronization and choose All Folders.

✦ **Download the Offline Address Book:** Go to Tools ➪ Synchronization, choose Download Address Book, and download either version of the Address Book. This will allow you to address messages offline.

✦ **Configure and test a dial-up networking connection:** This process will vary depending on your operating system, hardware, and network systems infrastructure. Whether you use VPN, modem, ISDN, or another dial method, make sure you can connect to the network, authenticate with your Windows 2000 user account, and ping the Exchange server by name.

✦ **Configure and test Remote Mail:** Please refer to the following section.

To configure Remote Mail, go to Tools ➪ Services, highlight Microsoft Exchange Server, click the Properties button, and switch to the Dial-Up Networking tab. This tab will configure how Outlook connects to the Exchange server when you are working offline and choose to synchronize, or start a Remote Mail session. Choose the dial-up networking connection to use to make the connection, or choose to use the existing network connection if you want to manually connect to the network before initiating a connection to the Exchange server.

Next, go to the Remote Mail tab of the Exchange Server dialog box (shown in Figure 12-12) and configure your Remote Mail connection options.

![Figure 12-12: Configuring Remote Mail options](image)
You can configure a scheduled connection or connect manually by choosing the Connect option from the Tools/Remote Mail menu. You are only able to initiate a Remote Mail connection if you are working offline.

Once the connection is made, the client should begin downloading message headers. When the headers are downloaded, a toolbar appears that allows you to mark items for download. In Figure 12-13, you can see that the smaller messages are marked for retrieval (you can see the marked-for-retrieval icon in the left-most column in the view), while the larger message is not marked. You can also see that we have added the size column to the default view for easier identification of large messages.

![Image](image.png)

Figure 12-13: Looking at messages and message headers marked for download

It is easy to see how this method is particularly effective in a low-bandwidth situation. Once the message is retrieved, the header disappears and the full message with any attachments is displayed.

**Offline with PST**

A third option for remote networking is to configure messages to be delivered to personal folders. In this configuration, the information is moved off the server and is available offline. If you are an Administrator, make sure your users are aware of the dangers involved and are willing to take reasonable responsibility for their data.

To configure messages to be delivered to a local PST, go to Tools ➪ Services and switch to the Delivery tab. Here, specify that new messages should be delivered to personal folders. After this change, Outlook will notify you that your default delivery location has changed and ask if you want to rebuild the Outlook bar to point to your personal folders. If you choose to rebuild the Outlook bar, the shortcuts will now point to folders in your personal folders instead of folders in your mailbox.
To install an Offline Address Book, you need to download it by going to Tools ➪ Synchronization. Depending on the size of the Address Book, it may be prudent to do this from a high-speed (LAN) connection the first time.

**Outlook Utilities**

Outlook 2000 is rich with utilities that help users manage and personalize their e-mail—for example, many of the tools have wizards to make them user-friendly. Although people buy Outlook for these utilities, many end-users are never exposed to them. Many an Administrator has gotten kudos for training end-users (especially executive staff) on these utilities.

**Assistants and wizards**

Wizards and assistants are used for executing common tasks. Wizards are usually applications or parts of applications that help you accomplish some task, like configuring a profile, by asking you a series of questions. Assistants are usually code that pops up with a character offering advice when you seem to need it or code that acts on your behalf automatically per some prescription you control. Some of the more common assistants and wizards are covered below.

**Rules Wizard**

Rules are an excellent way of filtering and organizing incoming information. For users that deal with a high volume of incoming messages, the Rules Wizard is a lifesaver. To use the Rules Wizard, go to Tools ➪ Rules Wizard. Choosing New will bring up the dialog box shown in Figure 12-15.
In the upper half of the dialog box, decide what type of rule you want to create. Next, fill out details that are underlined in the bottom half of the dialog box. Click on the underlined field to specify the value. Click Next and you will be prompted for additional information on how this rule works, exceptions, and so on. Once Outlook has all the information it needs to configure the rule, you will have the option to turn on and run the rule. You will see the logic that the server uses to process messages based on this rule and will be able to go back and fix it if it doesn’t work properly.

It is important to test rules. You should also be aware that some rules are only activated when the client is online and connected to the server. Client-based rules will be marked “Client-only” by the Rules Wizard, and you will be notified of this when they are created.

Users, even executives, can create self-inflicted wounds with this useful tool. An executive we were supporting called and said his own Administrators were useless: right in front of his eyes, his e-mail was disappearing as it was received. We explained to him and his Administrator that a rule must be moving (or worse, deleting the e-mail). Sure enough, the executive had created a rule on Friday and forgotten about it on Monday.

**Office Assistant**

Anyone who has any familiarity with the Microsoft Office suite of products is familiar with the Office Assistant — the paperclip guy (other characters, like Einstein, can be substituted from your installation CD) that everyone turns off the first time they use the product. While it is a somewhat interesting distraction, the Assistant functions much the same as other help modules. It provides easy access to some of the common task walkthroughs, answers free-text queries, and occasionally winks at you. It can be accessed through the Help menu, Help button on the toolbar, or by pressing F1.
Out of Office Assistant

This handy assistant will help you when you plan on being out of the office. Access the Assistant by choosing Tools ➤ Out of Office Assistant from the menu. A dialog box will appear, as shown in Figure 12-16.

You can set up rules for copying or forwarding messages, reply with custom text, and many other actions. In general, the Out of Office Assistant is a great way to make sure your business is being handled while you are away and to provide contact information to colleagues for urgent issues.

Import/Export Wizard

The Import/Export Wizard is an extremely useful feature. It provides the Administrator with an automated tool for exchanging data with other common user applications. The Wizard supports a variety of application data ranging from Lotus Organizer to ACT! The Wizard also allows control over the way data is exchanged. For example, if you are exporting messages to an archive file, the Wizard presents message filter options and asks how it should handle duplicate items.

A common use for the export wizard is to export your own contacts for use by others or into a PST.

Find capability

Once users begin taking advantage of the extensive feature set in Outlook, the amount of data in use can increase rapidly. As we have discussed previously, there are multiple ways to organize and present the data. Despite our best efforts to organize, there will occasionally be a need to search through the data set and find specific items. Outlook supplies a robust find utility that can be used to search through
any of the storage locations that the user has permission to access. This will include personal and offline folders, public folders, and the Exchange store.

**Accessing the find utility**

There are multiple ways to access the find utility. Here are the most often-used access methods:

- Click the Search button on the toolbar
- CTL+SHIFT+F on the keyboard for advanced find
- Right-click for an advanced find on specific folders

**Search criteria**

Once the find utility dialog box is open, there are many search criteria that can be used to extract very specific information. Some of the criteria include header and message content, sender or recipient, date, timeframe, and importance.

![Tip]

Sometimes Find is not the best way to get the information you want. Often doing a sort will get you what you want faster. To do a number of different sorts, click on the buttons above the message headers. If you find yourself using column sorting a lot, you might want to consider using different default views.

**Authentication and Bandwidth Profile**

Outlook is an application with many features that depend on server-side components. As a result, it has a fairly large footprint on the network. The application bandwidth usage depends very heavily on how the application is used. In general, Outlook 2000 is more bandwidth-efficient than other Exchange 2000 client applications on a task-by-task basis. Listed below are some of the Outlook processes that utilize server resources.

**Starting up and authenticating**

When a user starts up Outlook, several processes occur, such as name resolution, authentication to the Windows 2000 directory, and connection to the Exchange 2000 Information Store. Each of these processes generates traffic on the network.

**Store usage**

The network is used every time the Information Store is accessed to send and receive messages, tasks, appointments, and so on. This includes offline folder synchronization, remote mail header downloads, and marked message retrieval. Traffic is also generated when a user accesses public folders.
Directory access
When a client utilizes the directory service online—including downloads of the address book, addressing a message from a server-based address list, and viewing recipient properties or searches—traffic is put on the network.

Free-busy data
Periodically, free and busy information is updated on the server. Free and busy information is copied from the client to a public folder stored on one of the servers in the client’s site and is used for planning meetings and other calendar-related tasks.

Persistent finds
If a user leaves an Outlook search dialog box open when they exit, the search will be refreshed when a new session starts. This is referred to as a persistent find. Of course, refreshing searches can consume a great deal of bandwidth depending on how many searches are being refreshed and the type of searches being performed. Users should be encouraged to find other methods of extracting and organizing data from the server.

More information
While there are no hard numbers for utilization due to the number of protocols and usage characteristics, certain network utilization studies can be used for reference. There is an excellent technical paper available on Microsoft’s TechNet Web site called “MS Exchange Implementation: Client Traffic Analysis.” Just go to http://www.microsoft.com/TechNet/exchange/technote/trafanal.asp

Dealing with Security and Firewalls from the Client
Anytime remote users need to be supported in an environment, security and complexity become issues. Exchange servers are almost always located within the boundaries of the secure network. Most security administrators will not allow remote Internet connection directly to the Exchange server since direct remote access can introduce a great deal of risk to the overall network.

The alternative is to have clients connect to another device, such as a VPN server or other Remote Access server, and then access the Exchange server via a routing mechanism. The client will connect to an external network interface and is then routed to the internal network through a proxy or other means.
Normally, there is not a lot to do on the client except install and configure connectivity hardware or software that allows them to connect to the remote access server. This can include VPN software, an authentication smart card, or special dial-up software that works with a locally installed analog or ISDN modem. Usually, you will have to configure the network address or dial-up number of the remote access device, provide logon credentials, and configure authentication and encryption.

Security is an issue in almost any computing environment. Outlook data is often sensitive and company-confidential. You can ensure your data is safe by

✦ **Turning on encryption:** You can turn on basic RPC encryption to encrypt client-server MAPI sessions by going into the Exchange server properties, switching to the Advanced tab, and choosing “Encrypt data while using the network/while using dial-up networking.”

✦ **Keeping your application up to code:** Make sure you check Microsoft’s Web site for security updates. Periodically, improvements are made to the application that are specifically designed to address security issues.

✦ **Keeping data in the Information Store:** The Exchange 2000 Information Store is the most secure place to store Outlook data.

✦ **Using Best encryption for PSTs:** If you have to use personal folders, use the “Best encryption” option when you create the folder. You can’t change this once you’ve created it, so you’ll have to use Import/Export to move the data if you want to change the encryption options.

✦ **Use advanced security:** The best insurance for securing your data during transmission is to use Key manager or a third-party certificate server. This requires careful planning and configuration on the server.

## PIM Quickstart for Your End-Users

The amount of personalized information that we use to organize and optimize our daily business activities is increasing. As a result, it is becoming more important to have convenient ways to access, update, and manipulate this information. Personal Information Managers (PIMs) are rapidly gaining popularity as their manufacturers and independent software developers respond to the call for more features, better ergonomics, and improved interoperability.

Generally, users that purchase PIMs also use laptops or desktop computers. As a result, they end up using multiple applications for very similar kinds of information. Many early-version PIMs lacked support for exchanging data with other programs, and early synchronization programs were awkward and severely limited. However, with constantly improving hardware and supporting software, it is increasingly easy to synchronize your Outlook 2000 data with many popular hand-held, wireless, and other devices.
**Best practices for all devices**

Since PIMs are rapidly evolving, it is important to make sure that all your hardware and software is compatible. Do yourself a favor and research proven configurations on the manufacturer’s Web sites. Match your PIM with the correct BIOS, operating software, information management applications, and synchronization software.

- **BIOS**: It is important to have the correct version of the BIOS. Device manufacturers often release new BIOS versions that will enhance performance and enable or disable system features. Usually it is best to use the most current version of the system BIOS, and often updating a system BIOS will correct problems. If your device is working properly and a newer version of the BIOS is released, make sure you research known issues with the BIOS and read the release notes before you decide to update.

- **Operating software**: The operating software options are dependent on what type of device you have. Often this choice will determine the applications you can use on your device. Normally there are only one or two varieties of operating systems you can use, but periodic version upgrades will be released and should be researched and implemented if appropriate. The Palm OS software and WindowsCE are examples of common mobile device operating software. Sometimes upgrading the software for the OS will correct problems, but make sure you research known issues with your applications and device.

- **Information management applications**: This is the software application that manages the information in your PIM. There are many applications developed for PIMs, and you should periodically check the vendor Web site for new ones. Your choices here will depend on what device you have chosen and what OS it is running. As with all components in a PIM environment, version is important, so spend the time to find out known issues, compatibility problems, and tested and approved configurations.

- **Synchronization software**: Often, multiple synchronization programs will work with your device and operating software. Depending on the applications on the PIM and what type of data is being synchronized, your choice of synchronization software will vary. There are many synchronization applications that support Outlook, such as Intellisync, PocketMirror, and CompanionLink.

**Connectivity options**

There are a variety of methods — including RF, Serial cable, USB, and InfraRed — for connecting PIMs to other systems. Some devices can use more than one connectivity option, and you might try multiple methods to find out which works best.
Synchronization options

Depending on your setup, there are many synchronization options available that can help you manage your data better. You will want to consider:

✦ **Type of synchronization:** Many programs allow you to set up synchronization so that
  - The hand-held device overwrites your server-based mailbox
  - The mailbox overwrites the hand-held
  - The hand-held and mailbox are synchronized

On the first synchronization, carefully consider the synchronization type. It may be advantageous to overwrite instead of fully synchronize, to avoid duplicates and start with clean data. Be careful though, since data may be deleted. Consider backing up your data to personal folders before the first synchronization.

✦ **Custom fields:** Some synchronization software allows you to map fields from one application to another. If you have useful data stored in non-standard fields, consider using the mapping function to maintain the usefulness of the field.

✦ **Filters:** Consider using filters to organize and manage your data. The way you enter your information may have a substantial effect on the filters you can use. Consider using some of the fields that you had not previously used to make filtering more effective. For example, you might want to start using categories to filter personal contacts from being synchronized.

✦ **Backups:** Treat your PIM information like any other personal data. It is good to have a backup or export file to recover from if your data becomes damaged. Most PIMs allow you to back up or export your data to a file, so add this to your regular maintenance routine.

**PALM**

The Palm Pilot is a very popular pen-based hand-held device that integrates well with Outlook 2000. There is an impressive list of applications that have been developed for this device, and it has been around for quite some time. Using the PocketMirror synchronization software and the Hotsync manager that ships with the device, it is possible to synchronize calendar, contacts, tasks, and notes with your Exchange 2000 mailbox.

**WindowsCE and PocketPC**

Several devices use the WindowsCE operating system. These include PocketPCs, hand-held PCs, wireless devices, and other mobile organizers. The most common synchronization software for WindowsCE devices is a product called Intellisync by Puma Technology. It will work with devices running WinCE versions 2.x and above and usually provides synch for various components of Outlook like mail, contacts, and schedule.
Blackberry from Research in Motion

Blackberry is one product in a family of wireless products from a Canadian company called Research in Motion. Blackberry is a family of wireless devices that store and manage e-mail, contacts, and tasks. Some are in the form factor of a pager and others are larger. They operate very well with Exchange 2000 and have the capability to synchronize information with a user mailbox. There are two ways to use Blackberry with Exchange and Outlook. In smaller implementations, there is a desktop redirector that intercepts, compresses, and encrypts data and sends it through the wireless network to the device. In larger installations, you can run an Exchange service that does the redirection, alleviating the need to have your desktop machine on.

Wireless Access Protocol

Though still in development, utilities are emerging that allow Outlook 2000 to publish contacts, appointments, and tasks to a Web page. The data is published in a special format that can be accessed by a Wireless Access Protocol (WAP) compatible wireless device. There are other utilities that forward messages to hand-held WAP devices, which can be tremendously useful to the mobile user.

Summary

This chapter discussed the important aspects of Outlook 2000. Outlook was revealed to be a flexible, multi-faceted application that integrates well with Exchange 2000 and the Office 2000 application suite. You learned techniques for managing the data by using built-in features such as wizards, assistants, filters, and views. We covered online and offline data storage and security options and explored PIM devices that can extend Outlook information beyond the workstation.
Outlook Express

Outlook Express, which is installed in conjunction with Internet Explorer, is designed to utilize Internet-based mail, directory services, and newsgroups. It also functions as an Exchange 2000 client application, and in some situations is an excellent alternative to the full version of Outlook 2000. In this chapter, we explore features in the application and describe situations where they are suitable. We also cover basic setup and configuration tasks, security, and remote operations.

Components and Features of Outlook Express

Outlook Express may not have all the bells and whistles that the Microsoft Outlook client has (or the same profile for authentication or security that Outlook has) but it is feature-rich and very competitive with other popular e-mail clients, such as Eudora. For many users, it will do the job they are trying to do more than adequately. For almost every administrator, it is part of the toolkit. It also has a great price. The Outlook Express application is included free with Internet Explorer, and, depending on what you are connecting it to, may require no Client Access License (CAL).

Client application and basic features

The application has improved with each new version of Internet Explorer; for example, the application with Internet Explorer 4 and higher includes everything needed to configure mail, directory, and newsgroup services.

Note

In writing this book, a variety of versions of Internet Explorer were used, including 5.5. We suggest you make efforts to stay current with Internet Explorer and Outlook Express. You should include currency in updates and service patches to help maintain the security profile of your system. That is where Microsoft patches holes that are discovered as hackers attack.
Major features of Outlook Express, and their corresponding functionality, are presented in Table 13-1.

<table>
<thead>
<tr>
<th>Feature (Protocol Used)</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-mail (POP3/SMTP and IMAP4)</td>
<td>Send and receive electronic messages using standard Internet mail protocols</td>
</tr>
<tr>
<td>E-mail (HTTP)</td>
<td>Send and receive mail using HTTP and your WWW browser</td>
</tr>
<tr>
<td>Newsgroups (NNTP)</td>
<td>Search through, read, and post newsgroup messages by connecting to NNTP servers</td>
</tr>
<tr>
<td>Directory services (LDAP)</td>
<td>Access rich directories that support Lightweight Directory Access Protocol (LDAP) access for public Internet user information or proprietary internal directories</td>
</tr>
<tr>
<td>Extensive Address Book</td>
<td>Import and export addresses in common formats, access personal and shared contacts, and utilize LDAP directories</td>
</tr>
<tr>
<td>Multiple Identities</td>
<td>Use identities to enable multiple users to have individual password-protected application settings and services on the same computer</td>
</tr>
<tr>
<td>Multiple Accounts</td>
<td>Create multiple accounts for mail, newsgroups, or directories. Removes need to use multiple instances and applications</td>
</tr>
<tr>
<td>Offline Synchronization</td>
<td>Offline address and contact information, drafts, messages, and so on can be synchronized with certain server-based services</td>
</tr>
<tr>
<td>Message Rules</td>
<td>Create custom rules for controlling how messages are processed and managed</td>
</tr>
</tbody>
</table>

**Modular services**

Like Outlook itself, Outlook Express uses services that are configurable to access information stores and provide other utility to the application. The specific services that are considered appropriate depend on
What the user wants
✦ How Outlook Express will be used
✦ What connectivity methods are available

The services described in the following sections are designed to be flexible and can be configured to fulfill a variety of personal, business, and technical requirements. They can be added and removed as needed, or used in conjunction with identities to customize the application for multiple users.

**POP3**

Post Office Protocol, version 3 (POP3) is an Internet e-mail standard that has been around for quite a few years. This service is supported by applications running on just about any operating system platform you can think of. Because it was one of the first, widely adopted, cross-platform, Internet-based mail services, POP3 is very popular on the Internet. Using POP3, a user is forced to download all messages into a local folder after connection and authentication with the POP3 server. The downloaded messages are then accessed from the local machine. This creates an administrative issue in protecting, backing up, and restoring messages stored locally. Once POP3 has downloaded the messages, there may be no copy of the downloaded messages on the server unless you have specifically chosen to leave a copy (using the Advanced Property Page Delivery option) and specified the number of days downloaded messages should remain on the server. The Exchange 2000 Information Store has native support for POP3 and is an excellent POP3 server for Outlook Express and other POP3 client applications.

A POP3 client must have a valid protocol for sending e-mail in addition to POP3, which is used for receiving e-mail. People often get confused about what POP3 does; they think it is a complete protocol for receiving and sending messages. In reality, POP3 is a protocol for retrieving mail only. The send function is usually handled by SMTP. You need to understand this, because it is quite common to be called into situations where sending mail works and receiving mail does not, and vice versa. In many POP3 troubleshooting situations, you must look at both the POP3 protocol and the Simplified Mail Transport Protocol (SMTP) configurations.

**IMAP4**

Internet Message Access Protocol version 4 (IMAP4) is a newer Internet mail standard that is gaining popularity thanks to some of the features it offers over POP3. In particular, IMAP4 enables you to directly manipulate information stored on the server, or work offline and synchronize. This means that unlike POP3, where you are forced to download your e-mail to a local device to access it (sometimes leaving a copy behind on the server), your data can be accessed while remaining on the e-mail server and not be downloaded at all if you choose. Of course, if you want to
download, you still can. Like most of us, your clients may end up using a combination of server-based storage, synchronization, and download. For roving users who use multiple computers, server-based storage is often a crucial feature.

This feature also has positive implications for administrators. Backing up locally stored data can be a challenge, as we mentioned in the section on POP3—especially in conditions where users are mobile. By storing information that the user wants protected on the messaging server, your job is simplified.

Like POP3 clients, IMAP4 clients use SMTP for sending messages. The SMTP server must be properly configured and available in order to send outgoing messages.

**SMTP**

POP3 and IMAP4 clients cannot send messages using either the POP3 or IMAP4 protocols, because both protocols are for accessing—not sending—e-mail. Instead, in the world of Outlook Express, the client makes a Simplified Mail Transport Protocol (SMTP) connection to a server that accepts the incoming SMTP message. SMTP is the protocol for all transmission from POP3 and IMAP4 clients. The server then redirects the message to the appropriate host server or directs the message to a local mailbox. The SMTP service can reside on the same machine as the POP3 or IMAP4 server, or on a separate computer.

The POP3 or IMAP4 servers do not have to be part of the same company or in the same location as the SMTP server. For example, assume your e-mail is not sending because of a connection problem. You are on the Internet and are able to browse to various Web sites, but Outlook Express keeps saying that it cannot make the connection to send your mail. To troubleshoot, you can configure your POP3 or IMAP account to point to another SMTP server that you know is working for a colleague, as shown in Figure 13-1. There you see that the POP3 server is pop3.email.msn.com. MSN also has SMTP servers. That entry would have been smtp.email.msn.com, but the user was experiencing problems. So the Administrator suggested trying the company ISP (Metro2000) for SMTP. That entry was smtp.metro2000.net. To complete the configuration, the correct authentication credentials were set for the metro2000 SMTP server.

Figure 13-2 shows a user accessing mail from an ISP, and locally from a company-provided POP3 server. Both are accessed by the same client machine and Outlook Express session. One ISP has also chosen to use two different servers to deliver e-mail services to the client. When the user starts the application and begins downloading her POP3 messages, Outlook Express connects to the POP3 mailbox server called Pop3.MyISP.net. The server named SMTP1.MyISP.net is running the SMTP service and will process outgoing mail from the user. The user’s company has chosen to use a single server for incoming and outgoing messages.
Figure 13-1: Configuring a different POP3 and SMTP provider

Figure 13-2: A typical user sending/receiving e-mail (POP3/SMTP)
HTTP
HTTP can also be used as an Internet mail protocol and is sometimes called “Web mail.” The types of services provided by HTTP mail vary widely due to the underlying application architecture. HTTP-based mail is a rapidly evolving application with universal industry support for its anytime, anywhere accessibility and easy client setup. Outlook Express enables you to connect to e-mail providers running HTTP servers to send and receive messages. HTTP is commonly used to connect to some of the free mail systems available on the Internet, such as HotMail from Microsoft.

LDAP
The Lightweight Directory Access Protocol (LDAP) service provides connection to X.500-based directories. Directories can be multipurpose and can store hierarchical information about a variety of objects. When used in conjunction with Outlook Express, LDAP is used to access information about users.

Newsgroups
Newsgroups generally consist of information posted by Internet users, but large companies may also have internal newsgroups. Internet newsgroup servers host public forums where the Internet community can post questions or answers, discuss general or specific topics, or post topic-related information for other users’ education. Newsgroups exist for just about any topic imaginable and provide a rich source of contacts and information.

Installing and Configuring Outlook Express
Outlook Express is extremely easy to install and configure. Installation is performed in conjunction with Internet Explorer 4.x or 5.x. Your first post-install task will be to configure services to use with the application. Without any services configured, Outlook Express will not do much of anything. Once you’ve configured your services, you can start defining identities and configuring security and other advanced customization. Customization is fairly simple—if you have the right information from your Internet Service Provider.

Installing the application
Outlook Express is installed during a “typical” or “complete” setup of Internet Explorer 4.0 or 5.x and can also be installed during a custom install. The “browser-only” or minimal install does not install Outlook Express.

If you already have Internet Explorer installed and didn’t install Outlook Express, it’s easy to add. Run Internet Explorer setup from the Add/Remove Programs icon in Control Panel and mark the check box to install Outlook Express. Once you’ve got the application installed, you’ll want to start creating accounts for e-mail, directory services, and newsgroups.
Configuring mail connections

There are four protocols supported for Internet mail services: POP3, IMAP4, SMTP, and HTTP. Many ISPs offer different types of e-mail services for free, and it is up to the user to decide which type of service best fits their needs. Some e-mail services support access from multiple protocols. To connect to an ISP, you will need to get as much detailed configuration information as possible to ensure the property pages can be correctly configured without having to recontact your ISP.

Look at the property pages for configuring the protocol before you call your ISP so you know what information you will need. Explain that you are using Outlook Express, and ask for the specific information to fill out the property pages. Many ISP support personnel are familiar with Outlook Express and will walk you right through it.

You may also need to know whether there is a fire wall between you and the server you’re trying to connect to. If so, you will need to know which ports are open, because this will affect your choice of protocol.

In general, IMAP4 provides the greatest feature set with minimal network overhead set, so it’s usually a good choice if available. If you are connecting to an Exchange 2000 server using Outlook Express, you’ll most likely be using either POP3 or IMAP4, and many administrators and users will head toward IMAP4 for the greater utility.

The following section discusses setting up a POP3/IMAP4 server. Essentially, the setup is the same, but once you get it configured you’ll have a more extensive feature set with IMAP4. Before you set up POP3 or IMAP4, you must have the following information:

✦ What protocols are supported (POP3, IMAP4, SMTP, HTTP)
✦ The fully qualified name (DNS) of the POP3/IMAP4 server
✦ The fully qualified name (DNS) of the SMTP server
✦ The account name to log in to your POP3/IMAP4 server (if required)
✦ The account name to log in to the SMTP server (if required)
✦ What type of authentication is required or preferred
✦ What restrictions, if any, your fire walls impose on access or authentication
✦ Your e-mail address on the POP3 server

Tip

Have whoever is providing the information for the items in our configuration bullet list FAX or e-mail it to you. This type of information is commonly asked for. It is highly likely they have it in written form. (They probably also have it in e-mail form, but you need it to be able to connect e-mail, so you may need to use FAX instead.) Similarly, as an Administrator, you should prepare sheets that contain all the required configuration information, so that answering people’s questions is a smooth, efficient process.
The e-mail service provider or corporate e-mail administrator should provide this information for you. There are multiple ways to get into the configuration dialog boxes. The first time you set up Internet Explorer on your machine, you are prompted to create a mail account and a directory service. You can create these accounts later, if you choose; you’ll be prompted again the first time you start Outlook Express, and anytime you start Outlook Express with a new identity. In all cases, you will need to provide the same information.

To access the Configuration Wizard from within Outlook Express, select Tools ➪ Accounts from the menu bar. From this dialog box, click the Add button and then choose Mail. The next dialog box will prompt for a name. This name is your display name and will appear in the “from” or “sender” field when it arrives at the destination. This name does not necessarily have anything to do with the e-mail address you use.

Once you’ve filled this field out, click Next. Here you have the option of creating a new account or using an account that is already set up. We’ll assume that you are configuring a connection to an account that has already been set up by your ISP or corporate Exchange administrator. Leave the “I already have an e-mail address that I’d like to use” radio button marked, enter your e-mail address, and click Next.

Now you will choose which mail service to use for this account. The E-mail Server Names dialog box will appear, as in Figure 13-3.

![Figure 13-3: Configuring an IMAP4 account in the Internet Connection Wizard](image)

In the above example, an IMAP server called IMAPsrvr.Exch2K.org stores the mailbox for this user. A different server called SMTPsrvr.Exch2K.org will accept out-bound mail from the client and route it to the appropriate mail server on the Internet. When the information is complete, click Next.
In the Internet Mail Logon dialog box, enter the authentication information (including user name and password) and specify “Log on using Secure Password Authentication (SPA)” only if your ISP supports it. The Internet Mail Logon dialog box is one you will encounter while configuring any type of account in Outlook Express; it is shown in Figure 13-4.

![Figure 13-4: Setting authentication options for an account](image)

Once you’ve entered the authentication information, click Next and then Finish. At this point, the service is ready to use.

Some providers do not use the standard TCP ports for connections, and some require clients to use SSL security (Secure Sockets Layer). If this is the case, an account will need further configuration after the initial setup. If standard port numbers are not being used, your service provider or administrator must provide the correct port number for you to use. The standard TCP port numbers are listed in Table 13-2.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Non Secure Port</th>
<th>Secure Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>POP3 server</td>
<td>110</td>
<td>995</td>
</tr>
<tr>
<td>IMAP server</td>
<td>143</td>
<td>993</td>
</tr>
<tr>
<td>NNTP server</td>
<td>119</td>
<td>563</td>
</tr>
<tr>
<td>SMTP server</td>
<td>25</td>
<td>N/A</td>
</tr>
<tr>
<td>LDAP server</td>
<td>389</td>
<td>636</td>
</tr>
</tbody>
</table>
To change a server connection port number, select Tools ➪ Accounts, highlight the account you want to configure, and click on Properties. Go to the Advanced tab and enter the advanced connection information. Select SSL only if advised to do so by your ISP. See the dialog box in Figure 13-5.

![Figure 13-5: Configuring alternate port numbers and SSL in the advanced properties dialog box of a newsgroup account](image)

### Configuring directory services

Directory services are simple to set up in Outlook Express. The application has some directories configured by default, and you can access them by selecting Tools ➪ Accounts on the menu bar. A dialog box will appear, and you will then need to switch to the Directory Service tab.

If you want to add a new service, select Tools ➪ Accounts from the menu. Click on the Add button, choose Directory Service, and then input the LDAP server fully qualified DNS name. Mark the check box for account logon if the LDAP service provider requires authentication. The Internet Directory Server Name dialog box is shown in Figure 13-6.

If necessary, specify logon credentials and click Next. In the last dialog box, choose whether or not to check for names in this directory when sending messages and then click Finish. The service should now be ready to use.
Internet directory searches

Searches against Internet directory (LDAP) servers can be performed with new criteria and logical operators. If you want to access the directory search function, click on the down arrow next to Find on the toolbar and choose People. If you switch to the Advanced tab, you’ll see the dialog box shown in Figure 13-7. In this dialog box, you’ll have the option to define very specific search criteria.

Figure 13-6: Configuring an LDAP server connection

Figure 13-7: Looking for a user on the Internet
Configuring newsgroups

Setting up a newsgroup is straightforward. The only thing you really have to know is the NNTP server fully qualified domain name. Most newsgroup servers are public, and you may have to search the Internet to find the server names.

To create a newsgroup account, select Tools ➪ Accounts, click Add, and choose News. The wizard used for creating a newsgroup account is very similar to the wizards used for creating directory services and e-mail accounts. You’ll have to enter the following information:

✦ **Display name:** This display name will show up on messages you post to newsgroups.
✦ **E-mail address:** This address will be used in your newsgroup user profile. If other newsgroup users want to contact you directly, they will use this address.
✦ **Server address:** This is the address of the newsgroup server (for example, News.Microsoft.com).
✦ **Logon information (SPA):** Authentication information for connecting to the newsgroup server. Most public servers do not require authentication.

Using newsgroups

To access information in a newsgroup, you must first subscribe to the newsgroup; then you can download message headers or the entire message. After you’ve configured a newsgroup account, you’ll be asked if you want a list of the newsgroups on the server. Some servers have several thousand newsgroups, so downloading the list may take some time. Once you’ve downloaded the list of newsgroups on the server, you’ll see a dialog box, such as the one shown in Figure 13-8.

![Figure 13-8: Subscribing to newsgroups](image-url)
As shown in Figure 13-8, this user has chosen to look at newsgroups that contain the word “computer” and has subscribed to two that are marked with the subscription icon.

Choose the newsgroups you would like to subscribe to and click OK. This will bring you back to the main Outlook Express window, where you can view newsgroup postings. To view newsgroup messages, you have to set synchronization options. In the folders pane of the application, you’ll see newsgroups at the bottom of the accounts listed. If you highlight the server name, such as News.Microsoft.com, you’ll see all of the newsgroups you are subscribed to and their synchronization settings in the right-hand panel. Here you can choose new subscriptions and configure synchronization options. By default, accounts synchronize new messages only, so when you connect and synchronize, only new messages are downloaded.

A reinstall of your client may cause the read message flags previously set to be lost. This will potentially cause vast amounts of information to be downloaded. Roving users may also find that their flags are not maintained properly without some attention to a configuration that actively maintains the flags wherever they are stored.

If you double-click on one of the newsgroups, you will see its contents in the right-hand panel. To post information, click on the New Post button on the toolbar.

Some newsgroups do not allow posting, or they may require you to e-mail your post to a moderator who will approve or disapprove the content of the message before it is posted.

**Configuring and using identities**

To enable multiple users to have individual settings, identities are used. To create an identity, select File➪Identities and choose Add New Identity or Manage Identities. The first identity you create will become the default and will be associated with any services that exist in the application. When you define an identity, you’ll see the dialog box shown in Figure 13-9.
Although you can password-protect an identity, this only provides a moderate amount of security. In some cases, other users may be able to access your information if they log on to the computer with the same Windows account. For better security for Outlook Express data, you should use Windows profiles. Once an identity exists, the application will prompt you to choose one during application startup.

Creating signatures

You can add a personalized signature to any or all outbound messages and newsgroup postings. It’s also possible to create multiple signatures and associate them with different mail or newsgroup services. To create a signature, select Tools ➪ Options. When the Options dialog box appears, switch to the Signatures tab. Click on the New button and fill out the text of your signature, as illustrated in Figure 13-10.

![Figure 13-10: Creating a signature](image)

If you click on the Advanced button, you have the option of choosing which services use the signature. You can create multiple signatures for specific situations.

Advanced security

Advanced security is available in Outlook Express in the form of digital IDs. These enable you to digitally sign and encrypt messages sent by Outlook Express. The strength of the encryption will vary according to the type of certificate you install on your machine.
Obtaining and installing a digital ID
Public certificate authorities, such as Verisign, generally issue digital certificates. These certificates must be purchased and installed by the user or the user’s employer, and the certificate is assigned to a specific e-mail address.

If you have multiple e-mail addresses that need to be digitally signed, you’ll have to obtain multiple digital IDs.

The installation method varies, and you should use the instructions provided by the certificate authority. Once the certificate is installed, the application is aware of it. The certificate can then be assigned to an account created in the application.

Using your digital identification
Digitally signing or encrypting messages can be done globally, or on an individual basis. To encrypt or sign an individual message, begin composing the message and go to Tools➪Digitally Sign or Tools➪Encrypt from the menu. To sign or encrypt all messages, go to Tools➪Options from the main application menu, switch to the Security tab, and configure the appropriate options.

Outlook Express automatically adds digital certificates that others send you to the Windows Address Book. These can be used to verify signed or encrypted messages sent by those users.

If your users are using digital certificates, it is very important that the Windows Address Book be backed up regularly. Otherwise, a change in machines or a loss of a hard drive may make the messages unreadable without recollecting the certificates.

Working Remotely
Most often, users that choose Outlook Express as their mail client are working remotely at least some of the time. The application was built on this premise and has many remote-friendly features available. For example, network auto-detection is integrated with Internet Explorer. If the application does not detect a network presence, the user is prompted to work offline.

Integration with Dial-Up Networking
Many users do not have a persistent network connection available at all times. Outlook Express enables individual accounts to be associated with specific Dial-Up Networking (DUN) phonebook entries on Windows client machines for easy access to multiple network providers. To configure an account to use a DUN phonebook
entry, go to the properties of an account and switch to the Connection tab. You’ll see a dialog box like the one in Figure 13-11.

A discussion about configuring Dial-Up Networking (DUN) and its predecessor Remote Access Service (RAS) requires much more space than we can provide in this Exchange book. However, these topics are covered in some depth in various Windows Administrative Bibles and Secrets books.

![Figure 13-11: Configuring an account to use a DUN VPN connection](image)

To configure application-wide network connection settings, go to Tools ➤ Options from the main application window and switch to the Connection tab. In general, Outlook Express will share its connection settings with Internet Explorer, but you can change settings if necessary.

**Offline support and message synchronization**

Outlook Express data is stored in local folders. Actions performed offline (such as deleting messages and creating new IMAP folders) will be performed in the server-based mailbox when you go back online. IMAP will synchronize everything you’ll need for roaming such as the main mailbox, drafts, and sent items. You can also customize which IMAP folders are visible or hidden (subscribed or unsubscribed) in the folder list.

To access the folder manager, highlight the IMAP account from the folders panel in the application and click on the IMAP folders button. In this dialog box you can configure folder availability, as shown in Figure 13-12.
You'll notice that four of the folders in Figure 13-12 have icons next to them. These folders will be visible and available offline. Like newsgroup subscriptions, IMAP folders can be configured to synchronize all messages, new messages, or message headers only. This can be configured by highlighting the IMAP account in the folders list and using the Settings button in the right-hand panel. It is exactly the same as configuring newsgroup synchronization.

![Configuring IMAP folders for availability](image)

**Figure 13-12:** Configuring IMAP folders for availability

### Authentication and Bandwidth Profile

The authentication options are minimal for Outlook Express. You can’t really configure the authentication mechanism that’s used, but you can use Secure Password Authentication if your service provider supports it. This will encrypt the user name and password before the credentials are sent to the server, so a network trace will not pick up a clear text authentication packet.

In Exchange 2000 Server and Advanced Server NTLM, authentication method must be enabled for the protocol in question in order for SPA in Outlook Express to work.

Network utilization in Outlook Express depends on a number of factors. The type of services used, their configuration and frequency of use are all variables in the network utilization equation. There is no right way to use Outlook Express from a network standpoint, but here are some things to consider:

- **Number of active accounts:** Obviously, the more services you have running and set for automatic synchronization, the more traffic is generated.
- **Data storage location:** Using POP3 forces you to download everything to a local folder whether you maintain a copy on the server or not. If you have the option of using IMAP, you can leave some of the data on the server and not download it at all. For example, with IMAP you can choose to only synchronize message headers and access messages as needed off the server. This IMAP technique can also be used with NNTP accounts. Alternatively, once a message is downloaded by POP3 or IMAP4, it requires almost no time and no bandwidth to open it.

- **Authentication methods:** Secure Password Authentication will cause more network traffic than clear text. If it’s available, you’ll probably want to use SPA despite the marginal increase in network utilization.

- **Advanced security:** If Digital signatures and encryption is being used, more traffic will be generated than if they are not.

- **Blocked senders:** You can use filters to block unwanted messages, such as advertisements, saving precious bandwidth for those messages you do want.

- **Extra bandwidth:** How is your default message configured? Do you send messages in HTML format with stationery, signatures, and read receipt requests? All of these features use extra bandwidth.

One of the main decision points for choosing an Exchange 2000 client is network usage. There are three standard applications you can use for Exchange clients: Outlook, Outlook Express, and Outlook Web Access. Note that these three applications operate very differently on the network. Differences in their network profile include such basic things as protocol use, authentication methods, and Information Store and directory access methods.

From a network utilization standpoint, Outlook Express is generally more efficient than Outlook Web Access on a task-by-task basis. Outlook Web Access is considered by some to be a “lightweight” client application, but that only refers to local resources that are consumed on the client machine.

Compared to Outlook 2000, Outlook Express has a bigger footprint on the network. As we have mentioned earlier, it is difficult to make a direct comparison, due to the differences in the way each client application operates. However, for an individual operation, there is a substantial increase in network utilization when you move from Outlook 2000 to Outlook Express.

One last note about network utilization: This is only one decision point among many to consider. You’ll also want to keep in mind things like the client hardware and software environment, server environment, network architecture, bandwidth availability, firewalls, and differences in application features and functions.
Summary

Outlook Express can be used to connect to an Exchange server or with public e-mail, directory and newsgroup providers. It can also be used very effectively with Exchange 2000, but with some penalties in application flexibility and features over the Outlook 2000 client. In this chapter, you were exposed to Outlook Express components and features, installation and setup of basic services, configuring and using signatures, identities and digital IDs, remote configuration, and authentication and bandwidth characteristics.
Using POP3 and IMAP4 with Exchange 2000

The Exchange 2000 Information Store supports access from most clients that support POP3 and IMAP4. You’ve already seen how Microsoft clients like Outlook 2000 and Outlook Express access Exchange and other information stores using these protocols. We have also previously covered the basics of POP3 and IMAP4 protocols. But Exchange 2000 can be accessed (if configured properly) by any SMTP, POP3, or IMAP4 client compliant with the same versions of those protocols that are supported by Exchange. In other words, this chapter will cover some of the other popular POP3 and IMAP4 client applications, how they interact with the Exchange Information Store, and how to troubleshoot common problems.

The POP3 and IMAP4 Connectivity Model

A POP3 client connects to the server and authenticates and downloads messages for offline use. While connected to the server, the client application can make inquiries about messages in their mailbox, retrieve certain messages, leave copies of messages on the server, and mark messages for deletion. There is no direct manipulation of the remote folder or its contents.
When the client application connects to a POP3 store, it greets the server and authenticates if necessary. Once authenticated, the application queries the server about messages in the mailbox, downloads the messages, and marks the messages for deletion on the server—or leaves copies of them on the server if told to do so. The server deletes messages from the mailbox after the client issues the `QUIT` command to finish the session.

Some POP3 clients (like Outlook Express) can be configured to store the messages on the server for a period of time before deletion as well, but most users simply download all the messages and mark all downloaded for deletion. In both cases, all the new messages are downloaded. In one case a copy is maintained on the server after download of the original.

When the client sends a message, it connects to a server running the SMTP service. It issues a greeting to the SMTP server and authenticates if necessary. If the server responds positively, the client identifies itself and the recipient by e-mail address and then sends the message data. Once in session, the client continues to send messages until it has finished, and then issues a `QUIT` command. At this point, it is the responsibility of the SMTP server to route the message to the correct recipient. The SMTP server can deliver the message locally, or connect to another SMTP server that can route the message to the recipient. The basic POP3 send and receive process is shown in Figure 14-1.

IMAP4 is designed for online, as well as offline, mode. IMAP4 can do everything that we have described already, but it can also manipulate the mailbox and its folders on the server as if they were local. This is very different from the severe limitations users experience when manipulating data in a POP3 single-folder mailbox. IMAP4 allows users to create new folders, synchronize a local store with the remote store, connect to shared folders, and download parts of a message instead of the entire message, and only selected messages instead of all new messages, etc. IMAP4 is usually a better protocol to use if the server and the firewalls between you and your server support it and if you want the additional features. Keep in mind, though: Sometimes the additional features are not sufficient reason to move to IMAP4 because of the additional size of the IMAP4 clients and the overall decrease in speed of access.
Figure 14-1: Sending and receiving mail using a POP3 client application

All IMAP4 and POP3 clients are not the same, so if you are having speed or size issues, look at some of the alternative clients.

An IMAP4 client uses SMTP in the same manner as the POP3 client. Anytime messages are sent, they are submitted to an SMTP server for routing. Although a POP3 or IMAP4 client’s SMTP server acts as a relay, it should be noted that many SMTP servers are specifically configured to disable e-mail relaying. Figure 14-2 shows how IMAP4 works.
Figure 14-2: Overview of typical IMAP4 connections

**POP3, IMAP4, and Exchange 2000**

IMAP4 and POP3 were incorporated into Exchange as early as Exchange 5.0. They were the first Internet-based connectivity options for client applications. Support for POP3 and IMAP4 continue in Exchange 2000, providing cross-platform Information Store access for most popular Internet mail client applications.

**How it works in Exchange**

When either protocol is enabled on an Exchange 2000 server, the server begins listening for client connections on the specified port. Both protocols are enabled by default.
You can change the standard port numbers on the server, but you should be aware that every time users set up an application to connect to your server, they will have to change the connection port number as well. This may become a serious support issue, since most client applications assume use of the standard port numbers.

The standard port numbers for POP3 are 110 for an unsecured connection and 995 for an SSL-encrypted connection. The standard port numbers for IMAP4 are 143 for an unsecured connection and 993 for an SSL-encrypted connection.

When a POP3 or IMAP4 client makes a connection, the Exchange server accepts the command as if it were a POP3 or IMAP4 server. The POP3 and IMAP4 interfaces on Exchange 2000 run as processes of the Information Store. The client application has no indication that it is connected to an Exchange server other than the initial greeting it gets from the server.

Popular POP3 and IMAP4 clients

Not all Internet e-mail clients are created equal. The POP3 and IMAP4 specifications are broad and open, so there is an incredible variety of applications available. POP3 and IMAP4 applications are available for many hardware platforms, including Windows, UNIX (many varieties), MAC, and PIMs, such as the Palm Pilot devices. Since there are so many freely available clients, you have a great amount of flexibility in choosing which application to use when connecting to your Exchange 2000 mailbox.

**Tip**

Make sure you thoroughly test any POP3 or IMAP4 applications that will be used in conjunction with Exchange 2000 in a pilot before you commit to production. Things you’ll want to test include basic stability, connectivity, mailbox access, authentication and encryption methods, attachment support, and signature support. Make sure the client has a rich address book.

**Eudora**

Eudora is a popular Internet mail client for POP3 and IMAP4. It has a rich feature set, including signatures, a robust address book, integration with LDAP, and excellent message filtering. The Eudora interface is user-friendly and, like most Internet mail applications, the program can be downloaded from the Internet at no cost. You can find Eudora at [www.qualcomm.com](http://www.qualcomm.com).

The Eudora installation is wizard-based. During the install, there are very few decisions to make — except which directory to install the program. Once you’ve installed the product, double-click on the shortcut, and you’ll be prompted with a number of configuration questions. After a brief introduction, you’ll be asked whether you want to create a new account or import settings from an existing account. Eudora will import settings from other Windows mail programs, such as Netscape Communicator or Outlook Express. If you are creating a new account, you’ll be prompted for information to configure the application to connect to your Exchange 2000 mailbox. The information you’ll have to provide is the same as every POP3 or IMAP4 application. Table 14-1 shows the information you’ll need to provide.
Table 14-1
Required Configuration for Connecting Eudora 4.3.2 to Exchange 2000

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real name</td>
<td>This is your friendly e-mail name. When you send a message, the recipient sees this sender name.</td>
<td>This name can be anything you like, but it’s helpful to use the format Last, First so it is easily referenced in other users’ address books.</td>
</tr>
<tr>
<td>Return address</td>
<td>This is the e-mail address of your mailbox. When someone replies to your message, the reply will be sent to this e-mail address.</td>
<td>This address can be any of the valid SMTP addresses assigned to the Exchange 2000 account. SMTP addresses are listed in the user properties, E-mail addresses tab in the Active directory.</td>
</tr>
<tr>
<td>Mail server</td>
<td>This is the Exchange 2000 server that contains the Mailbox store for this user.</td>
<td>This information is stored in the Active directory, in the Exchange general tab of the user properties.</td>
</tr>
<tr>
<td>SMTP server</td>
<td>This SMTP server relays outgoing messages for the POP3 or IMAP4 client application.</td>
<td>Any SMTP server accessible to the client that relays mail from external sources will do. It can be the Exchange mailbox server or another SMTP server.</td>
</tr>
<tr>
<td>Login name</td>
<td>In the case of Exchange 2000, this is the user’s Windows 2000 Active directory user login name.</td>
<td>If you have recently created the Windows 2000 account, the client application may have trouble connecting if you force them to change their password the first time they log in. You may have to clear this option in order to have success.</td>
</tr>
</tbody>
</table>

Once you’ve finished the wizard, you should be ready to begin using Eudora to access your mailbox on the Exchange server. The interface is friendly and easy to use. Therefore, you should have no problems working with Eudora right away. Most of the message functions are located in the Message menu, and configuration options are located in the Tools menu. Figure 14-3 shows the interface for the sponsored version of Eudora.
Outlook 2000

Outlook 2000 with the Internet Mail information service is a Microsoft-provided client that works well with POP3 and IMAP4. It can be used in Corporate or Workgroup mode, or in Internet-Only mode.

To see if you are running in Corporate or Workgroup mode, open Outlook. Once in the application, go to Tools/Options and switch to the Mail Services tab or Mail Delivery tab. Click on Reconfigure Mail Support to see which mode you are running in. If you switch from Corporate or Workgroup mode to Internet-Only mode, you will lose any functionality that is provided by the Exchange server Information service, which includes Exchange address book browsing, server-based rules, the out-of-office assistant, and more.

Outlook 2000 works particularly well if your company runs Microsoft Exchange and you want to integrate your personal and business e-mail into one application. Running in Corporate or Workgroup mode allows you to use the Exchange server Information service in conjunction with the Internet mail Information service. This means you can connect to your Exchange mailbox and ISP-provided e-mail at the same time, in the same application.

From an e-mail perspective, running Outlook 2000 in Internet-Only mode is similar to running Outlook Express. The interface is slightly different, and you can use additional data types such as tasks, notes, and journal.
Outlook Express

Outlook Express is an excellent choice for users running only POP3 or IMAP4 mail accounts. See Chapter 13 for more detail on this client application.

Pine

Pine was originally developed at the University of Washington as an easy to use e-mail client; its functionality was limited on purpose for the sake of portability and usability. Though the core of Pine remains simple, the application has been extended over the years to include many powerful features to suit advanced users. Pine is a good e-mail application for users running Unix (as well as versions of Linux) and is available in a Windows version (PC-Pine). This is open source, freely distributed software that is well-documented and stable.

Though designed for novice users, Pine can also be customized to provide advanced features for more powerful users. Pine will connect to an Exchange 2000 mailbox account using POP3 or IMAP4, or NNTP, and provides an easy to use application that can be extended. Pine has a message index to show a summary of messages with sender names, message subject, size, etc. The basic command set is very simple, as you can see in Figure 14-4.

![Figure 14-4: Main menu with the Pine command list](image-url)

Beyond basic e-mail handling, the application performs spell checking, message formatting, and personal signatures. The address book allows you to create and save personal addresses and distribution lists, which can be associated with friendly nicknames. Pine supports MIME attachments for processing documents, spreadsheets, graphics, and sound files. Pine is a popular Unix e-mail client. It is used in many educational institutions and in other organizations that have Unix mail servers or Unix clients.
Pine can be obtained on the Internet, but the download location will vary according to what version you need, which operating system you are using, and whether you want a compiled version or something precompiled. Some versions of Unix, such as Red Hat Linux 6.x, include Pine in their package. Instructions for compiling and installing the product are available for download with the application. The basic configuration information is stored in one or more files on your local machine. Sometimes there will be a file containing system defaults, but there will always be a personal configuration file (usually called pinerc). Be careful to pay attention to the application documentation, especially if you are using extensions to Pine. You'll want to make sure that the type and version of operating system, e-mail application and extensions, and the Exchange 2000 server are compatible.

**Netscape Messenger**

Netscape Messenger is a good e-mail application for many popular hardware types, such as Unix, Mac, and Win32. It's freely available and can be downloaded from [www.netscape.com](http://www.netscape.com). Messenger is packaged with the suite of Internet access products called Netscape Communicator. Messenger is a sophisticated application with a rich feature set; it offers great performance and flexibility.

Once Messenger is installed, you have to provide the same basic information as other POP3 or IMAP4 e-mail clients in order to connect to your Exchange 2000 server. When you start the application, you'll be prompted to create a profile that contains all of your personal settings, including e-mail. The profile creation wizard will assist you in configuring all of the information necessary to connect to your Exchange 2000 mailbox.

When you open the application, simply go to the Communicator menu and choose Messenger. This will bring you into the Messenger application, where you can access your mailbox and start working.

**Troubleshooting**

When you are troubleshooting Internet client connectivity, there are several steps to take before you go any further. First, check the client settings. Next, find out if there is a name resolution or connectivity problem at the client. If everything seems in order, turn on protocol logging on the server or on the client to get additional information about the actual problem. With this information, you will have far more knowledge about what problem to fix. If it turns out that the problem is one that can be remedied with client configuration, you may have to alter one of the client settings.
Checking client configuration settings

The procedure for checking the client application settings will depend on the client application and operating system you are using. The focus of your troubleshooting effort will depend on the type of problem you are experiencing. However, the basic information you need to verify is the same no matter what system or application you are using. You’ll want to check the following information to make sure it’s correct:

✦ Incoming mail server name (POP3 or IMAP server) and server TCP port used
✦ Outgoing mail server name (SMTP server) and server TCP port used
✦ Authentication type, user name, and password
✦ Mailbox name or e-mail address

The application configuration information is normally accessed through the application itself. With the applications we have mentioned, you can check configuration settings in the locations listed in Table 14-2.

<table>
<thead>
<tr>
<th>E-mail Application</th>
<th>Location of Configuration Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eudora</td>
<td>Go to the Tools/Options menu and look at the Getting Started section.</td>
</tr>
<tr>
<td>Netscape Messenger</td>
<td>Go to the Edit/Preferences menu, look under the “Mail and Newsgroups” heading, and check the settings in the “Identity” and “Mail Servers” sections.</td>
</tr>
<tr>
<td>Outlook 2000</td>
<td>If Outlook is configured for Internet Only, go to the Tools/Accounts menu, switch to the Mail tab, and double-click on the account you are troubleshooting. If Outlook is configured for Corporate or Workgroup, go to the Tools/Services menu and double-click on the Internet Mail service, or highlight it and choose Properties.</td>
</tr>
<tr>
<td>Outlook Express</td>
<td>Go to the Tools/Accounts menu, switch to the Mail tab, and double-click on the account you are troubleshooting.</td>
</tr>
<tr>
<td>Pine</td>
<td>From the main menu, choose “S” to go into setup and configuration.</td>
</tr>
</tbody>
</table>

Checking connectivity and name resolution

Checking your connectivity to a server is getting more difficult as the Internet protocols and IP security practices mature. One problem is that many more types of IP
traffic (such as the Internet Control Message Protocol (ICMP)) are being commonly filtered in an effort to close off potential access points for a malicious entity. Another problem is that Microsoft has changed its IP implementation so that Telnet is no longer integrated with the protocol. Telnet is now a separate service that does not run by default.

The result of these two factors is that you can no longer be sure that Ping (which utilizes ICMP) or Telnet will tell you anything useful in a connectivity troubleshooting effort. Our suggestion is that you try to ping the server names (POP3/IMAP4/SMTP) by name to see if you can eliminate two of the basic issues you could be having. Figure 14-5 shows a Windows-based client, using ping to see if their IMAP4 server is available on the network.

![Figure 14-5: Using ping to check name resolution and possibly network connectivity](image)

If you ping the incoming and outgoing mail servers by name and get a response, you can be reasonably sure that the server is running, that you have good IP name resolution, and that you have network connectivity to the appropriate servers. If you do not receive a response, it may mean that the server is not available or that pings are being filtered out at a router.

If you try to ping the server by name, and you get a “Bad IP address or Unknown Host” response, you probably have a name resolution problem. At this point, you should check your DNS settings to verify they are correct. Also, see if you can ping the DNS server. If it is not available, you will not be able to resolve the IP address of your incoming and outgoing mail server names. If you have a local host file, make sure that any information that pertains to your incoming and outgoing mail server is correct, and that the IP address of these servers has not changed.
Protocol logging

You can use diagnostic logging to record significant events related to authentication, connections, and client actions. Diagnostic logging for both POP3 and IMAP4 is written to the Windows 2000 event log.

Logging must be configured on the server that hosts the virtual server. If multiple POP3 virtual servers are added to one Exchange computer, all POP3 virtual servers are logged at the level set on the host server. By default, a logging level of None is selected to log only critical errors.

To set the logging level, start system manager on the Start menu, point to Programs, point to Microsoft Exchange, and then click System Manager.

Drill down into the system manager and highlight the server that hosts the POP3 or IMAP server you are troubleshooting. Right-click on the server and go to Properties.

Next, switch to the Diagnostics Logging tab. In the Services box, you'll want to highlight POP3 or IMAP4 and set logging level options in the Categories pane. As shown in Figure 14-6, you can set options for General events, connections, authentication, etc.

Setting the logging level to medium will usually provide enough information to successfully identify a problem. A medium setting generally logs the individual tasks in a process without recording the results of each line of code being processed.
Setting logging on the client is usually an option as well, and may provide different indicators than server protocol logging. For example, in Outlook Express you can go to Tools/Options, switch to the Maintenance tab, and turn on logging for any of the supported Internet protocols. In other applications, logging can be set in an “.ini” file, or as a command-line option. Protocol logs can get quite large. Therefore, make certain you disable the function and delete the logs when you are done troubleshooting.

Authentication, Security, and Bandwidth Profile

Many users, and even some administrators, do not understand the differences between POP3, IMAP4, and other clients (like the Outlook 2000 client) with regard to authentication, security, and bandwidth. Yet in most large, and even some small, environments, you cannot avoid considering their impact on your system. In essence, some of the clients default with unsecure authentication methods, making it easy for anyone with a network protocol analyzer to “steal” passwords. To be more specific, plucking a “plain text” password is a bit like starting a car that has been left unlocked with the keys still in it located in the seedy section of town. It is almost too easy. To understand the differences between these clients, not only do you have to address the security of authentication, but how often and when the client authenticates. Frequent authentication puts additional packets on the network, driving up bandwidth requirements and putting additional load on the server(s) involved.

Authentication methods

Basic, or “clear text,” is a simple process of passing an unencrypted user name and password to the server. In this scenario, it is easy to discover someone’s password by simply looking at a network trace of the session. In the data portion of the authentication packet, you will see something that looks like:

```
USER moften ...PASS 99Pty
```

Needless to say, if the session is being traced or captured by a protocol analyzer, your password is soon discovered.
When an application supports Secure Password Authentication (SPA), it should be used. In Exchange 2000, SPA uses the Windows NT Challenge/Response (NTLM) authentication method, which is much more secure because it uses a randomization algorithm and an encrypted password to authenticate users. The following process describes the NTLM authentication method:

1. The client sends an AUTH NTLM command to the server on TCP/IP Port 110.
2. The server responds with a +(space) response to indicate the port will accept commands.
3. A Negotiate message is sent to the server with some random information.
4. The server responds by creating a unique challenge with the random information.
5. The client encrypts the challenge with its password and sends the response to the server.
6. The server then looks up the user’s password and encrypts the challenge sent to the client.
7. If the encrypted data matches the client response, then a Successful Acknowledge is sent to the client. If a Successful Acknowledge is not sent, an Access Denied is returned again to the client.

Data integrity and security

Even if you have secured your authentication credentials, you still have not secured the information in your message. The way to ensure that the data in your message is not being transmitted in clear text is by using some form of encryption.

Types of session encryption

Secure Sockets Layer (SSL) is one method that can be employed to encrypt a session. It has to be set up on the server and supported and enabled on the client. SSL protects the session between the client and the server. It uses public/private key technology to ensure communication privacy and data integrity. Here are the basics of how SSL works:

1. **Client Obtains Server Certificate**: The client and server introduce themselves to each other and the client obtains the server’s certificate (with its public key) from the server.

2. **Client Verifies Server**: The client goes to the server’s certificate authority and checks to make sure that the certificate is valid. Once the certificate is validated, the client sends an encrypted authentication challenge to the server using its public key. If the server can authenticate (by using its private key) it is assumed trusted.
3. **Client/Server Determines Encryption Key to Use for This Session**: The client and server now agree upon a secret key that will be used to encrypt all session data. Since only the client and server know the secret key, only they will be able to read.

Virtual Private Network (VPN) is another way to encrypt an authentication and data session. The session is normally encrypted between the client and the external interface of a VPN router. Session data on the internal network is not normally encrypted, so the client/server session is not entirely secure.

Some VPN solutions have difficulty passing traffic through one or more proxy or other translation-type network devices (NAT/PAT). To bypass proxy devices you could possibly use an RRAS solution to connect to an internal network routing and authentication device. RRAS allows the authentication and data traffic to be encrypted, but it requires that you allow a direct connection to your internal network via ISDN, analog modem etc. VPN and RRAS can be used together with the RRAS server acting as a VPN authentication and network routing device.

**Signing a message**

Some POP3 and IMAP applications can digitally sign a message. To digitally sign a message, certificate, or digital ID, a plug-in application has to be installed on the client. Client e-mail certificates can be obtained from a variety of sources, including Verisign, Inc.

When you send a signed message, the recipient verifies the integrity of the message contents by using the public key included with the signed message. Once this recipient has your public key, they can also send you encrypted messages that only you can read.

**Encrypting a message**

Some applications can encrypt a message before it is sent, by using a digital signature, certificate, or encryption plug-in. The sender encrypts the message by using the recipient’s public key. The recipient opens the encrypted message by using their private key.

In order to encrypt a message, you must have the public key of the recipient. The windows address book automatically detects a digital ID (including the user’s public key) that another user sends you and will add it to the address book automatically. If you do not have their ID, you need to have them send it to you. If they refuse to send it to you, or you do not yet have it, you will be prohibited from sending to their address by most clients like Outlook.

Applications vary greatly in their support for digital signing and encryption. If these are features that will be used, documentation and training for end-users on how to use them, as well as careful testing for interoperability, is strongly recommended.
Bandwidth profile and considerations

From a bandwidth perspective, it is difficult to profile POP3 and IMAP4. POP3 and IMAP4 network profiles can vary greatly depending on how the protocol is implemented on the client and how it is implemented on the server.

IMAP4 will probably work better for those who travel, since it gives you the option of delaying large message retrieval until you have a robust network connection. Since both of these protocols use TCP, they are tolerant of low-bandwidth environments.

In comparison with other types of connections to Exchange, please see the Bandwidth Profile section in Chapter 13.

Resources

There is plenty of support for the POP and IMAP protocols on the Internet. There are Web sites such as www.imap.org that deal specifically with the protocols and their implementation. As we have mentioned previously, not all application developers choose to implement the protocol the same way. Application support should come from the application developer. For example, Pine information is available at www.washington.edu/pine.

Summary

In this chapter, we presented various aspects of using applications based on the POP3 and IMAP4 support built into Exchange 2000. We walked you through some of the more popular client applications and how they are configured to connect to Exchange. We also showed you some ways to troubleshoot configuration and connection problems. We warned you about security exposures with both protocols. Finally, you learned about some common e-mail data and network session security techniques.
Outlook Web Access

As the impact of the Internet swept the world, users and Administrators wanted an easy way to get mail from their browsers. Outlook Web Access (OWA) was created to satisfy that need. Since its inception in Exchange server version 5.0, OWA has gone through many enhancements with each subsequent major release and Service Pack. Now, with Exchange 2000 server, the feature has been completely overhauled, and enhancements to the Exchange 2000 architecture will dramatically affect the way Outlook Web Access works. With these enhancements, Outlook Web Access offers significantly increased scalability and functionality.

Old and New Versions of Outlook Web Access

OWA in Exchange 2000 is a vast improvement over the version introduced with Exchange server 5.0. To satisfy a browser client request, Outlook Web Access 5.x used Active Server Pages (ASP). The ASPs communicated with the Exchange server, which was using Collaboration Data Objects (CDO) 1.2 and Messaging Application Programming Interface (MAPI). The number of users that a server could effectively support was limited by the immense overhead required by the interpreted scripts in ASP and the additional overhead to run MAPI sessions within ASP. In a sense, it seemed that Outlook Web Access was a part of Microsoft Internet Information Services (IIS) and not quite a full-fledged part of Exchange. For example, a company we were quite familiar with found that a powerful server box running Exchange 5.x, which could support 1000 users running Outlook clients connected to it, could support only 300 OWA clients successfully.
OWA was the first production use of Active Server Pages from any major manufacturer. It was a pioneering product and had its limitations (as version 1 of any program does).

The server side of OWA in Exchange 2000 no longer uses MAPI or ASPs; OWA is now built into the Microsoft Web Storage System and uses IIS only to receive requests and pass them to the Web Storage System.

One implication of this change occurs if you use Exchange 2000 to create multiple root level stores to replace, or coexist with, the All Public Folders root level store. Your OWA client will be able to access them if it has the Installable File System (IFS) loaded. By contrast, MAPI clients such as Outlook are able to see only the initial (All Public Folders) tree.

Client access continues to use HTTP, making OWA accessible to Unix and Macintosh clients as well as any Wintel clients with a compliant browser.

When OWA is used with Internet Explorer 5.x, it is more efficient (less bandwidth is required). Other browsers communicate to the server after every click, but Internet Explorer 5.x and OWA are built to eliminate that excess communication when used together.

Internet Information Server Version 5.0 is now installed as part of Microsoft Windows 2000 and is required by Exchange 2000. Because IIS is now part of the operating system and the protocols are now basically transport stacks, IIS can run the transport protocols independently. IIS remains the server recipient for all incoming HTTP requests from all types of Web browsers. It remains responsible for sending HTTP responses from both Exchange 2000 server and OWA. If the information requested resides in a local Exchange 2000 database, OWA will satisfy a browser user request by taking advantage of high-speed channels to access the local mailbox store (in earlier versions of Exchange, this was not possible). If the server is only a front-end server and has no local Exchange data store, OWA will still handle the request but will refer it to one of your back-end servers using HTTP. This process allows a topology that really scales. You can design a topology with front-end servers that handle all incoming HTTP requests. Those servers feed all requests to back-end servers where all the data resides.

**Features and browser selection**

The new features in OWA bring it closer to the features in Outlook 2000. An example is shown in Figure 15-1, where an OWA client using Internet Explorer 5.x accesses My Calendar on the Exchange server. In addition, Exchange 2000 and OWA now can support certain clients with shortcuts for menus, editing drag and drop, rich text editing, native Kerberos authentication, and open/create new folders tree control.
Client-side features may need to be supported in the client-side browser. Some features require Microsoft Internet Explorer 5.x or a browser that has the same capabilities. Microsoft’s implementation of Kerberos authentication also requires Windows 2000.

Microsoft would like your browser of choice to be Internet Explorer 5.x. To help you select Internet Explorer version 5.x, they have enhanced the way it works with OWA. Internet Explorer 5.x uses Dynamic HTML (DHTML), Extensible Markup Language (XML), and an HTTP protocol extension called WWW Distributed Authoring and Versioning (WebDAV) to provide viewing more similar to a standard Outlook client than anything you have seen before in OWA. Older browsers that do not support DHTML and XML may deliver only basic functionality with OWA, since providing backward capability sacrifices some of these newer features. For basic functionality, the Exchange 2000 version of OWA supports most of the recent Internet browsers (version 4.0 or above is recommended for IE and Netscape Communicator). Your browser must include, at a minimum, HTML version 3.2 and JavaScript functionality. Before you go into production, test the browser to ensure the features you need are working properly.

The new OWA features in Exchange 2000 are as follows:

- With IE Version 5.0 and above, OWA is closer to the full Outlook client interface and more efficient for users; it includes text composition, drag-and-drop editing, preview pane, and tree control.
All of these features are Internet Explorer DHTML/XML-dependent, and will not work with earlier IE versions or most other browsers.

✦ Public folders can contain contact and calendar items.
✦ Embedded items such as messages, appointments, and meeting requests, as well as contacts and posts, are supported.
✦ Named URLs (plain-text addresses) that reference items are supported. In previous versions, Globally Unique Identifiers (GUIDs) were used to reference items in the Information Store. Now, object items (messages, folders, and so on) are accessed by using the familiar URL format. For example: http://servername.domainname.com/exchange/userID/All Public Folders, http://servername.domainname.com/exchange/userID/inbox, and so on.
✦ Using named URLs is a primary access method. Explicit URL addressing opens any item and performs many functions. Even if you want to access your own Intranet, you can use the same familiar URL method; for example, http://servername/exchange/userID. To access a server, you would use the fully qualified domain name (FQDN): http://servername.domain.com/exchange/userID.
✦ Command verbs and other options can also be used with a URL: http://servername.domainname.com/exchange/usermailboxID/inbox/?Cmd=new (the URL for opening a new message for editing).

For more information on commands, especially those that are specific to Microsoft extensions, see the Exchange 2000 Server Software Developer’s Kit.

✦ Multimedia messages, including audio and video clips, can be added to a message.

Limitations of Outlook Web Access compared to Outlook 2000

It is common to find multiple messaging client applications in organizations of any size. Outlook Web Access is not a wholesale replacement for the full-featured Outlook 2000 messaging client, although it can be an important addition to your mix of clients. OWA in Exchange 2000 with IE5 does not support the following Outlook features:

✦ Offline use and Offline Address Book
✦ Tasks and journal
✦ Templates for printing
✦ Outlook rules (it supports Out-of-Office Assistant, but not the others)
Outlook Web Access

- Copying between mailbox and All Public or other public folders
- Telephony options and user-defined fields when accessing contacts
- Spell-checking mail features, “Do not deliver before,” and expiration options
- Typing and editing directly into the Calendar view and reminders
- Complete set of views and customization of views

Likely Scenarios for Outlook Web Access

Though Outlook Web Access may not have all the features of Outlook 2000 and is not for everyone, it does offer a significant amount of functionality — especially when used with Internet Explorer 5.x. That functionality becomes especially important when an organization attempts to provide messaging to all its constituents in many different business scenarios — including a “light” messaging client, roving, and kiosks.

“Light” messaging and collaboration client

If you are concerned about memory, disk space or budget, OWA provides an acceptable, not quite so full-featured, alternative to the full Outlook 2000 client. OWA is also attractive if you wish to reduce the cost of managing desktop applications. Using the Web browser for messaging means that you have one less application to support on the desktop, and it also means that you have a subset of features to support. Although some features are missing, the addition of access to folders, rich text, and so on makes it possible to do a fair amount of what users normally do. Sophisticated users of Outlook may feel limited by OWA, but users who do not use the more advanced features of Outlook may find the trade-off reasonable. Even if users really want a full-featured client, it is simply not practical sometimes — for example, if you want to use Virtual Private Network (VPN) for messaging access through the Internet, or if you are on the wrong side of a firewall and ports for POP3, IMPA, or Outlook are not open. Not every user can get the connectivity to use a full-featured client like Outlook 2000. OWA, which is HTTP-based, can also solve bandwidth or latency problems.

Note

Few organizations block HTTP at their firewalls, so access to the WWW on the Internet is not blocked.

Knowing the benefits and limitations of the OWA client can help you determine which users have limited connectivity, or are otherwise appropriate candidates for a browser-based, light messaging client connected to OWA.

Cross-Reference

Outlook 2000 has many benefits not covered in this chapter, whose focus is on OWA. For thorough coverage of Outlook 2000 and its many features, see Chapter 12.
Supporting roving users

Many find it challenging to support users who move from computer to computer. When users move, important information about them may not move with them. The common way to solve this problem is by using system policies and server-based profiles. But this elegant solution can also require time-consuming administration, as well as server and network-performance overhead. OWA, on the other hand, can solve the problem of roving user support with little administrative effort, because nothing (not mail, not MAPI profiles, not flags — simply nothing) is stored on the client. Further, almost any client with a fairly recent browser can provide the basic functionality.

One way to get client management and administration is to use Microsoft IntelliMirror with full-featured clients like Outlook 2000. But if you cannot use IntelliMirror, you could just implement OWA for a quick, simple, low-cost solution. The trade-offs include functionality, features, administrative control, security, and elegance.

Kiosks and public area access

Most of us have seen kiosks or computers in public areas — school common areas, libraries, factory floors, conference rooms, lobbies, and so on — to provide e-mail capability in strategic locations. OWA can provide users with e-mail, calendaring, public folders, and other basic messaging functions while controlling costs.

Migration and coexistence strategies

During many migrations and in some cases of coexistence, providing continuous e-mail functionality can be difficult — for example, when client desktop software is being altered. With OWA, no mail is downloaded, and almost any client can access the server. Machines can easily be shared.

If users are migrating to Outlook 2000 but are not familiar with it, using OWA and Internet Explorer 5.x can help acclimate them, and client configuration is minimal.

Planning, Installation, and Administration

Planning, installing, configuring, and administering OWA is easy in Exchange 2000. The complicated part is deciding between a single-server or multiple-server deployment, capacity planning, dealing with upgrade issues, and choosing a type of security (client authentication method).

Upgrading from 5.5 and Outlook Web Access

If you are currently running OWA on a server that is running Exchange 5.5, upgrading to Exchange 2000 automatically replaces your existing configuration.
If you have customized the ASP files in OWA, you may lose some of the customization in the upgrade. Customized ASP files remain in the Exchsrvr\Webdata directory, but do not function.

To preserve your customization of OWA, you can continue to run Exchange 5.5 OWA with Exchange 2000 back-end servers. But if you do, you receive none of the newer advantages, such as performance and scalability improvements in Exchange 2000 Server and other new features. Using earlier versions of OWA may make sense as a temporary measure, however, especially if you cannot update the customization before you migrate the users. For example, if you are upgrading an Exchange back-end server, you might want to continue using Exchange 5.5 and OWA.

Scrutinize any proposal to continue using old versions of OWA. It is rare that you cannot find a better solution by upgrading to the Exchange 2000 version immediately.

**Upgrading**

When you upgrade your OWA front-end and back-end architecture to Exchange 2000, you probably want to avoid disrupting service, and you may want to avoid changing certain users’ URL addresses for a time. To accomplish this goal, you can deploy Exchange 2000 in stages.

You cannot perform an in-place upgrade for an OWA front-end server without first upgrading the corresponding back-end server.

One way to address legacy ASPs and preserve some URLs for a period is to only upgrade some of your front-end and back-end servers to Exchange 2000 Server at first. Early adopters in your organization (users who don’t mind learning a new URL for OWA) can connect to an Exchange 2000 front-end server to reach their mailboxes that are on an upgraded back-end server. They will begin to use the new features and create positive feedback for others encouraging them to make the change. Meanwhile, remaining users can continue using Exchange 5.5 OWA with earlier versions of Exchange running on the older back-end servers. Their URL does not change, and ASP customization is preserved until these servers are upgraded. This method provides time for users to gain confidence and for the development team to replace the ASPs.

**Single-server environment**

When you use OWA in a single-server environment, clients connect directly to the Exchange server that stores their mailboxes. An Exchange virtual root and a public virtual root are added to Internet Information Services (IIS). These virtual roots point to their corresponding directories in Exchange.
**Multiple-server environment**

When you use OWA in a multiple-server environment, consider the following when planning your topology:

✦ Use front-end and back-end architecture for your Exchange servers.

✦ Use load-balancing software or hardware to efficiently respond to user requests and improve server availability.

In a multiple-server topology, the front-end server receives all the original connections and sends appropriate HTTP requests to a back-end server running OWA. To configure the front-end/back-end architecture for OWA, you must create an HTTP virtual server under the HTTP protocol in System Manager. You will find that the default virtual mail directory is Exchange. Select a hostname for the virtual server, bearing in mind that it determines the URL that users will type to access their mailboxes. The combination of hostname, TCP port, and IP address uniquely identifies a virtual server.

Tip

Choose a name for the virtual server that is consistent with your corporate naming policies and standards.

If your organization uses Secure Socket Layer (SSL) encryption, there is an additional advantage to front-end/back-end architecture. SSL takes processing power to encrypt and decrypt; the front-end/back-end architecture can improve performance by distributing this load off the server providing OWA at the back end.

**Using load balancing**

By allowing multiple servers to handle requests addressed to a single IP address or name, load balancing scales your Exchange 2000 messaging environment for OWA clients. Load balancing has the following advantages:

✦ Users need only one URL to access their mailboxes. The load-balancing software or hardware determines which server handles user requests.

✦ Users are directed to another server if a server running OWA stops responding.

✦ Load-balancing software or hardware efficiently distributes users across multiple servers, so users do not overload a single server.

Note

If you decide to use load balancing, the server must have round-robin DNS.

If you have round-robin DNS, map your hostname in DNS to several front-end servers. That way, if one of those servers stops responding, incoming client requests go to another computer capable of responding. Even though new clients are directed to an available server, clients connected to a failed server will need to re-authenticate and start a new session.
For more reliable load balancing, many organizations use add-on hardware or software. A number of third-party vendors offer load-balancing hardware; Microsoft provides Network Load Balancing, which comes with Windows 2000.

**Network security**

Microsoft provides a number of options for OWA authentication; each comes with attributes and costs. Choosing the appropriate option(s) for your organization usually depends on the capabilities of the client operating system, the robustness of your servers, and your specific security policies.

The following are the available options for authentication:

- **Basic**: Uses clear text to perform a simple challenge and response
- **Integrated Windows**: Leverages the native security attributes of the Windows client
- **Anonymous**: Provides access to public folders that are intended for general access (remember to monitor vigilantly, lest you provide anonymous access to the wrong folders)
- **Secure Socket Layer (SSL)**: Although not an authentication method, provides a secure communications channel that can be used in combination with any of the above methods (while providing a high level of security, this setting also requires more work for your servers doing the encryption and decryption)

<table>
<thead>
<tr>
<th>Caution</th>
</tr>
</thead>
</table>

Piping clear text across the Internet leaves your organization with no security. This is almost always a bad idea.

In a single-server OWA environment, the default authentication methods are Basic and Integrated Windows authentication. The authentication choice is set on the HTTP virtual servers configured for OWA.

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
</table>

OWA does not provide the user a button for logging off; to log off deliberately, the user must close the browser. Timeouts and other planned or unplanned interference may cause the session to close with no user intervention.

**Basic authentication**

Basic authentication typically appears on company intranets and on casually or erroneously installed and configured Internet clients.

<table>
<thead>
<tr>
<th>Caution</th>
</tr>
</thead>
</table>

If your intranet carries confidential data (human resources, accounting, and so on), do not use basic authentication.
With basic authentication, critical information is being transmitted in such a way that any thief with a bit of knowledge and a protocol analyzer can readily capture and display it. The alternative NTLM protocol accepts established users’ identification through the access token; basic authentication does not even provide that level of protection. It relies on users to enter their user name, domain, and password to authenticate to OWA. Basic authentication has advantages and disadvantages:

- **Advantages:** With browser independence, basic authentication becomes platform-independent. It is also easier to configure front-end/back-end server topologies than with more sophisticated authentication methods.

- **Disadvantages:** Unencrypted, plain-text, easily captured passwords are transmitted over the network; the only way to be less secure is to require no password, or common passwords (such as `password`). Users also find basic authentication inconvenient, because they must enter their user name, domain, and password each time they log on.

To gain a modicum of security with basic authentication, combine it with SSL.

### Integrated Windows authentication

Integrated Windows authentication requires you to have a Microsoft Windows client — optimally, Windows 2000 and Internet Explorer 5.x or higher. This configuration takes advantage of Microsoft’s implementation of Kerberos. If you do not have Windows 2000 but some other Microsoft Windows networking client, NTLM protocol rather than Kerberos is used for authentication.

The advantages of integrated Windows authentication are as follows:

- It encrypts the client’s password, providing significantly improved security for authentication credentials.

  Encrypting your authentication credentials does not mean that your message is also encrypted. Protection of your message and attachments is covered in Chapter 22.

- It can eliminate the repetitive logon required with basic authentication by affording native authentication from Windows networking clients. It allows browser access without prompting the user for user name and password.

- Windows 2000 clients running Internet Explorer 5.x can use Kerberos.

Integrated Windows authentication also has disadvantages:

- It requires a Windows-based client operating system, and will not work with browsers other than Internet Explorer 4.x and 5.x.

- It prohibits front-end/back-end server configuration (because of passing the encrypted password from one server to the other).
Anonymous access
Windows 2000 IIS enables the creation of a special user account that users can use to connect anonymously. This is a powerful feature common in many Internet scenarios, but potentially dangerous — someone can log on with no credentials, and you will have extremely limited information about who it is. When used properly, however, anonymous access can enable limited access for specific public folders and directory information. For instance, it can provide the utility to publish a public folder to the Internet, enabling random people to access it. Of course, you should not allow anonymous access to any information if you want to limit the access to known individuals and groups. Anonymous access has advantages and disadvantages:

- **Advantages:** All recent browsers support anonymous access, so it is an easy and fast way to provide insecure access to public folder data. Administration is made simple with a single point of configuration.
- **Disadvantages:** It is impossible to identify users with anonymous access, and therefore to track usage by user.

If the server providing access is inside a firewall, you may have to ensure that anonymous users external to the firewall can get through your firewall and to the Exchange server successfully. There are usually many other ways your WAN and LAN teams have purposely configured obstacles to anonymous access. If you choose to use anonymous access, topological placement of your server will be important.

Secure Socket Layer
Although Secure Socket Layer (SSL) is not an authentication mechanism itself, it is an important part of authentication. SSL provides whatever authentication method you choose with the highest level of encryption security: the entire communications session is encrypted. The most common implementation is basic authentication with SSL; other authentication methods can also be made to work with SSL, but they retain their native restrictions.

Advantages of using SSL are as follows:

- All of the communications session becomes encrypted.
- Most recent browsers support SSL communication.

Using SSL, however, also presents disadvantages:

- Because it causes processing overhead for the encryption and decryption, it may reduce the overall performance of the authenticating server.

If the server has more than enough power, processing speed is not a problem. But if the server is already taxed, SSL could affect other processes as it contends for resources.
When it is used with basic authentication, users must enter user name, domain, and password each time they log on. That repetition increases the OWA server’s SSL encryption/decryption load.

Its overhead can cause latency.

SSL was originally designed as a transport layer protocol to secure TCP/IP based protocols like IMAP, NNTP and is now frequently used with Transport Layer Security (TLS) with HTTP.

Setup and administration

To be installed, OWA requires Windows 2000 and IIS 5.0. If they are present, OWA will install automatically as a part of the default setup of Exchange 2000. Table 15-1 shows where and what is installed for OWA during Exchange 2000 setup.

### Table 15-1
**Exchange 2000 Directories Pertinent to OWA**

<table>
<thead>
<tr>
<th>Directory</th>
<th>Contains</th>
</tr>
</thead>
<tbody>
<tr>
<td>\exchsrvr\bin</td>
<td>Wmtemplates.dll, which defines the default templates used to render OWA</td>
</tr>
<tr>
<td>\exchsrvr\exchweb\bin</td>
<td>Exwform.dll, which handles form processing</td>
</tr>
<tr>
<td>\exchsrvr\exchweb\controls</td>
<td>Client Microsoft JScript and Internet behavior script, to facilitate caching by IIS the client (code is separated from the Outlook Web Access ISAPI application)</td>
</tr>
<tr>
<td>\exchsrvr\exchweb\lang</td>
<td>OWA Help files in their localized versions</td>
</tr>
<tr>
<td>\exchsrvr\exchweb\img</td>
<td>OWA Graphics</td>
</tr>
</tbody>
</table>

During setup, Exchange 2000 creates four virtual IIS directories that are used by Outlook Web Access, as shown in Table 15-2.

### Table 15-2
**Virtual Directories Used by OWA**

<table>
<thead>
<tr>
<th>Web Virtual Directory</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>/exchweb</td>
<td>Graphics and other ancillary files</td>
</tr>
<tr>
<td>/exadmin</td>
<td>Exchange Administration public folders administration tool</td>
</tr>
<tr>
<td>/exchange</td>
<td>Mailbox root</td>
</tr>
<tr>
<td>/public</td>
<td>Default public folders tree</td>
</tr>
</tbody>
</table>
Server configuration

Configuring OWA is straightforward: by default, it is configured to allow access to users’ mailboxes and the default public folder tree. If the clients your OWA is serving have Internet Explorer 5.x, you may want to configure the server to provide customized access that goes beyond the default configuration. For these clients, you can specify which users are allowed to access the server from a Web browser, the allowable authentication option(s), and the public folders that are exposed to users.

If you wish to perform this configuration, use the Microsoft Management Console (MMC) System Manager console and Active Directory Users/Computers console. In response to your changes, ADS will store the changes within its own directory and apply them to the appropriate Exchange server.

When you create Virtual Web servers and directories with the Exchange Administration tool, they also appear in the Internet Services Manager console. All configuration changes made in the Exchange Administration tool overwrite any changes made to the same items using Internet Services Manager.

To avoid confusion, we strongly suggest you establish and follow a policy of using Internet Services Manager to make changes only to items not available in the Exchange Administration tool.

To configure a front-end server in Exchange 2000, you must select the “This is a front end server” check box in the server’s Properties dialog box, as shown in Figure 15-2. After making your selection, you must stop and restart the Exchange and IIS services, or restart the computer.

Never leave a server half-installed waiting for a restart before the changes take effect. Doing so can cause many hours of troubleshooting to figure out why the server failed after a scheduled—or even an unscheduled (power outage)—shutdown.

Figure 15-2: Configuring a front-end server
Once this change takes effect, you must instruct the HTTP, POP3, and IMAP4 components of Exchange to redirect all traffic to a back-end server. That server (not on the front-end server) must contain the user’s mailbox. The Exchange Information Store on the front-end server is not altered. Only calls from MAPI clients will reach it, as HTTP, POP3, and IMAP4 are being redirected.

Configure all front-end servers immediately after installing Exchange 2000 Server. Doing so establishes your topology and server roles explicitly and helps avoid the need to move users and reconfigure.

User administration
By default, OWA in Exchange 2000 is enabled for all users. To change this value, use the MMC, Active Directory Users, and Computers. Select View Advanced Features to enable the Exchange Advanced tab in the user properties (otherwise it will be hidden).

Find the Protocols Settings button. This allows you to modify the HTTP, Post Office Protocol version 3 (POP3), and IMAP4 access settings for the user, as shown in Figure 15-3.

Virtual servers and virtual directories
Virtual servers enable you to create separate Web server instances for different types of users— for example, internal and external users, or different business units or departments. Virtual servers also allow special instances for users with special security profiles or requirements.

If you have never used virtual servers for this purpose before, you will probably find it easier than you think. There are only three property pages to focus your attention on to create HTTP virtual servers, as shown in Figure 15-4.
✦ **General:** Use this tab to configure the unique virtual server identification. The values are host header, IP address, and port. The General tab also enables you to configure the number of connections available for content areas, such as private mailboxes, or for specific portions of the public folder tree and logging.

When you are configuring virtual server identification, the combination of identification values for each virtual server must be unique.

✦ **Access:** Use this tab to configure the type of authentication (for example, anonymous) used for access to secure content.

✦ **Security:** Use this tab to configure administrator permissions to the virtual server. Changes made here do not affect client connections.

**Users connecting to a virtual server**

To connect to a virtual server, a user browser must specify, in a URL, a specific IP address and port, or host header name and port, that uniquely identifies the specific server the user wants to connect to. In both cases, the port does not need to be specified if it is the default TCP port 80, as shown in Figure 15-5.

So that the client can find the server on your network, register the host header name as a host record in DNS, add it to the client’s host file or match the server’s computer name if the connection occurs on an intranet. We prefer to see you rely on DNS to minimize later administrative work altering local host files.
Disabling virtual servers

You can stop, start, or pause each virtual server: select the server, right-click, and choose the appropriate option.

**Note**

Only the Internet Services Manager allows you to administer the default Exchange virtual server.

If you stop the default Exchange virtual server, you are also stopping the IIS default Web server. If you want to eliminate Exchange access but allow this Web server to be available, remove the Exchange, Exadmin, and public virtual directories. If you wish to go further, you can configure security to disable access.

**Caution**

If you remove virtual directories, you effectively disable management of public folders on the server.

Virtual servers allow you to configure multiple virtual directories to point to different public folders or to the private mailbox store. Virtual directories can even be created within other virtual directories. You can use this technique to create a Web-accessible hierarchy. Users could navigate your hierarchy using Web folders, or your development team could take advantage of it in their Web applications.

**Tip**

Think of virtual directories as the enhanced public folder shortcuts used in previous version of Exchange.
Multilingual support

OWA in Exchange 2000 supports nine languages: English, French, German, Japanese, Italian, Spanish, Chinese (traditional), Chinese (simplified), and Korean. For inquiries on support for other languages, consult Microsoft’s knowledge base by navigating to www.Microsoft.com/Exchange and selecting Support.

Summary

OWA in Exchange 2000 provides a mix of features and solves common problems of access, bandwidth, and cost. For certain uses — such as roving users — it is hard to match. Although earlier versions created significant stress on server resources, this version is much improved. With new functionality and better security, Exchange 2000 OWA may be a useful component of many messaging environments.
Microsoft has continued to improve installation, deployment, and coexistence in Exchange 2000. They have added new server collaboration capabilities in Conferencing Server and for Internet users. Putting this all together for both simple and complex messaging environments can appear challenging. However, once broken down into manageable units, sophisticated installations are well within the capability of modern Administrators and consultants.

In this part of the book, you will be able to follow our thinking through numerous planning, installation, and topology design considerations. We walk you through a quick install and then examine a more thoughtful installation. We share with you our insights on how to map software to hardware. You will be introduced to Conferencing Server. We also introduce you to the various ways you can migrate from or coexist with cc:Mail, MS Mail, Notes, or other legacy environments. In addition, we have a whole chapter where we focus on the improvements made for using Exchange with and on the Internet.
Performing a Quick Install

We were hesitant to include this chapter in our book because we so passionately believe there is a profound difference between installing the software without a plan and configuring an Exchange environment. In a sense, the difference between planning and installing versus pressing install is also the difference between a smart install and a not-so-smart install. Like many of you, we are the kind of people who start building tools, toys, and objects without reading the instructions or following a plan of any sort (other than “Let’s go”) — until we run into an impasse. Then we have to go back and fix the problem, often undoing hours of work. But for almost every Exchange scenario you will encounter, planning, preparing, and arbitrating through the myriad of possible choices before you begin is the smart way. We believe you will be more likely to use the smart install methods (covered in Chapter 20) that include planning and configuration than what we will walk you through in this chapter except in a few rare cases. What we cover in this chapter is useful for a “Play LAN,” some simplistic pilots, or simply to get a bit of “hands-on” before delving further into the book. For all other scenarios, we suggest you spend more time on the planning and preparation and do a complete configuration at the time of install.

Installing a Server

Before you can install Exchange 2000, you must have Windows 2000 installed. A basic install with a few caveats is acceptable. For our Play LAN, we installed a basic Windows 2000 Advanced Server installation using mostly default settings. We’ll show you the relevant non-default settings we chose as we progress through the installation. Once that was complete, we installed Exchange 2000, using the versions that were included in the Corporate Preview set. Windows 2000 Advanced Server was build 2195. Let’s cover a bit of the Windows 2000 install, highlighting the important steps.
Windows 2000 install highlights

The machine we used had dual Pentium III 450 MHz processors, 256MB of memory, and two 20GB, ATA66, ATAPI hard drives. A single Pentium processor would also have been adequate. Memory, disk space, multiple disk spindles, and a fast processor were all factors that drove our choice.

If you have a previously installed beta or pre-beta Windows 2000 or Exchange version on your machine, you should reformat the drive, deleting all old files. Results can be unpredictable if you install one beta or released code over previous betas.

A topological picture of what we created is shown in Figure 16-1.

Figure 16-1: Logical and physical view of Play LAN

To install Windows 2000 on the first server, we boot our server under NT 4.0 and then start the install from the Windows 2000 CD.

Tip
Install Windows 2000 Advanced Server on a fresh NTFS partition created by Windows 2000, choosing the default options for installation.
We install into a clean NTFS partition under the subdirectory name WinNT2K to avoid any confusion with other WinNT subdirectories on the machine. We walk through the multiple boots of the installation process, supplying information in response to the dialog boxes. The specific responses we gave to these dialog boxes are shown in Table 16-1.

### Table 16-1

**Our Responses to the Installation Dialog Box Queries**

<table>
<thead>
<tr>
<th>Dialog Box Query</th>
<th>Our Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name and Organization</td>
<td>@Ilmarin, LLC</td>
</tr>
<tr>
<td>Licensing Mode</td>
<td>Per Seat (Exchange supports Per Seat Licensing only)</td>
</tr>
<tr>
<td>Computer Name</td>
<td>Lorellin</td>
</tr>
<tr>
<td>Password for Administrator Account</td>
<td>(We don’t share this even with our mothers!)</td>
</tr>
<tr>
<td>Windows 2000 Components</td>
<td>Default—except we also included file and print services, which you will not normally do on an Exchange Server unless it is germane to your pilot or Play LAN plans. We select the NNTP component for IIS.</td>
</tr>
<tr>
<td>Terminal Services</td>
<td>Not chosen, as it was beyond the scope of a simple install.</td>
</tr>
<tr>
<td>Display Settings</td>
<td>800 X 600 or more</td>
</tr>
<tr>
<td>Time and Date</td>
<td>Current</td>
</tr>
</tbody>
</table>

Many of the entries in our Windows 2000 install are made using the Windows 2000 setup program and before the Windows 2000 Configuration Wizard.

**Tip**

If you intend to load NNTP when you load Exchange, you need to select it (under IIS) when loading Windows 2000—failure to do so will either stop your install of Exchange later or prevent you from installing the NNTP service.

Use the Configuration Wizard to install Active Directory, DHCP, and DNS.

To install Active Directory, the system asks you a number of questions to set up the domain. Active Directory requires an NTFS partition of the type that comes with Windows 2000, which we took care of when formatting the hard drives. Figure 16-2 shows how to specify whether you want to create a domain controller for a new domain or have the server become an additional domain controller for an existing domain.
We specify that we want a new controller in a new domain tree, as shown in Figure 16-3. Had there already been a controller and domain tree in existence (as there would be when a second server is added to Play LAN), we could choose to join the existing domain tree, or we could still create a second tree.

There are warnings in Figure 16-3 associated with the option of joining an existing domain. When we add a second server later in this chapter, we join it to the existing domain. That ensures that there are no local accounts we care about, as well as no cryptographic keys. We also ensure there are no encrypted e-mails using keys on this server. In our case we have neither, because the second server is also a fresh install.
The next decision is whether to join a forest or create a new one. We choose to create a new one in Figure 16-4. This choice is required if you are the first server in the Active Directory organization. In Active Directory, you are required to have at least one forest for other objects to exist in.

**Figure 16-4: Creating a forest**

For more information on the implications of joining a domain versus creating a new one, or joining or creating a forest, you will want to refer to the *Windows 2000 Server Administrator’s Bible*, published by IDG Books Worldwide, Inc.

To give us a chance to play with the interaction between Windows 2000/Exchange 2000 and NT4/Exchange 5.5, we specify that we want permissions set for compatibility with pre-Windows 2000 servers, as illustrated in Figure 16-5.

**Figure 16-5: Permissions selections**
Figure 16-6 shows a summary of our selections. All of the information on this screen is important and used later to connect clients or add servers. Note that the system tried to separate the log files from the database by assigning each to a different drive letter. Exchange does this in an attempt to assign different drive spindles for each for improved performance. Whether or not assigning different drive letters will also assign different spindles is determined by whether you have partitioned your drives. We discuss this issue in more detail in Chapter 19.

Figure 16-6: Summary of selected options

This summary screen is a good one to capture using a screen capture program like HyperSnap as a really simple and fast way to document your selections. A demo version of HyperSnap is on the CD.

The domain name we use is the one we registered with an Internet Corporation for Approved Names and Numbers (ICANN) accredited registration company. (The best way for you to find a domain name is to search on ICAAN and look at their list of accredited entities.) In our case, the domain name is Ilmarin.com (see Figure 16-6). Windows 2000 also installs and sets up DNS for this computer. When asked for a workgroup name, we choose “headquarters.”

We click OK and the system goes on to configure Active Directory, as shown in Figure 16-7.
Now we take a short walk outside and enjoy the cloud formations, because it will take a while for this step to complete, as noted in Figure 16-8.

You can verify that Active Directory is installed by looking in the event log for errant problems, or by opening Programs ➪ Administrative Tools, and looking for the Active Directory programs, as shown in Figure 16-9. If there are no entries, Active Directory is not loaded. If you see Active Directory Domains and Trusts, Active Directory Sites and Services, and Active Directory Users and Computers, all is probably well.
For a step-by-step delineation of a Windows 2000 installation, and commentary on why you should select one choice over another, you should take a look at the *Windows 2000 Server Bible*, published by IDG Books Worldwide, Inc. This chapter includes just an overview of the Windows 2000 Server installation. While we cover many of the steps, we do not cover all, and we dwell on those of particular interest to Exchange Administrators only.

Before we go on to load Exchange, we update to Windows 2000 SP1, ensuring that we have Internet access through a connection to the network.

Always ensure that you check for any operating system updates, and implement them before you go on to load Exchange. Updates and patches can alter registry settings, DLLs, or protocols that Exchange also alters or relies on.

### Installing Exchange 2000

After successfully installing Windows 2000, we load the Exchange 2000 CD, which autostarts and brings up the Exchange 2000 Installation Wizard. We choose Exchange Server setup shown in Figure 16-10.

We agree to the licensing and enter the CD key.

The old 11111-11111-11111-11111-11111 will generally not work any more. We suggest you write CD keys in marker on the CD so they will forever be handy. You should also make a backup copy of the key for the person that manages legal licensing to maintain in their file.
If you select only the default settings, you are not installing NNTP when you load Windows 2000 IIS. (The screen shot in Figure 16-11 shows you what happens if you fail to load NNTP.) The solution is to either quit the install and add it now or continue without the NNTP service and deal with it later.
In the Component Selection screen, the default selects “Microsoft Exchange Messaging and Collaboration Services” and “Microsoft Exchange System Management Tools,” as illustrated in Figure 16-12. We also check to see if we have sufficient disk space to load all the components, as well as appropriate space for our data. As you can see from Figure 16-12, we have 810MB on C and over 4GB on D where we will be loading Exchange.

Figure 16-12: Exchange 2000 install component selection

Next, we are prompted to specify whether we want to create a new Exchange Organization or join an existing Exchange 5.5 Organization. For Play LAN, we choose “Create a new Exchange Organization,” as shown in Figure 16-13.

We are then prompted to provide an organization name. An organization, which contains all Exchange objects, acts as the top of the Exchange hierarchy. We choose Ilmarin as our organization name, as illustrated in Figure 16-14.

Note

If you choose a name that has characters that are not acceptable to X.400 or other messaging systems, you may get a warning from Exchange noting that while the characters are acceptable to Exchange, they may cause problems with other systems. It is better to select names that satisfy the popular standards.
Chapter 16  ✦  Performing a Quick Install

Figure 16-13: Installation Type dialog box

Figure 16-14: Organization Name dialog box
After the Organization Name dialog box, we come to the screen on licensing. Because Exchange only supports Per Seat licensing, we need to either select that when we load Windows 2000 or commit to handling licensing properly and acknowledging that we know Exchange does not allow concurrent licensing, as shown in Figure 16-15.

![Licensing Agreement dialog box](image1)

**Figure 16-15:** Licensing Agreement dialog box

Next, we come to a confirmation screen where we reiterate that we want the typical install for Exchange and the System Management Tools. When we click OK, the system begins copying and generates a warning, as shown in Figure 16-16, regarding some of our selections. In our case, we are not concerned about the lack of security of the domain on Play LAN. In a more sophisticated setting, we would heed the warning.

![WARNING!](image2)

**Figure 16-16:** WARNING!

The installation then copies files and makes alterations to subsystems like the Active Directory, where it makes hundreds of changes to the schema to add the objects of interest that Exchange needs.
Adding Exchange to Active Directory causes significant changes to the schema. Any change to the schema requires a complete re-replication. In our Play LAN, that’s not a big deal. In real-world environments, however, schema changes require significant planning, thought, and communication, as the implications on latency and WAN bandwidth are serious. An error here could affect service levels throughout your company for days.

This step takes a while (more than 30 minutes). You can use the time to browse the manuals, increase your skill with Free Cell, get a sandwich, or surge ahead in this book. We spent our time writing the chapter text and documenting our selections. Because the stars were in alignment that day, all the services started properly for us in the Post Installation procedure that followed. If it goes for you like it did for us, you will be greeted by a screen that tells you that you have completed the installation successfully.

**Tip**

After a new Exchange install, the services can take some time to start up. Patience is a virtue at this point.

Now take a look at Start ➪ Programs and you will see that you have a Microsoft Exchange entry that includes an Active Directory snap-in and the System Manager, as shown in Figure 16-17.

Install any hot fixes or service packs for the version of Exchange you have just loaded. In our case, there aren’t any. We move on to bringing up the Active Directory Users and Computers snap-in.

**Adding users**

As you go through the following section on adding users, note the number of groups and users that the system installed for you. Select Users ➪ Action ➪ New ➪ User and add a new user, as we do in Figure 16-18.
The second screen of the user creation process, shown in Figure 16-19, enables you to specify where you want an Exchange mailbox created and notifies you about the details.

We added a few more users, and you should do the same—to gain some experience and to populate your Active Directory.

**Tip**  
If you find that you cannot select New, it may be that you have a user or group highlighted. You will need to deselect them.
Creating groups

There are two types of security groups that can be created: Domain local and Global. There are also three types of distribution groups: the two that have been previously mentioned, and Universal, shown in Figure 16-20. We create one of each by checking the appropriate box. We call the Domain local group “WritersLocal,” the Universal group “WritersUniversal,” and the Global group “WritersGlobal” to illustrate their differences.

In a Domain, local group membership can be from any trusted domain. In a Global group, membership can only be from the local domain, but it is global in scope. Global groups can only have one level of nesting. The Universal group is similar to an Exchange 5.X DL. Groups are covered in more detail in Chapter 6.

![Figure 16-20: Global and local group creation](image)

In the process of creation, we receive a warning from Exchange, as illustrated in Figure 16-21. We check the box that enables us to create an Exchange e-mail address, which completes the creation for illustration purposes. When creating the Universal group, the box that creates the e-mail address was already checked by default.

![Figure 16-21: Universal group](image)
We add members to the group by highlighting the group, double-clicking to open it and then finding the members property page. Select Add Members and double-click on all members you want to add, as shown in Figure 16-22.

![Figure 16-22: Adding members to a group](image)

**Creating a Public folder**

We create a Public folder by invoking the Exchange System Manager. We highlight Public folders and then choose Action ➪ New ➪ Public folder. This brings up the Public folder Properties page, as illustrated in Figure 16-23.

![Figure 16-23: Creating a Public folder](image)

Now let's move on to the process of adding a second server to our Play LAN.
Installing a Second Server

We install a second Windows 2000 Advanced Server to make our Play LAN a bit more robust and to learn about the interaction of a two-server Exchange environment. There are a number of different ways a second server in your organization could be added. For our Play LAN, we add the server to the existing domain and then to the existing Exchange site. Depending on what you want to be able to demonstrate and work with in your Play LAN, one server may be sufficient—or you may decide that you want that second server. If so, you may want to do a more complex configuration than what we did. Variations of interest would include combinations from Table 16-2, which you should fill in with values that are appropriate for you (we used our Play LAN choices for the purpose of illustration). Some choices are mutually exclusive. The entries in Table 16-2 are made in both Windows 2000 and Exchange 2000 installation and configuration processes.

### Table 16-2

<table>
<thead>
<tr>
<th>Action</th>
<th>Server 1</th>
<th>Server 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC for new domain</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Additional DC for existing domain</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Create a new tree</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Create a domain in an existing tree</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Create a new forest</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Place domain tree in existing forest</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Create new Exchange Site</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Join an existing Exchange Site</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Connecting with a Client

An installation, even a quick installation, is not complete until you ensure connectivity from a client. For that reason, we continue this chapter with the installation of an Outlook 2000 client and the connection of that client to the Exchange server on PlayLAN.
Outlook 2000 install and connection

We begin our Outlook 2000 installation by inserting the CD, which brings up the Setup Wizard. We enter our name, initials, and organization, and then move to the licensing page where we accept the terms. We choose Install Now on the next screen.

Once the install completes, we click on the Outlook icon. This brings up the Startup Wizard, as in Figure 16-24.

![Outlook 2000 Startup Wizard](image1)

**Figure 16-24:** Outlook 2000 Startup Wizard

Outlook asks if we want to configure “Exchange” or “none of the above” and we choose “Exchange.” When the E-Mail Service Options screen comes up, we choose “Corporate or Workgroup,” as in Figure 16-25.

![E-mail Service Options](image2)

**Figure 16-25: E-mail services options**
The Outlook Wizard then inquires which information services we want to use. We choose “Microsoft Exchange Server” and “Internet E-mail,” as shown in Figure 16-26.

![Figure 16-26: Information services](image)

The Wizard brings up a screen to configure for Exchange server access. We fill in the name of the Exchange server and the mailbox name, as captured in Figure 16-27. We state on the next screen we do not travel with this computer.

![Figure 16-27: Configuring Exchange server access](image)

The Setup Wizard brings up the screen for Internet Access configuration as shown in Figure 16-28 because we selected both Internet and Exchange earlier, but we will skip that here, as it is covered in other chapters.
We now have Internet e-mail, Microsoft Exchange Server, and an Outlook Address Book configured.

**Outlook Express install and connection**

To connect with Outlook Express, we click on the Outlook Express icon that brought up the Internet Connection Wizard. We choose “Use an existing Internet mail account” in the first screen that appears, as shown in Figure 16-29.
The Wizard then detects our account, as shown in Figure 16-30.

![Figure 16-30: Detecting an Internet account](image)

Once we accept the settings, we are finished. We chose not to import all the addresses and mail from our other mail client, although we could have.

**What’s Missing?**

What we did was the most basic of installs, without dealing with such issues as naming conventions, backup, and restore considerations. The type of install we just did is a good way to gain some hands-on experience or to test some specific feature. It is also a good way to explore the Microsoft Management Console (MMC), especially for groups and users. But it is not a good way to configure and install a production Exchange environment. That requires preparation, planning, and configuration that goes beyond a simple install. Performing a smart install will be covered in Chapter 20.

**Summary**

A basic Exchange 2000 installation is easy. It requires a quick install of Windows 2000 Server and then relies on some of the decisions you made during your Windows 2000 installation. Installing Windows 2000, Exchange 2000, and Outlook
2000 “quickly” can take half a day for a basic, bare bones install with hardware that is on the HAL and no equivocation on the answers to the questions presented by the wizards.

If you are planning a more thorough configuration that will more appropriately reflect the needs of your organization or allows you to delve into the more complex configurations possible with Exchange 2000, then you must prepare, plan, and install Exchange 2000 with more effort. In addition, a more complex installation may involve issues of migration or coexistence.
Planning an Exchange Implementation

This chapter covers the basics of preparing for an Exchange installation and deployment. It reviews ideas in planning, organizing, assigning roles, and building a plan for an Exchange implementation that meets your organization’s needs.

Plan, Test, Implement

Why plan? Why test? Why not just implement?

The arguments against planning and testing before implementation usually involve cost, either of money or of time. People sometimes leave planning and testing out of the budget to suggest that without them, a project will be delivered more cheaply and more quickly. But experience indicates otherwise: even with serious money and time allocated to planning and testing, it can be difficult to keep on schedule and under budget. Without planning and testing, it is virtually impossible.

There are myriad reasons for schedule and budget difficulties. For one thing, it is difficult to orchestrate communication and deliverables in a complex endeavor — especially if deadlines, process, and timing are communicated unclearly or not at all. Sometimes a plan is so inadequate that it is misleading. Testing helps you to vet a plan and find its flaws; without testing, iteration and adaptation of the plan usually occur too late, driving up the cost or ruining the schedule, or both.

Interestingly, some senior technical executives claim that the reason they do not plan or test is that the business people in their own organizations demand they focus the dollars on
implementation. The same people, however, generally insist that you make credible efforts to deliver with few surprises. And the best way to diminish surprises is by planning, testing, and predicting. The predications should be based on the results of actual tests and responsible plans.

Although many types of plans work, the following plan delivers rapid deployment with scalable, realistic, cost-effective controls for most organizations.

**How to Plan**

Plan using a top-down, orderly approach. Start with identifying the business and technical objectives for the project—and once they are established, adhere to them. Get a definition for the business reasons for the project. What business utility do the users or business units get if the project is delivered on time and on budget? What functionality do they have now? What will they lose? What will they gain? What changes in functionality are neither a gain nor a loss?

Users often perceive any change from what they know to be a loss. Although it may appear that users lose no business utility in some aspect of a change, they may not see it that way.

For technical objectives, you should identify

- Standards that you wish to adhere to
- Tasks to be performed
- Resource requirements
- Training requirements
- Necessary security
- Preferred methods
- Procedures

You should also establish how the plan fits with others—especially WAN and Windows 2000 plans. For example, if you expect to migrate Exchange 5.x resources to Exchange 2000 during your Windows 2000 deployment, it will probably make sense to have a native-mode Windows 2000 domain available (for Universal groups, SID-history, and so on). Find out when the Windows 2000 project team plans to switch to native mode, and adjust your plan accordingly.

Plans consist of lists of resources, prescriptions for allocating these resources, and schedules of activities and events. You should divide the plan into tasks that are small enough to be predictable and controllable—even if there are a lot of them, they can be executed quickly and deftly. Strive to plan to the department-level, specifically tasking the Exchange installation by server, workgroup, and connector.
As obstacles arise, identify and document them. Deal with them as further tasks, instead of combining them into existing assignments. If you combine them into existing assignments without redefining the task and rescheduling, the person assigned is likely to miss the original deadline, or not even complete the original task.

The Microsoft Foundation (MSF) model is a valuable resource for planning. MSF was originally conceived and used by Microsoft as a software-development process, but can be easily adapted to fit many types of complex projects—such as an Exchange 2000 deployment.

Anatomy of an Implementation Plan

Although some plan components vary from organization to organization and from implementation to implementation, there seems to be a core of components with universal utility.

A successful implementation plan should include the following components:

- **Executive Summary**: Most executives will not read the whole plan, yet need to know the basics, including schedule, resource and cost estimates, and the expected utility.
- **Mission Statement**: Mission statements are hard to write well, but help everyone involved to understand the reason for the project.
- **Objectives**: Specific, measurable goals and the associated schedule.
- **Budget**: Cost in monetary and resource units.

A corporate standards installation plan should include the following components:

- Topology
- Time lines
- Checklists
- Guidelines

A training plan should include the following components:

- Evaluation of existing training plans
- Definition of training objectives
- Definition of training team
- Mapping of needs to models for acquiring skills
- Budget
A deployment support plan might include the following components:

- Evaluation of existing support plans
- Definition of support objectives
- Membership of deployment support team
- Deployment support plan
- Corporate Support Transition Plan
- Evaluation of previous support transition plans
- Definition of support transition objectives
- Membership of transition management team
- Transition plan

The Plan

In any Exchange 2000 server deployment and implementation plan, one of the chief resources is human. To get the right people assigned to the right tasks, you need to consider the roles, responsibilities, and skills required.

Roles

Early in the planning, define the key roles individuals will play in the project. Whether your project has 30 people assigned to it or 5, it is still important to assign all the roles that are required by the project. If the group is small, everyone will probably have multiple roles.

It is crucial to identify and assign qualified people with the skills to fit these roles. In cases where the skills are not already in place, part of the project may be to acquire those skills through training or education. In many organizations, some roles may already be defined although the ownership of the responsibilities resides in other groups. For example, Windows 2000 Active Directory Service administration may be assigned to people outside the messaging group. If so, ascertain whether people will assist you in the project, or provide yet another obstacle to overcome—and formulate your plan accordingly.

At different phases of planning and implementation, different human resources and roles will be required. Your plan should include the specific number of resources needed during each stage in the process. That number is determined by adding up the estimates of time required for constituent pieces.
Certain laws of scheduling are immutable, such as “It takes about a person-day to do a person-day’s worth of work” or “Holidays happen!” If you are cavalier in your estimates or miss obvious impediments such as holidays, your project will suffer in many ways — including credibility. Make sure when you include holidays you look at all the religious, national, and cultural holidays likely to impact your organization.

Occasionally you can find or create a breakthrough process to save significant time or achieve sufficient parallel processing to provide improvements in cycle time. But usually, the clock is an unforgiving monitor. Be honest about the time metrics, accurate in your estimates (testing helps here), and rigorous in your arithmetic. Holistic planning usually leads to disaster.

**Project manager**

The ideal project manager has insight and connections into the entire organization, and a good sense of how the technology maps to business utility. Highly technical skills, though welcome, are not as important as leadership and organizational skills, as well as an intimate knowledge of existing corporate systems and organizational computing strategy.

A project manager who knows both business and technology may be better able to separate reasonable requests from wasteful ones. On a personal level, a technically astute project manager is more likely to inspire confidence in the technical staff; similarly, business acumen inspires confidence in the business staff. A project manager should at least be familiar with the high-level concepts of Windows 2000, Active Directory, and Exchange 2000. Project managers in smaller organizations should be even more familiar with the technology, since they will be more intimately involved with the technical aspects of the project.

In larger implementations, it may be useful to assign a motivated assistant to the project manager to focus on facilitating inter-departmental communication and coordination. Lack of communication can cause a project to appear to be a failure, even if it has accomplished its goals.

**Make sure the project is technically—and even economically—as good as it can be. Then communicate quickly and sufficiently with the people who need to know about it.**

**Product compatibility engineer**

Ensuring compatibility with existing software and hardware platforms and applications is crucial. The product compatibility engineer needs to have an outstanding knowledge of Windows 2000, Exchange 2000 (as well as Exchange 5.x, if the installation is a migration or coexistence), internally developed applications, third-party add-ons (especially gateways and connectors), the Internet, and end-user requirements.
In some cases, the job of product compatibility engineer is split into two parts—one focusing on hardware and the other on software—because it can be hard to find engineers that know both well.

The product compatibility engineer must learn the new collaboration applications developed for (and in) Exchange, as well as new third-party add-ons—identifying any incompatibilities, requirements, and constraints that may affect the installation (for example, a third-party add-on that has been tested only with an older version of Exchange).

Often, the product compatibility engineer also ensures that all licensing requirements for software are met.

Hardware requirements should be specified down to the configuration level, balancing tradeoffs among economics, scalability, security, bandwidth, reliability, serviceability, vendor reliability, availability, and previous experience.

During and after installation, the project testing engineer and the assistant testing engineers may play a major role in resolving compatibility issues, monitoring the performance of the system, and suggesting alternative configurations to adjust for problems.

**Installation engineer**

The installation engineer is responsible for developing detailed procedures and instructions for installing Exchange server and the client. Excellent organizational and written communication skills are required, as well as a thorough knowledge of Exchange server, Windows 2000, the LAN, the WAN, and applications.

In larger installations or quick-install smaller installations, probably one or more assistants to the lead installation engineer will actually install and configure the clients and perform post-installation troubleshooting. These technicians should be very experienced with the hardware platform, Windows 2000, Exchange server, time management, and client installations, and should be very familiar with using the MMC for administration and troubleshooting.

Almost all the tools have changed from earlier versions of Exchange. Gaining familiarity with the new versions is a worthwhile investment before the installation.

Engineer positions can be filled by outside consultants. In many cases, industry certifications and professional resumes can help to qualify appropriate technical skill levels. Many companies that participate in the Microsoft Solution Provider channel, especially at partner-level, have certified Windows 2000, Exchange 2000, and BackOffice engineers available for projects or day work. Other alternatives include Microsoft Consulting and third-party day labor (body shops). These firms can often supply a higher level of experience than local resources, especially when after-project staffing levels need to be maintained during rollout.
Training lead
Effective end-user training makes users more productive and reduces the need for post-installation support. Training end-users is an important aspect of delivering the business utility you committed to; it can also dramatically affect your ability to maintain the level of service you promised. The person responsible for training should be experienced in Exchange Server and Windows 2000, the business functionality required by the rollout, and the messaging clients used. Training is often outsourced to professional organizations such as Microsoft Certified Training Education Centers (CTEC), where certified Microsoft Exchange training is available.

Project support engineers
Most likely, project support engineers will come from a corporate help desk, MIS, or LAN support group. Since these people will need to identify potential support issues before rolling out Exchange across the organization, they should be part of your early-adopter program or pilot program. Support staff and installation engineers should coordinate before, immediately following, and often long after installation. Choose project support engineers who are able to stick with the project for the long-term.

Responsibilities
Once you know what your human resources look like, you need to assess and group the responsibilities that need to be assigned to them. Complex tasks are often assigned to teams and then subdivided among team members according to ability and experience.

Planning team
Members of the planning team should include a project manager/coordinator and all lead project engineers. The planning team establishes project goals, boundaries, scope, and direction. It must meet regularly to review implementation issues, review milestones, and schedule. This team is also responsible for most planning documentation and communication, internal and external.

The planning team usually owns the difficult task of risk assessment and management. This subject is covered in more detail later in this chapter.

Evaluation and testing team
The evaluation and testing team tests for and determines the level of compatibility with existing software implementations. They test per the standards set by a standards committee and/or the planning team.
Setting standards should not be optional. They are a requirement of any organization trying to contain costs and satisfy reasonable expectations.

The evaluation team might also test any third-party extensions or applications needed for Exchange Server for Windows 2000 in a way that affects Exchange 2000. During implementation, the evaluation team resolves compatibility issues concerning legacy applications, standards, and installations.

Members of the evaluation and testing team often include a product evaluation leader, the evaluation team, and product testing and installation engineers. Some areas of focus for the team include the following:

✦ **Applications developed for Exchange**: Any applications built to use features of Exchange 5.x, such as MAPI, SMTP, the Event service, Collaboration Data Objects (CDO), and so on

✦ **Directory synchronization/replication processes**: Routines for synchronizing or replicating their directory information between two messaging systems

✦ **Connectors**: All Exchange connectors, including EDK-based, custom third-party or Microsoft-provided connectors

✦ **Migration tools and strategies**: Migration tools and methodologies used to migrate, consolidate, or update Exchange resources

✦ **Third-party support software**: Antivirus, backup, message or content filtering, monitoring and alerting software

✦ **Client applications**

### Deployment team

Led by the primary installation engineer, the deployment team enters the project between evaluation and deployment. The deployment team handles the pilot and rollout phases of the implementation. It is critical to choose people who are comfortable with accurate, timely communication. Because of overlapping functions before and after these phases, the deployment team shrinks and grows during the project.

Members of the deployment team include installation engineers, a product evaluation engineer, the product testing leader, and the project support leader.

### Training team

Depending on your organization’s structure and size, the training team may consist of many trainers and support staff, or just one or two people responsible for coordinating and reviewing training. The training team should deliver a knowledge level sufficient to ensure that administrators and support personnel can meet their promised level of service, and that end-users actively use the products.
Members of the training team should include the lead project training engineer and the lead support engineer.

Support team

Once the project pilot and rollout are under way, support team members track and review outstanding issues with the installation, evaluation, and testing team engineers to minimize problems during implementation. The support team can be thought of as a help desk for implementation team members.

The support team is also responsible for developing the support transition plan covered later. Usually, they develop that plan for the planning team to include in the overall plan.

Members of the support team should include the project support engineer, the installation leader, and the testing leader.

Elements of a Successful Plan

The implementation team should have a clear vision of the project’s business and technical objectives—at least the big picture. Early in the project, experienced members should establish general principles and objectives for future activity.

Executive sponsorship and buy-in

During all phases, especially early ones, allies are needed at all levels of the organization and in both the business and technical communities. These allies can be instrumental in fending off critical or turf-protecting attacks from inside or outside the organization. On a more positive note, the prestige of executives can lend legitimacy to the project—presumably they support the project only if it addresses their needs and those of the organization, and only if they expect it to succeed.

Often, the best way to show executives how a project addresses their issues is through presentations, which provide venues for audiences to articulate their concerns. If you use a presentation to win over sponsors, make sure it suits its audience. It does little good to have the CIO attend a slick technical presentation on Exchange Server performance considerations. Instead, at the executive-level try to provide insights into the business utility that the system will deliver, such as workflow and document management. Microsoft has done a credible job providing marketing material targeted to many specific audiences. Executives will also be interested in how the utility will be delivered by the application of technology, but keep any presentation on technology for this topic at an overview level. Their Web site is available to all, and their Solution Provider Channel has access to additional materials. Your Microsoft representative, if you have one, will also have ideas and materials to help get the message across.
Scope

Identify the scope of the project—the affected area of the organization and committed deliverables—early. Your scope should include departments marked for rollout, network infrastructure, legacy messaging systems, and other areas that would be affected by the Exchange implementation.

You should also manage against scope creep, or the inadvertent expansion of the scope. A good example of scope creep is a messaging infrastructure project that begins by covering deployment of Exchange 2000, but ends up needing to deploy an underlying network infrastructure and Windows 2000. Although it may sometimes be appropriate to expand scope, expansion can change the project’s budget and resources significantly. Worse, sometimes no one notices that the scope has changed significantly—while resources, schedule, and budget remain the same—until it’s too late.

To avoid scope creep, avoid ad hoc decisions that can be difficult to alter later. If an issue is real and needs attention, alter the scope, plan, budget, and schedule to address it.

Boundaries

A boundary is a subset of the scope of a project—the subset in which the project may operate without external coordination or approval. Outside of the boundary, external authority is required. For example, consider an underlying Windows 2000 Active Directory architecture consisting of forests, trees, and domains defining the administrative model; an individual may have permissions and rights to manage resources within a tree or domain, but not the entire forest and all trees, which may be necessary for your Exchange implementation. Likewise, project elements and contacts would be identified for those on the other side of this boundary. The following list has some particular areas that should be considered as potential boundaries during your planning:

- Physical network components such as router configurations and firewalls
- Existing connectors and gateways
- Legacy messaging and collaboration systems
- Internet service providers and connectivity
- DNS
- Underlying Network Operating System (NOS) infrastructure
- Departmental network resources, such as guaranteed bandwidth for production applications

Other examples of boundaries are those bounding resource and political geographies. Although they are not always immediately apparent, careful consideration should be paid to boundaries in these two areas.
Fiscal resources

If your implementation appears to be part of a large program with an infinite budget and implementation deadline, the reality is probably that the infinite budget reflects an infinite scope, making the project so large that it can never be completed. What seems like a far-off deadline may be closer than you think if you have underesti-mated the amount of work you need to do. Therefore, it makes sense to be smart about both money and time (usually the two are linked). Fiscal considerations include the following:

✧ Time associated with motivating and directing the human resources
✧ Availability of hardware and software required for server or client upgrades, network upgrades, retrofits, and so on (if the project is of long duration, changes in hardware and software may affect it)
✧ Training and orientation costs (often outsourced or supplied and billed through another department)
✧ Licensing software and future upgrade provisions (remember to include third-party software—many organizations will need to deploy new versions of backup, virus protection, monitoring software, and so on, at significant cost)
✧ Continuing support (a considerable requirement in large organizations)

Human resources

Finding the right people for the job is a challenge, but nonetheless essential. The combination of Windows 2000 and Exchange 2000 is extraordinarily powerful when designed and implemented properly. Assess the skills available within your organization and the human resources that may be required; then map them to each other, and internally or externally acquire the skills that are missing.

Three methods of acquiring human resources are available:

✧ **In-house**: Native to your department, new hires, or on loan from another department. Normally, these people are familiar with the corporate computing environment and departmental needs and organization.
� **Temporary**: Can serve a useful function in many aspects of implementation. The turnover rate of temporary workers, however, is often high, and quality is rarely guaranteed.
✧ **Contract**: Most often supplied through a systems integrator, solution provider, or even directly from Microsoft or other partners. Where high levels of experience are required, the consultant or solution provider can often provide resources that would not be available to you in-house.

Train internal staff to fill skill holes as much as the schedule will allow for. Insist that external resources transfer knowledge to internal teams.
Schedule

Realistic scheduling is absolutely essential. Many factors, some beyond your control, will influence when a particular resource may be available. Although there are 24 hours in a day, the actual availability of resources or periods during which deployment or testing operations can be safely scheduled will be substantially less. When considering how best to manage your project time, take into account people’s other responsibilities, their work habits, burnout, vacations, and holidays, as well as backup schedules, scheduled network down time, and so on.

Almost immutable laws

Almost immutable laws dictate that certain activities will take a set amount of time. You should not act as if cycle times are elastic and all LAN (worse, WAN) wires are infinite in bandwidth. These factors are often overlooked during planning, and are sometimes difficult to gauge even when soberly considered. Therefore, spend time analyzing physical constraints that can adversely affect implementation scheduling. For example, the first installation of an Exchange 2000 Server causes the schema for the entire Windows 2000 enterprise to be extended and re-replicated. Even planning that installation for off-hours may not be adequate and may be delayed additionally by off-hour backups running over the wire used for the re-replication.

**Note**

Testing, coupled with basic arithmetic, is the best way to identify unknown restrictions and obstacles (such as bandwidth restrictions) so you can plan around them.

Politics

Corporate, departmental, and office politics are facts of life in organizations. One of the roles of the project manager is to identify political barriers and obstacles, and either avoid them or find champions and sponsors to overcome them. Avoid any political battle that is not necessary to the success of the project.

Risk management

Successful projects usually have plans that astutely identify and minimize risk. In a perfect world, risks would be identified and preempted well before a call is received from the help desk. During all phases of planning and implementation, be on the alert for areas of process and functionality that may be placed at-risk, particularly when end-users are affected.

The largest risk factors in most messaging environments are system outages (down time), degradation of performance (bottlenecks), pilot error due to insufficient knowledge (ignorance) and lack of interoperability (incompatibility). Interoperability, ignorance, and incompatibility can be addressed during evaluation and testing. Outages and poor performance are more difficult to minimize.
Bottlenecks
In a well-designed and well-implemented topology and system, Exchange can operate very efficiently and be somewhat self-healing. Yet increased functionality of the new Exchange 2000 environment will often result in higher user activity than in previous systems (even earlier versions of Exchange). This increased use may or may not generate an increase in network activity, but the increased use of bandwidth may cause network component failures or saturation of your network or server hardware. To determine areas likely to saturate, analyze current bandwidth use and compare it to estimated traffic generated by the planned Exchange system. The testing team should run performance testing on all back-end hardware and set maximum user and traffic thresholds based on the projected classification of users. Use the Exchange modeling tool included with the server to predict basic user and server load.

Although the Exchange modeling tool is useful for comparing hardware and getting some feel for utilization, it does not simulate connectors’ usage or LAN access. Its effectiveness is limited, but it remains a very useful tool for modeling basic user/server load. For more detail about using the Exchange modeling tool to predict basic user/server load, see Chapter 19.

Some organizations can address bottlenecks by simply oversizing the server hardware and the LAN and WAN pipes. If your budget prevents that solution, place acceptable performance thresholds on backbone and client services and test hardware against these parameters. Remember that the scalable nature of Exchange allows for addition and distribution of services to take advantage of new hardware and multiple pipes.

Down time
Down time can take many forms in an Exchange environment. You are down if more than one user cannot do their work or if a structural component of your system is not functioning. Of the two conditions, the former is usually more serious than the latter. Other than the messaging clients themselves and their transport to the Exchange servers, the following traffic patterns should be considered:

- **Intraserver:** Messages delivered between users on the same physical server
- **Intrasite:** Messages destined for a server within the same site
- **Intersite:** Messages transferred outside a site
- **Internet:** Messages destined for an external system

These patterns are singled out because of the functionality they represent. Your system should be tolerant of failures but not lose functionality. When you assess risks appropriately and design solutions, all you need is funding from management to implement the risk-reducing solutions.
Minimizing risks

You will probably be able to plan for and avoid some risks. Try to quantify those that remain from the business-utility, end-user perspective, assigning numbers to them to measure against service levels. Common risks are up time and end-to-end transport for e-mail transmission. However, be cautious. When asked, “What level of up time do you need?” the business unit usually answers, “99.9%.” But such a fault-tolerant system is not an option for most of us, due to its tremendous cost. To get a more realistic answer, you may want to use a risk assessment table. Build a table of functionality from the business-user perspective. Specify the functionality the system delivers (for example, delivering an e-mail message within the site, or delivering an e-mail message anywhere in the organization). Then map that functionality to acceptable outage to answer these questions:

✦ How long would be acceptable for you to lose the ability to read or write server-based e-mail?
✦ Would it be acceptable if intraserver e-mail was out for half an hour? One hour? Two hours? A day?

This type of questioning helps you implement reasonable service levels. Few users expect to be able to read and write mail every minute of the day. By asking functional questions, you are more likely to avoid whimsical answers (such as “always”) that will cause you to build service levels that are unnecessary and far too expensive.

Once you have the functions and the acceptable outages specified, delineate the critical components required to maintain that functionality and possible bottlenecks. List those bottlenecks that are under your control and those that are not. Review the list with your customers, and clarify possible methods for attaining their desired service levels.

Risk assessment is a serious undertaking; it will be used to determine costs and schedule. Delineate your risk assessment conclusions in your written plan and in meeting notes. For example, include targets for how long your users can wait for restoration of a mailbox, or be without e-mail connectivity to the Internet.

Table 17-1 quantifies acceptable outages for an imaginary system and its critical components. A similar chart can be made for performance considerations.
Table 17-1
Risk Assessment Table

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Critical Components</th>
<th>Acceptable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Outage Duration</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intraserver Messaging</td>
<td>Home Exchange Server, Network Transport, Win2K Authentication</td>
<td>0.5 hours</td>
</tr>
<tr>
<td>Intrasite Messaging</td>
<td>Exchange Servers, Network Transport</td>
<td>3.0 hours</td>
</tr>
<tr>
<td>Intersite Messaging</td>
<td>Connector Servers, Routing Server, Network Transport</td>
<td>8.0 hours</td>
</tr>
<tr>
<td>Internet Messaging</td>
<td>Connector Servers, Routing Server, Network Transport, Internet Service Provider</td>
<td>3.0 hours</td>
</tr>
</tbody>
</table>

Once you have assessed risks, you can minimize them by scaling the hardware and pipes appropriately, and by adding fault tolerance components to the point of meeting acceptable outages. The network transport may be out of scope for the project, so it might be useful to add redundant circuits or routing where appropriate; doing so would further decrease the risk of outage by using resources within the boundary of control.

Time lines

You should schedule in light of the risk assessment table shown in Table 17-1. Remember, in most cases the users are not specifying the outages acceptable after your rollout, but outages expected at any time, including during deployment. Although outage periods may be negotiable, you must communicate explicitly. If your system is in trouble now, and may — during installation — go through some catharsis that increases end-users’ risk of outage, you should speak with the business units prior to installation.

It is a good idea to agree to declare a “period of volatility,” during which the steady state outage and service level agreements are replaced with a lower level of functionality; all parties understand that during this period there may be some problems, but a better world awaits. This level of understanding can often save vast amounts of money and end the pain of installation sooner.

Build time lines based on realistic schedules, and build in some slack for the unexpected. What types of problems are unexpected? Are any problems not dealt with explicitly in your plan? In your plan’s time line, show any potential problems that are not dealt with explicitly as other.
Building a Deployment Plan

The deployment plan in this chapter is structured for an Exchange 2000 implementation. It can be used as a template, in whole or in part, during design, deployment, and implementation. Specific issues dealing with design, scaling of hardware, and other factors are dealt with in detail in other chapters and are highlighted in this chapter only in the context of incorporating them into the implementation plan.

Another way to use this template is as a model to learn a structured approach to planning, deployment, and implementation. Using it this way can help you avoid being overwhelmed by entropy. When this planning method is properly applied, the project becomes a series of steps, each of which is manageable enough to be accomplished on time and within budget.

Concept phase

The concept phase directs implementation and scope. Information gathering, synthesis, and analysis are early duties of the planning team.

Teams that have answers before they ask questions of the business personnel may be applying technology for the sake of technology, and not as a business tool. Even if the team has the right answers, this approach often distances them from the business community and should be avoided.

The planning team should first conduct end-user, departmental, business-unit, and corporate surveys to determine functional requirements of the Exchange system. Messaging and collaboration implementations are about business, not just technology. By connecting with the business functionality needed and expected at the very beginning, you eliminate the possibility that the project will become, or be viewed as, no more than a technical *tour de force*.

Whether your Exchange implementation is autonomous or part of a larger project, its mission and goals within the organization should be stated explicitly and documented.

Objectives

Make sure that all objectives for implementation fall in line with overall corporate goals. Whenever possible, state your project objectives in written, measurable terms and, when applicable, map them to a tangible customer benefit. The following examples will better illustrate this point:
“A customer survey in [month/year] indicated that 40% of customers were not satisfied with the response time of the help desk to electronic correspondence. This problem is caused by the inability of 80% of the help desk staff to access e-mail when they are in other custom applications required by their roles. One goal of the Exchange project is to provide 100% of the help desk support personnel with a Digital Dashboard that allows access to e-mail and our company’s custom application information from different windows on the same desktop. This deliverable will be accomplished within the first six months of the project.”

“An employee survey in [month/year] indicated that 80% of employees felt that e-mail was a reliable medium for business communications and wanted to be able to access their business e-mail in the off-hours when they needed to work from home. Our goal is to give 80% of employees access to their corporate e-mail from their home offices within two months after completion of this project.”

**Project overview**

The planning team should begin with a project overview — often a general description of the implementation in terms of scope, boundaries, and responsibilities. This description, along with well-stated project goals and objectives, provides an early basis for team initiative and coordination. It should be a written, living document. The project overview should also be preceded by an executive summary that includes a concise mission statement. Although the executive summary and mission statement are targeted at upper management and third-party interests, they are useful for the whole team. A team without a clearly understood mission will not deliver as well as a team on a common, clearly articulated, and understood mission.

**Training**

During the early stages — in project marketing and presales, when information is being acquired from business units — you should arrange for your top engineers to train in Windows 2000 and Exchange 2000. For your top personnel, on-the-job training will not suffice, but for some, a few months in the right lab might. At a minimum, be sure that the installation, evaluation, and support leaders receive the knowledge they need. Usually this means formal training, which may best be accomplished by a Microsoft CTEC using Microsoft materials and self-study. Whatever training methods you employ, make sure that you start early and that the selected model will work for your team.

Instead of public enrollment classes, you could develop a private class, using public materials, but focused on your specific environment and project. On the last day of class, the students present, explain, and defend what they have learned in front of the managers. Doing so tends to pull the team together, gives it confidence, and teaches the members how to apply their newfound knowledge. Although such a class does not substitute for bringing in external consulting resources, it can significantly facilitate the learning process.
Development

The development phase of a Windows 2000/Exchange 2000 implementation is one of the most interesting; topologies are designed, tested, and validated. If Windows 2000 has already been rolled out, this is where its design needs to be reconciled to the needs of the Exchange project. Hardware performance is gauged, bandwidth analysis performed, install procedures started, routing established, administrative models designed, and checklists formed.

Design

Potential Windows 2000 and Exchange topologies will undergo scrutiny from several angles, including that of their capability with other BackOffice components such as SMS and IIS. Remember, unlike previous versions of Exchange, which had some dependency on NT but also a lot of autonomy, Exchange 2000 topology is absolutely and inextricably reliant on the Windows 2000 topology.

Server architecture should not be provided in a single design for review, but several. One may provide high performance, another may focus on maximum redundancy, and still another may emphasize ease of administration.

Topology designs accepted for testing and validation should provide a great deal of flexibility at this stage. Testing and validating a design may cause it to be accepted with modifications or modified for further testing. Designs might emerge especially for coexistence and migration strategies. These designs, along with documentation and design objectives, are passed to the evaluation, testing, and validation team for that level of comfort required to commit to a design.

Testing and validation

During testing and validation, all potential designs are put through rigorous modeling and validation tests. We like to think of these tests as “proof of concept” for the proposed designs. You should simulate real-world factors, by such means as network modeling, load simulations, and mock migrations. Analyze the design and modify it, if possible, to target and remedy the point of failure.

During this time, submit hardware specifications and infrastructure pipes (that is, WAN and LAN circuits) for approval as part of the validation of a specific topology.

In the testing and validation phase, you should verify that all the critical pieces of the design will work the way you expect them to. This is your chance to identify major technical and process roadblocks to your project. Some of the things you will want to test for include the following:

✦ Stress
✦ Compatibility
✦ Server and software configurations
During this phase, it is extremely important to control changes strictly and to document diligently.

**Checklists**
Once the testing team has validated and approved the selected designs, it should provide basic, written technical checklists for installation and follow-up. To some, checklists seem like overkill; but they are the best way to ensure that successes can be replicated and failures diminished or eliminated.

**Rollout**
The team should also develop basic, written rollout procedures for the selected designs. Included in the procedures should be detailed descriptions of server installations, client installation, connections, migration, coexistence procedures, and quality control, along with the time requirement for each of these steps. The procedures are honed further during the pilot phase.

Once the project is ready to move into the pilot phase, the planning team should review the work completed during the development phase, incorporate the lessons from the successes and failures, update the objectives and budget, and document objectives for the pilot phase.

**Pilot**
The pilot phase should build on the lessons of the testing and validation done in development. Its purpose is to verify that all current and expected applications will function correctly in the new environment, and that the new environment will deliver the expected business utility and service levels. The pilot phase also establishes performance monitoring and training requirements, and hones installation procedures and checklists. During this phase, you should pay attention to time requirements for installation, migration, and support.

**Selection**
Selecting the right department(s) to be involved in the pilot is crucial. Select groups based on the following criteria:

- Enthusiasm about the project
- Adequate technical level of expertise
- Eagerness for increased functionality
Meet with department heads and end-users to set the start date and expectations. Supply department heads and other executive-level stakeholders with specifics of the pilot implementation and testing plan. Departments must be aware of the potential impact of pilot testing and agree to work within its confines.

**Preparation**
Thoroughly prepare your team and the customer for your pilot rollout. If you or your customer are not ready, postpone the rollout; a serious failure of the pilot can mean shelving or scrapping the project. Although a pilot effort is technically still a test environment, to the end-users in the pilot it is a production environment.

**Contacts**
In each department, identify a primary departmental liaison and a backup contact. The primary contact should be technically adept, have the time to participate actively, and work well within the department. This person should also be very familiar with any systems currently in use within the department and have a good knowledge of any legacy e-mail system used.

**Communication**
Provide the pilot group with a mechanism for regular written and oral feedback to the deployment team. Public folders are usually an excellent method of collecting feedback and addressing general support questions. If you use public folders, give the users instructions for using them. In short, ensure that each user can report and escalate all issues.

**Documentation**
Thoroughly document all aspects of the pilot environment to ensure it is entirely reproducible. Pilot testing is a scientific process, and if variables are not documented, replicating success and troubleshooting problems may not be possible.

**Back-end services**
Sometimes it is possible to implement backbone and other back-end service components completely, with little disruption in service, even in a pilot period. Because Exchange 2000 is inextricably linked to Windows 2000, this may not be possible in your project, but if it is, you should consider this option carefully. Since disrupting backbone and back-end service carries the most risk, try to have the pilot’s backbone and back-end services in place for at least one week before implementing the pilot.
Rollback procedures
No plan is complete without rollback procedures at all stages of implementation, and rollback procedures are especially important in the pilot phase. Although no one expects to have to roll back, your plan should delineate exactly how a rollback would be accomplished. A formal rollback plan should protect you even from otherwise unrecoverable errors, such as destroying data (entering `delete *.*` erroneously); how would you restore the end-user environment to its previous state? This question needs an explicit, written answer. Catastrophe or some unexpected incompatibility can happen, especially in this early phase.

Your plan should also estimate how long it takes to accomplish a rollback. That estimate will help determine the point beyond which a rollback is no longer possible.

Consistency of techniques is important. For the pilot, use rollout techniques very similar to those planned for the production rollout. If you cut corners, you may encounter problems in production you never saw in the pilot.

Use the same support staff in the pilot and production rollouts. Give all available members of the installation and evaluation teams some hands-on experience during the pilot, and have them thoroughly document the process and any problems.

Plan review
After completing the pilot, review all procedures, techniques, checklists, and schedules. Adjust the time line and final design. Survey the pilot end-users and liaisons to assess whether their expectations were set appropriately and met.

Once you have addressed all issues, present the complete results of the pilot phase formally to the decision-makers. This presentation should secure final approval for departmental rollouts and ground everyone in the reality of the project. Present the pilot documentation and the real experiences of the pilot, and explain how problems arose and were handled. If necessary, make the case for adjusting the schedule or budget, using the pilot documentation as supporting material. Request formal acceptance of the pilot tests and adjustments to the plan, and secure permission to continue departmental rollout.

Rollout
The rollout itself will strongly resemble your pilot, if the pilot was designed properly. It should be separated into manageable units; this often translates into departmental units. Using this approach, your rollout is in fact the aggregate of departmental rollouts and should be scheduled as such. Before you roll out a particular department, perform an audit for unique requirements, such as custom applications.
Adequate on-site support

Each department must assign adequate local staff and a local support liaison to work closely with the installation team during the rollout. The local connection is especially important in international projects, or in other projects where you are installing remotely. The local support liaison provides a technical point of contact, and ensures local involvement of both the deployment group and end-users. End-user and business-unit issues should be filtered through this person, if possible, before being passed to the support staff. This strategy allows limited support resources to concentrate on the most important issues.

The local support staff must document and clarify all problems that are encountered. Information gathered should include the following:

- User name and workstation experiencing problem
- Time and date of problem
- Specifics of problem
- Ability to reproduce problem
- Preliminary troubleshooting efforts
- Resolution or proposed resolution, if any
- Severity of problem

You can use this information to gauge the effectiveness of support staff, perform triage, provide detail for pattern recognition analysis as well as trend analysis, and report bugs.

Off-site support

The support engineer should alert off-site support staff of the departmental rollout schedule and any special considerations involved in the rollout. Alerts are especially important when rollouts are happening in off-hours.

Schedule

Scheduling a rollout requires balancing many variables. Use the data you have to determine time lines for particular tasks. Combine the data from departmental units to schedule the business units they are part of, but also ask the business units how they want to be scheduled; they often have valuable insights into how it should be done. For instance, it may be inappropriate to do a roll out in Europe in August, when people are on holiday — or that may be just the right time to do it. The business community will lead you; try to accommodate their desires, and explain any cases where you cannot. Gain acceptance from each department for its scheduled migration and installations. If a department declines a proposed schedule, request the reason and a counteroffer date.
When selecting the order of departments, begin with the smaller, less complex installations and work your way to the more complex. As the installation teams become more confident in their procedures, they will better handle new, more difficult situations.

**Assigning resources**

The pilot installation provides the logic for resource assignments. Using the actual time and resource requirements from the smaller scale pilot rollout, project your requirements for each departmental rollout. If additional resources are needed to complete departmental rollouts, request them and budget for them.

During the rollout, if a team is ahead of schedule or has dormant resources, don’t be afraid to redistribute. Make your installation teams aware of this model before rollout begins, so they will not feel punished for being ahead of schedule.

**The moment of truth**

All your efforts in the planning, development, and pilot phases were in preparation for the rollout phase. Now, although you should continue to seek feedback, it is important not to try and micromanage the deployment. Require updates of successes and problems during the rollout. Repeat and retry where you encounter real problems that cannot be overcome with experience. Keep in touch with the departmental liaison or department head during implementation to allay any fears or reservations.

**Dealing with unexpected problems**

Despite due diligence and practice for the actual rollout, snags are bound to come up. Keep your wits about you during these times, and keep your executive sponsors involved. Continued feedback from all parties involved is a must.

**Documentation and checklists**

Document each stage of the rollout immediately. This is especially important when parallel deployment efforts across units are scheduled.

A large portion of the documentation you collect will be completed installation forms and checklists. Gather, compile, and review this documentation. Do not place an undue administrative burden on your installation teams by requiring them to provide polished documentation; simply request complete, legible content.

Any problems encountered during the rollout, whether resolved or not, should be documented and passed to the support team.
Post-Installation

After installation, monitor your system and request more feedback from end-users.

Monitoring

Tools to help you monitor your system include the following:

✦ Windows 2000 and Exchange logs, especially event logs
✦ Windows 2000 Performance monitor counters
✦ Server and link monitors provided with Exchange Server

You should monitor both activity and performance.

Activity

Activity to monitor falls into these categories:

✦ Server: Look for indications of failure, symptoms of pending failure, or stoppage. Exchange and Windows 2000 provide tools to monitor many aspects required for a healthy, functional system. These include monitors for message delivery between and within servers, for Active Directory Services, and for many hardware problems. Analyzing communication between servers can help determine whether, and where, topology adjustments are necessary. Pay particular attention to the service status of all servers, and queue levels of individual components such as the MTA and IMC. Link monitors and server monitors provided with Exchange are excellent tools for gauging the health of the system. Be prepared for problems early after rollout, when the servers get their first production usage and failure rates are higher.

✦ Network (WAN and LAN): Although network traffic is more difficult to track, coordinate with the network group to analyze traffic among affected segments before rollout and immediately after. These statistics can be valuable later in deployment, or for diagnosing network bottlenecks.

✦ Clients: The Exchange Server technical resource kit provides useful tools for tracking user messaging. Several performance counters are available to gather useful information on the number of client logons, protocol usage, and so on.

Performance

You should monitor two areas of performance:

✦ Back end: Monitoring back-end performance can provide useful statistics about the accuracy of simulation testing, pilot implementation, and bottleneck determination. Exchange and Server versions of Windows 2000 provide
all the necessary tools to gauge many aspects of intraserver, interserver, and intersite performance. You can use the monitoring at first to determine problems, but it will also be useful later, in capacity planning.

✦ **Client:** The system is designed for end-user communication. Several performance monitor counters allow the administrator to determine back-end response time to user requests. You can also simply survey users. Other information can be gleaned from end-user operating systems, especially if you are using Windows 2000 Professional.

**Morning-after user surveys**

Survey end-users immediately after rollout. These surveys provide statistics to compare to the project’s goals, to gauge improvements and the success of the project. These surveys can also indicate problems with rollout procedures or in the system itself (for example, sluggish connector transmission).

**Modification**

In larger projects, the pilot and development phases rarely cover every possible angle; expect to make modifications.

Modification often causes changes to time lines and rollout schedules, but proceed with caution: It is often better to adjust the resources to fit an existing schedule than to reschedule a departmental rollout. If you have to reschedule a department, try to adjust that department only, or as few departments as possible, instead of rescheduling the entire project. For example, move two departments from their current slot in the project schedule all the way to the end, instead of keeping them in the same order and delaying all the other departments. Moving all dates is much harder and can lead to confusion and frustration. Inform the affected departments of any changes and the reasons for them.

**Transition to a steady state**

Often, the staff that is ultimately responsible for supporting the Exchange system is not the same staff that designed and implemented the project. In that case, include the lead support engineer as a member of the permanent support staff.

Once a departmental rollout is complete and stabilized, it is time to transfer support and administration to permanent staff. This transition is often logistically difficult, especially since many business units want to stick with the most knowledgeable team. But with sufficient effort and documentation, you can make this crucial transition smoothly.
Transition is easier if a large number of people in the deployment group are permanent support staff members. However, subsequent rollouts and the overall project usually benefit if a comparatively small number of support personnel are involved in the deployment group. Undue attrition of the deployment group during transition will adversely affect the remaining rollouts. So you have to find a balance in your organization between having enough permanent support staff people in your deployment to achieve continuity while ensuring you do not have so many that subsequent deployments become more difficult.

A transition is successful if the deployment team can transfer support calls to permanent support staff.

**Administration**

Depending on your administrative model, it is possible that not much administrative authority will be transferred from the deployment team to the support or administrative staff before the implementation is complete. Even if administrative support cannot pass to the permanent staff, you can still transfer knowledge and establish a semi-permanent staff for the administrative role.

Since architects and deployment teams accumulate very specific knowledge of a system’s hardware, parameter settings, idiosyncrasies, and other proprietary information, many organizations leave them with administrative responsibilities long after deployment is done. In fact, it is often practical for several deployment group members to become permanent administrative staff, to ensure continuity of their knowledge.

**Using milestones**

Milestones are markers of success according to predetermined criteria. Set milestones and use them as stop points for peer reviewing. When done properly, milestones can formally track and document a previous phase.

Set milestones at logical locations in the implementation plan—for example, at the end of each phase.

After implementation, compile all milestone reviews to form an executive-summary review of the project. This document is normally required for formal closure of the project.
Areas of Special Attention

When building your plan, special attention is needed in certain areas.

Topology and design

Even if you are familiar with earlier versions of Exchange and NT, give special attention to the significant changes in concept for physical and logical topology and design; these can reflect the way your company is organized, or even act as a catalyst for change.

Topological and design issues are covered throughout this book, but especially in Chapter 23.

Exchange on the Internet

Most organizations that implement Exchange will do so with significant interest in the capabilities of Exchange on the Internet. Many aspects of your plan will probably be affected by this interest.

The dynamics of Exchange on the Internet are consolidated and explained in Chapter 24 and covered in chapters throughout the book.

Active Directory Service and Exchange 2000

Your plan will either hinge on Windows 2000 or will need to become a joint Windows 2000 and Exchange 2000 plan. You cannot have Exchange 2000 without some Windows 2000. Your plan will have to embrace issues involving both products:

- Extending the Schema
- Domain controllers and catalog servers
- Address Book

For more information about Active Directory Service, see Chapter 6.

Exchange 5.x integration and migration

If you are doing a migration or coexistence with Exchange 5.x, focus your plan on the following issues:

- Mailboxes and accounts
- Active Directory
- Domains versus sites
- Groups and distribution lists
For more information about Exchange 5.x migration, see Chapter 18.

Migration wizards

If you are migrating from other legacy messaging systems, your plan will have to address migration wizards.

For more information on using migration wizards, refer to Chapter 18.

Summary

Years ago, when I was taking a class in woodworking, I asked a local doctor for help. He asked to see my plan. I said I was building the clock without a plan. He said, “Draw it for me. If you can’t draw it, you can’t build it.” Similarly, if you can’t—or don’t—plan it, you won’t build it.

The steps and components for building a successful plan are not hard to learn. This chapter explains one model. Once you have your plan, you can modify it and iterate it based on your successes and failures, to ensure that it continues to fit your organization and the evolution of Exchange. Of course, maintaining links to the business community and delivering real business utility is the raison d’être of your project.

Remember, a successful project is rarely a surprise. It is usually the result of considerable planning, deft execution, and real effort.
Migration and Coexistence

Because e-mail has been around for some time, it is doubtful that many companies or institutions are installing Exchange 2000 as their first messaging and collaboration application; most people are already using another system when they decide to move forward with Exchange 2000. This chapter prepares you to coexist with your existing messaging platform and introduces tools and techniques for migrating from some popular environments to Exchange 2000.

Whether you will be migrating to Exchange 2000 or simply coexisting with it, read this chapter for a solid understanding of how Exchange integrates with other messaging platforms. The chapter examines the tools available for connecting to other systems and details the systems requirements for supporting those connections. This information will help you build a supportable infrastructure to rely on while you coexist with your existing system.

This chapter also examines the tools that can help you migrate your system to Exchange 2000—tools for migrating directory information, a variety of shared and personal data types, applications, and more.

Earlier Versions of Exchange

Many of you have an existing platform of Windows NT 4.0 and Exchange 5.5 in place and are probably very interested in finding out what it will take to get to Exchange 2000. Exchange 2000 architecture is radically different and improved from that of previous versions of Exchange. Because of the architectural changes and the reliance on Windows 2000, it is not possible to upgrade from Exchange 5.5 to Exchange 2000 in the same manner as previous versions, when a quick upgrade on top of
the existing operating system was possible with little preparation or planning. Before you consider installing Exchange 2000 into the production environment, you need to upgrade the operating system to Windows 2000, and you should plan and prepare thoroughly. You should also do a pilot to test your design and migration plans.

For more information on project planning for Exchange 2000, see Chapter 17. For more information on Active Directory, see Chapter 6.

Checking your existing environment

The first thing that needs to go into place for an Exchange 2000 implementation is a rock-solid Windows 2000 Active Directory deployment. If you have not implemented a detailed Active Directory Design specification and your forest root domain, develop and implement that infrastructure first. Your Active Directory infrastructure will be the backbone of your Exchange system, so it must be efficient, reliable, and designed with Exchange in mind.

Windows 2000 Active Directory

To install Exchange 2000, you must first prepare the Active Directory. Some of the steps in this process include creating special accounts, delegating rights to those accounts, revisiting frequency and placement of catalog servers, and perhaps running ForestPrep, the special installation program that extends the AD schema to include Exchange 2000 object classes and attributes.

For detailed information about preparing the Active Directory, see Chapter 20.

You should also create the service account for the Active Directory Connector (ADC). The ADC service account needs to have domain account rights in the domain where the ADC (described in detail in the “Directory coexistence and migration with Exchange 5.5 systems” section later in this chapter) is installed. If you have not run ForestPrep in your domain, the account you log in with to install the ADC must be a member of the Schema Admin and Enterprise Admin groups.

Verifying Exchange 5.5 system readiness

It is important to make certain that the existing Exchange 5.5 organization has been prepared for coexistence with Exchange 2000 servers. Consider operating system and version requirements as part of your migration plan.

Operating system and Exchange version requirements include the following:

- Any machine to be upgraded in place must be running on Windows 2000 (Server or Advanced Server) with Service Pack 1 installed and Exchange 5.5 with Service Pack 3 or higher.
Any server that Exchange 2000 is installed on must be a member of a Windows 2000 Active Directory domain.

One server in each site must be running Exchange 5.5 with at least Service Pack 3 in order to synchronize its directory with Exchange 2000.

If installing Exchange 2000 into an existing site, one server with Exchange 5.5 Service Pack 3 or higher is required in the site, for directory replication.

You may need to upgrade some of your systems in order to meet these requirements.

### Design for the ideal distribution of data and services

You should design the ideal environment and work to migrate to that ideal. Take inventory of your systems and see if you can optimize the way your mailboxes, connectors, public folders, and applications are distributed. There are many new functions and features that will affect your placement of Exchange components and services, including the following:

- Your replication topology is changing. Now your Windows 2000 and Exchange 2000 replication mechanism is the same, removing the need for separate Exchange directory topology and Exchange bridgehead servers (you will instead have AD bridgehead servers).
- New services such as real-time collaboration and conferencing are available with Exchange 2000.
- The underlying Information Store architecture has been improved drastically to accommodate new ways of using and managing your Exchange environment. This improvement includes multiple private and public databases, storage groups, and more.
- Exchange 2000 can be configured in a front-end/back-end configuration, separating Exchange functions from protocol functions.
- Improved capabilities such as OWA scalability or the enriched application development environment might encourage you to use Exchange in a different way.

To design the end state, start with a blank sheet of paper. Set aside thoughts of your Exchange 5.5 environment for a while, and figure out where accounts, data, applications, and services are best distributed in a brand-new Exchange 2000 environment.

In many cases, you will find that server consolidation is possible, possibly saving money on hardware and systems maintenance. There are many ways to migrate Exchange 5.5 to Exchange 2000. In order to end up with an efficient and manageable Exchange 2000 environment, take advantage of the design and migration process.
Introducing Exchange 2000 into your existing environment

There are two methods for upgrading an Exchange 5.5 site. The first method involves upgrading the servers in place, installing Exchange 2000 on the existing Exchange 5.5 machines. Existing servers are upgraded one at a time to Exchange 2000, migrating the mailboxes, connectors, and public folder data in the process. If there are multiple servers in your Exchange organization, it is important to upgrade them in the proper order.

You may also install a new Exchange 2000 server into the existing Exchange 5.5 site and begin moving the mailboxes, connectors, and public folders to it. As Exchange 5.5 mailboxes and other components are moved, the 5.5 servers can be safely removed from the site. This is a more conservative approach than upgrading in place, and risk is minimal.

In spite of the additional risk involved, in-place upgrades can sometimes succeed. There are no exact formulas for deciding if an in-place upgrade is right for your situation. In general, the simpler the environment, the better when it comes to directly upgrading machines.

Consider the in-place upgrade if you have the following conditions:

- Single Exchange server
- Relatively small amount of users (<500)
- Limited amount of overall data (<25GB per server)
- Limited number of Exchange sites
- Connectors that are supported in Exchange 2000
- No large multi-server Exchange sites
- No budget for new equipment
- A penchant for excitement

In an environment that is complex or involves a lot of data, you should probably install a fresh Exchange 2000 server and migrate components individually. This technique enables you to limit the risk associated with an in-place upgrade, where you have to migrate all components at the same time. It also enables you to do your migration in manageable phases.

Consider installing new Exchange machines into the existing organization if you have the following conditions:

- Multi-site Exchange 5.5 organization
- Many servers in the organization (single-site or multi-site)
Crucial need for systems availability
Large amounts of users or data
Preference for a more conservative approach to systems upgrades
Sufficient budget to mitigate risk

A component-level look at upgrading

User mailboxes can be upgraded by upgrading the mailbox server or by moving them to an Exchange 2000 server in the same site. Mailboxes cannot be moved between sites using the move-mailbox method.

Distribution Lists are migrated to Exchange 2000 as mail-enabled Universal Groups. Universal groups can be used only in Native mode in Windows 2000. If the Exchange 5.5 Distribution Lists are required during coexistence, create a Windows 2000 Native mode domain to host them. Sometimes, a dedicated administrative AD root domain in native mode is used for this purpose.

Most connectors can be moved to Exchange 2000 by directly upgrading a server or by migrating to new connectors on a freshly installed Exchange 2000 server. You can configure new Exchange 2000 connectors with higher costs for testing — before you begin using them as your primary transport. When you are confident about their configuration and reliability, simply reduce the cost designation to put them into production.

SNADS and PROFS connectors, custom-built EDK, and third-party connectors cannot upgrade directly. If these connectors are required in your environment, use a 5.5 server to maintain them until suitable Exchange 2000 connectors can be obtained or developed.

Public folders on Exchange 5.5 servers can be migrated in place by simply upgrading the server. Alternatively, public folders can be moved to an Exchange 2000 server by replication. Public folders that are being migrated via replication are first replicated to an Exchange 2000 server. Once folder replicas exist on the Exchange 2000 server, the folders can be re-homed to it. This procedure effectively migrates the folder to Exchange 2000, and the original folder instance can be removed. It may, however, make sense to leave a copy of the public folder on the server, if users are still there.

Directory coexistence and migration with Exchange 5.5 systems

Exchange 5.5 servers build their Global Address List (GAL) from the Exchange 5.5 Directory service. Exchange 2000 uses the Global Catalog servers for this function. Since Exchange 5.5 and Exchange 2000 use different source directories to build the
GAL, there needs to be a replication mechanism between the two address books. Two services support directory replication between Exchange 5.5 and Windows 2000: the Site Replication Service (SRS) and the Active Directory Connector (ADC).

The SRS is installed when a server is upgraded or joined to an existing Exchange 5.5 site, or when a directory replication bridgehead server is upgraded. It makes the Exchange server look like an Exchange 5.5 server when directory connections are being made from Exchange 5.5 systems. An SRS server has a copy of the Exchange 5.5 directory, even though it is running Exchange 2000. The SRS must be removed from all systems prior to switching to Native mode.

The ADC also helps to integrate an Exchange 5.5 directory with a Windows 2000 Active directory. This integration helps create and maintain a unified GAL for Exchange 5.5 users and Exchange 2000 users in mixed mode. There are two versions of the ADC. One version ships with Windows 2000 and allows for synchronization of recipient objects in the Exchange 5.5 directory and the Windows 2000 AD.

The other version of the ADC ships with Exchange 2000. In addition to synchronizing the GAL, it also synchronizes configuration information such as site topology, server information, connector configuration, and more. Designed for mixed-mode operation of Exchange, the ADC allows much better integration between the AD and the 5.5 directory. It enables bidirectional sharing of connectors between Exchange 2000 and earlier versions and enables Exchange 5.5 management from the Exchange 2000 and Windows 2000 administrative MMC snap-ins.

Use the version of ADC that ships with Exchange 2000 instead of the one that ships with Windows 2000, especially when coexisting or migrating from Exchange 5.5 for maximum compatibility between the two directories.

**Connection agreements**

The ADC and SRS run as Windows 2000 services and control replication. The Connection Agreements (CAs) specify how the replication will occur. CAs define a single replication connection between the Windows 2000 AD and the Exchange 5.5 Directory. The CA is configured with information such as the local and remote server names, source and destination recipient containers, replication schedule, and connection security information. In the case of the SRS, a read-only CA is created automatically when the service is installed.

Manually created connections can be configured to work in only one direction, where changes in one directory are replicated to its partner directory, but not in the reverse direction. One-way connection agreements are used in a variety of scenarios. For example, you may want to continue administering your Exchange 5.5 Directory with the Exchange 5.5 administrative tools instead of moving wholesale to MMC-based administration. In this scenario, you create a one-way connection from each of your Exchange 5.5 sites to the AD and create a single-connection agreement to one of your Exchange 5.5 sites to replicate Windows 2000 accounts. Figure 18-1 shows an example of this setup.
A one-way agreement can populate a test environment as well. For example, suppose you want to create a test Win2K environment and would like to use your existing user data for testing. Simply create your test Windows 2000 AD and create a one-way connection agreement with your production Exchange 5.5 environment. Once the replication has completed, your test AD will contain all of the accounts and attributes that exist in your Exchange 5.5 organization. At that point, the ADC can be removed.

CAs can also be created to replicate changes in both directions, creating full synchronization. When two-way CA is created, it replicates changes from the Exchange Directory service to the Windows 2000 AD and from the AD to the Exchange Directory. Whether to use one-way or two-way connections depends on a number of factors, such as the existing messaging topology and your desired administrative model during the coexistence phase. For example, if you wish to use MMC-based
tools to administer your AD and Exchange 5.5 Directory during coexistence, you should use a two-way connection agreement, so that changes in the AD are replicated to your Exchange 5.5 organization. With this configuration, you can administer Exchange 5.5 mailboxes by manipulating their AD counterparts and replicating those changes through the ADC.

CAs can be created as primary or secondary agreements. A primary agreement replicates all object changes in a designated container to its destination container—including creating or deleting accounts or mailboxes or changing the attributes of those objects. Secondary agreements replicate only the attribute changes on existing recipient objects in the partner directory.

The ADC supports multiple CAs. Configuring multiple CAs is usually necessary in larger environments and allows for a lot of administrative flexibility configuring your enterprise directory replication topology.

**ADC prerequisites**

The Exchange 2000 Active Directory Connector requires a Windows 2000 Active Directory. The AD and a DNS server must be available and accessible at the time of install. Once the ADC is installed, Exchange 5.5 server with Service Pack 3 or higher installed can create a connection agreement between the systems.

**Installing and configuring the ADC**

To install the ADC, use the Exchange 2000 installation media. Begin by using the CD auto-run feature, or by starting Setup from the \ADC\I386 directory on the CD-ROM. After the welcome screen, you are presented with a component selection screen, as shown in Figure 18-2. Here you can choose to install the Active Directory Connector Service component, the Active Directory Connector Management components, or both. The service component is the actual connector, and the management component is the ADC administration MMC tool.
Once you’ve chosen the components to install, choose an installation folder location. You are then asked for the ADC service account, as shown in Figure 18-3.

![Figure 18-3: Specifying the ADC service account](image)

The service account should have been created already. It is usually an account that is used only for running the ADC service and needs to have administrative rights in the local Windows 2000 domain.

After you have specified the service account, the installation begins copying files and performing other system modifications. If your AD has been prepared with ForestPrep, the ADC installation should only take a few minutes to complete.

If your AD has not been prepared with ForestPrep, installing ADC could take a long time and will re-replicate the entire directory.

When the installation is done, click on Finish. You are now able to create a connection agreement.

**Creating connection agreements**

The ADC and connection agreements can be administered from the Active Directory Connector Manager, located in the Administrative Tools program group. From the Action menu, choose New, and choose Recipient Connection Agreement, as shown in Figure 18-4, to create a new CA for synchronizing Exchange 5.5 mailboxes and Windows 2000 accounts.
In the General tab, decide whether the CA is one-way or two-way. If one-way, decide which directory the information will replicate. If you want only Exchange mailboxes to appear in the AD, choose From Exchange to Windows. If you want only your Windows 2000 AD accounts to be added to the Exchange 5.5 GAL, choose From Windows to Exchange. If replication should go both ways, choose Two-Way. Figure 18-5 shows the general tab of a CA.
If you create multiple bidirectional CAs to the same Exchange 5.5 organization, you can end up with duplicate mailboxes, as Exchange 5.5 attempts to create the same mailbox in multiple containers or on multiple servers.

The Connections tab contains the bridgehead server and authentication information for connecting to the two systems. In the Windows Server Information section, specify the Windows 2000 bridgehead server for this agreement and an account that has permissions to the AD. In the Exchange Server Information section, identify the Exchange bridgehead server for this agreement and an account that has administrative rights to connect to it.

So far, you have configured the replication direction and the connections to each system. Next, choose the From tabs—From Exchange and From Windows. Here you will determine which types of objects are replicated and which containers are used for the source and destination directories, as shown in Figure 18-6.

Once you have chosen the containers and objects you want to replicate, set a replication schedule in the Schedule tab of the agreement. Just as in Exchange 5.5 directory replication connectors, you must base your replication schedule on the
frequency and amount of changes occurring in each directory, your network traffic patterns, the urgency of replicating the information, and time-zone adjustments. You will probably want to alter the default schedule to suit your needs.

You have now configured the critical information for your Connection Agreement. Choose the Deletion tab to set object-deletion rules for Exchange and Windows. Choose the Advanced tab to configure the agreement as primary/secondary or as an Inter-Organizational Connection Agreement. In the Advanced tab, you can also configure the initial flow of directory information in a two-way agreement, and some Windows 2000 object creation rules, as shown in Figure 18-7.

![Figure 18-7: Configuring advanced properties of a connection agreement](image)

### Mixed Mode versus Native Mode

After you configure the ADC, create Connection Agreements, and start moving objects and services to Exchange 2000, several tasks remain before you can switch to native mode:

- **Move all residual first-server items to Exchange 2000**, including responsibilities such as the organizational forms, Schedule Plus Free Busy, offline address book generation, Exchange 5.5 GWART, and system folders.

- **Delete all Exchange 5.5 servers from the organization**, using the Exchange 5.5 administration tool. The server cannot be deleted if it is on the network and running Exchange. The SRS will replicate deletion of these servers to the AD.
Delete the SRS and connection agreements to Exchange 5.5 from the Exchange system manager.

Once you have completed these tasks, switching to native mode is easy. Open the system manager, and choose the properties of the organization object. In the General tab of the organization properties, click the Change Mode button to switch your organization to native mode, as shown in Figure 18-8.

Figure 18-8: Switching to native mode

Now that we’ve switched to native mode, we can start taking advantage of some of the Exchange 2000 features available only in pure Exchange 2000 networks, such as changing routing group and administrative group topology and moving mailboxes between administrative groups. Other changes like complete migration away from use of NETBIOS to gain additional security on your network may also require a pure Windows 2000/Exchange 2000 environment before they can be accomplished, but are important goals of your project.

Coexistence and Migration from Other Systems

Many companies install Exchange in environments where a messaging system is already in place. Others integrate messaging systems due to a corporate merger or acquisition. Others have technical or organizational reasons for wanting multiple messaging systems. Coexistence with most mainstream messaging systems is possible with Exchange 2000. Migration is made possible by a variety of tools available from Microsoft and third-party vendors.
Support for coexistence is achieved in part through the built-in Exchange connectors that ship with Exchange 2000. Out of the box, Exchange 2000 supports connections to the following systems:

✦ Lotus Notes
✦ Lotus cc:Mail
✦ Novell GroupWise
✦ Microsoft Mail
✦ SMTP-based systems
✦ X.400-compliant systems

**Installing a connector**

Installing a connector is simple. It can be done during the initial installation of Exchange or later. In a typical installation, the SMTP and X.400 connectors are installed automatically. If you need to add a connector, start the Exchange setup and follow the prompts until you get to the Component Selection screen.

In the Component Selection screen, choose the Change action for Microsoft Exchange 2000 and Exchange Messaging and Collaboration Services, as shown in Figure 18-9. Then, choose the Install action for the connector(s) you wish to install, and click Next until you’ve started the install.

![Figure 18-9: Installing connector software](image-url)
Since system changes and the amount of code associated with connectors are limited, installing a connector takes only a few minutes. Some of the important system changes occur during the installation of a connector:

✦ Connector software is copied to the Exchange installation path, and directory changes are made. Executables such as CCMC.exe (for the connector for cc:Mail) are copied to the Exchsrvr\Bin directory.

✦ Registry keys such as LME-NOTES (Connector for Lotus Notes) and MSEXCHANGECCMC (Connector for Lotus cc:Mail) are added to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services to control the corresponding Windows 2000 service that runs the connector. Under each service key, you will see a Parameters key. Generally, any connector configuration that cannot be performed through the System Manager application is added to the Parameters key under the connector service key.

✦ A proxy address generator .dll file is added to the %InstallPath%\Exchsrvr\Address\ shared directory. For example, when you install the Connector for Novell GroupWise, a \Gwise\I386 directory is added to the Address share. The GWXPXGEN.dll file is placed there. This .dll file is used to generate a GroupWise-formatted e-mail address for a Windows 2000 account or contact in the Active Directory.

✦ A new address type is added to the default recipient policy, enabling you to configure the system to create and associate the new address type for AD accounts in the Exchange organization. When an Exchange user sends a message through a non-exchange connector, the address is given in the native format of the foreign messaging system.

✦ New objects appear in the Exchange organization in the Connections container. In this container you configure your new connector to communicate with another messaging system. You can find the Connections container in the Exchange System Manager snap-in.

Using the Migration Wizard

The Migration Wizard — installed when you install Exchange, and accessible from the Microsoft Exchange program group — can assist migrations from a variety of systems, including cc:Mail, Lotus Notes, MS Mail, IMAP4, GroupWise, and others. In addition to migrating messaging and collaborative types of data, the Wizard is extremely useful for migrating directory data to Windows 2000. The Wizard can be run from an Exchange server, or it can be installed on a separate workstation.

During migration, the Wizard generates primary migration files, secondary migration files, and a packing list file. The primary migration file is a comma-separated value (CSV) file that contains directory information, mail-message headers, and personal address lists. The secondary migration file contains other data such as attachments, message bodies, binary data, and any text with carriage returns.
Special pointers link secondary migration file data with corresponding entries in the primary file. The packing list file details the contents of the primary and secondary migration files and specifies the code page of the information.

The Migration Wizard can be used for a one-step or two-step migration. A one-step migration is a direct migration from the native system to Exchange. Information is copied from the cc: Mail environment and added directly to the Exchange environment in a single process. Mail-enabled Windows 2000 accounts are added on the fly (if they do not exist already) and populated with mailbox data converted to an Exchange format. This type of migration works well if you have a relatively clean directory that doesn’t need to be changed during the migration and if you have a small amount of users to be migrated.

The two-step migration enables you to copy information from the native system and view it or edit the migration files before importing it into Exchange. Once you have reviewed and/or edited the migration files to your satisfaction, run the migration utility again to import the information into Exchange. This type of migration works well if you need to clean up directory information before it is imported to Exchange. For example, you might want to change the user’s display name to conform to your Exchange 2000 naming conventions.

Before migrating user data, have users clean up their mailboxes and make sure you have run database utilities to clean up and ensure the integrity of the information. For example, run utilities such as Reclaim and Analyze on a cc: Mail post office prior to migrating it to Exchange. The Migration Wizard ignores some inconsistencies, others generate errors, and more severe database issues stop the migration process.

Ensure that your Exchange infrastructure is reliable and ready for production; perform monitoring, backups, virus protection, and all other production support and maintenance mechanisms. Also, to ease the transition to the new system, ensure that users are trained on its basics.

Carefully consider which information you want to migrate. Some corporations choose to leave legacy data such as mailing or distribution lists, bulletin boards, and even user data on legacy systems and do not migrate them to Exchange. Others have deployed Exchange without migrating anything, deeming the existing data and structure not worth the effort of cleanup, migration, and coexistence.

Some companies migrate data to personal folders instead of directly to the Exchange server. The Wizard provides this option so you can migrate the data to an Outlook-compatible format without actually having to move it into the Information Store. The advantage of this approach is that users still have access to their historical data, but you are not dragging it into the new system. In some cases, organizations find it easier to enforce new mailbox size limits if they start with empty mailboxes in the new system. The disadvantage of the approach is that you have to support personal folders, a task with its own challenges.
Depending on which system you migrate from, you have options for choosing which types of data are migrated and how. Carefully review the options that you have for your system, and run trial migrations using real data. Testing the migration process will accomplish several tasks:

- Familiarizing you with the individual steps of the migration process and their logical sequence
- Familiarizing you with the tools used for the migration
- Verifying that your specific migration process will work as expected
- Indicating how much time will be required for the migration

The actual migration process of directory information depends on whether accounts or contacts already exist for the migrated users. The Migration Wizard merges the source directory information into an enabled Windows 2000 account, if one exists already. If a contact exists for a migrating user, the contact information and the source directory information are merged into a newly created Windows 2000 account, and the contact is removed. If a disabled Windows 2000 account exists, the account is enabled and the source directory information merged into it. Of course, if no Windows 2000 account or contact exists prior to the migration, one is created.

**Active Directory Account Cleanup Wizard**

Some migration scenarios result in multiple accounts for the same user. For example, suppose you set up replication with another directory and choose to place the synchronized accounts in an Organizational Unit (OU) created just for that purpose. You decided to do this because you wanted to separate these users from your standard users. Now you want to migrate the users, and you would like to migrate them to another container in the Active Directory. After the migration, you may end up with one user object created from the synchronization and another created during the migration. The Active Directory Account Cleanup Wizard helps you reconcile these accounts into a single active account, merging the information from multiple sources.

**cc: Mail coexistence and migration**

Lotus cc: Mail is a popular shared file–based messaging system. It supports both LAN-connected and mobile users and uses shared folders (bulletin boards) and forms. The cc: Mail system architecture consists of a Post Office located on the network. The Post Office (PO) is a passive database made up of several directories that contain directory information, message headers, message content, attachments, bulletin board data, and other information. The messaging client application reads and writes to the PO to send and retrieve data. Administrative tools and agents accomplish message transfer, directory updates, and other administrative functions by reading and writing to the PO.
cc: Mail is being replaced with other systems for a number of reasons. Companies are migrating to Exchange for the following reasons:

♦ The DB6 database is not Y2K-compliant.
♦ IBM/Lotus has announced that all cc: Mail sales, development and live support will be discontinued by the end of the year 2001.
♦ cc: Mail does not scale well to large organizations and lacks the collaborative features and management flexibility of most modern messaging and collaboration applications.
♦ cc: Mail database recovery scenarios often involve some loss of data.
♦ A merger or acquisition with a company running Exchange may force coexistence or migration.
♦ A current SMTP gateway may need to be replaced with something more robust and stable.

Exchange 2000 integrates very well with cc: Mail. Coexistence and migration can both be done very methodically and precisely, and cc: Mail migrations are very well documented. Coexistence is achieved through the cc: Mail connector. The connector service transfers user messages between Exchange and cc: Mail and enables Exchange to participate in cc: Mail Automatic Directory Exchange (ADE) for directory synchronization.

The connector works by connecting to a single cc: Mail post office and transferring messages and directory updates to that PO. Once in the cc: Mail PO, the native cc: Mail message routing and directory replication topology routes the information to the appropriate Post Office(s). The cc: Mail connector runs as a Windows 2000 service. It uses the cc: Mail program files Import.exe and Export.exe to connect to the cc: Mail PO.

At first glance, some are tempted to connect the Exchange connector to a hub PO in the existing cc: Mail environment. This technique works in some cases, but the best practice is to connect the cc: Mail connector to a dedicated connector PO created just for this purpose. Using a dedicated connector PO creates less potential risk by enabling you to isolate the connector environment from the production cc: Mail systems. For example, in cases of PO corruption, you can run database utilities on the connector PO without affecting production systems. Since the PO has no users, you can very quickly recover it from backup without having to worry about user mailbox data. If you are in a large environment, you may have thousands of messages utilizing this connector every day. You can install the connector PO on the Exchange server or on a dedicated machine. From a performance perspective, the important thing is to give the Exchange server robust access to the connector PO. This translates to fast disk I/O on a local exchange server or a fast network connection if the PO is on another machine. Figure 18-10 shows the connector architecture for typical cc: Mail coexistence scenarios.
You can create multiple connectors to your cc: Mail systems, but unlike the MS Mail connector, this connector enables connection to a single PO only. Also, you can create only one connector per Exchange server. You might consider using multiple cc: Mail connectors if you are running in a large environment with multiple cc: Mail hubs. As with all connectors, configuring cost and scope controls message flow in an organization. You can use multiple connectors to form a parallel hub structure with cc: Mail for more efficient message transport in a regional-hub topology. By using multiple connectors, you can also create redundant paths to cc: Mail. Since the cc: Mail message routing architecture does not dynamically update routing information, this approach will not provide redundancy from cc: Mail to Exchange. You should create only a single cc: Mail connector for directory synchronization and use all other connectors solely for message transport.

Figure 18-10: Examining the cc: Mail connector architecture
cc: Mail connector requirements
The cc: Mail connector uses cc: Mail software to transfer directory and user messages between Exchange and cc: Mail. The software you will use depends on the version of your connector PO database. Before you connect Exchange 2000 to your existing cc: Mail system, you should check your existing cc: Mail system version. In order for Exchange to connect to your cc: Mail PO, you must use database Version 6.x or 8.x. If your system does not meet these requirements, upgrade it before connecting to Exchange.

To check the database version, go to the \ccdata directory and run analyze.exe or analyze32.exe. You can run these tools without a password, and they will report the Postoffice database version, which is what you are interested in.

After identifying the database version, make sure you have the correct version of the Import/Export software for your PO database. Version numbers for the Import and Export programs are different from the database version, so make sure you have the correct software for your PO. Table 18-1 shows the correct Import/Export versions of the software to connect Exchange to your cc: Mail PO.

<table>
<thead>
<tr>
<th>Post Office Database Version</th>
<th>IMPORT/EXPORT Versions Acceptable for Exchange Connectivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB6</td>
<td>IMPORT 5.15/Export 5.14</td>
</tr>
<tr>
<td>DB8</td>
<td>IMPORT 6.0/Export 6.0</td>
</tr>
<tr>
<td>DB8</td>
<td>IMPORT 8.0/Export 8.0</td>
</tr>
<tr>
<td>DB8.1</td>
<td>IMPORT 8.5/Export 8.5</td>
</tr>
</tbody>
</table>

The Import and Export software can be located on the cc: Mail distribution media or obtained from Lotus support.

If you are using Import/Export Versions 6.x or 8.x, obtain the language resource file for use with these programs. Version 6.x uses a file called ie.ri, and Version 8.x uses a file called impexp.ri.

Configuring a connection to cc: Mail
Before connecting to cc: Mail, make sure you have done the following:

✦ Installed the connector software from the Exchange installation media.
✦ Identified the PO that the connector will use. In most cases, you should install a dedicated connector PO; it can be installed on the Exchange server or on a separate machine, depending on how much performance you need.
Configured (if directory synchronization is to be configured) and tested cc: Mail directory propagation to the connector PO. The cc: Mail Administrator should be able to assist with designing and configuring a suitable ADE topology. In many environments the connector PO is configured as a broadcast PO, so that Exchange directory objects can be propagated to other cc: Mail POs.

Obtained the correct version of the Import and Export programs, and copied the software to a system directory on the Exchange server. If you place the Import/Export files in a non-system directory, add the folder path to the system environment variables. If using database Version 8.x, copy the language resource file (ie.ri/impexp.ri) to the same directory.

Verified that you can make a UNC connection from the Exchange server and the connector PO. Many cc:Mail environments run on Novell File servers. In this case, you may have to run Services for NetWare and the IPX/SPX protocol on your Exchange server. See your Windows 2000 documentation for instructions on how to configure network connectivity to Novell file servers.

Once you have installed the connector and copied the correct software to the Exchange server, open the Exchange system manager and expand the Connectors container. In the Connectors container, you will see the cc: Mail connector along with any other connectors you have installed. In the properties of the connector, you will see the Post Office tab, as shown in Figure 18-11. In this screen, configure the basic information required to connect to the cc: Mail PO.

![Figure 18-11: Configuring a connection to a cc: Mail PO](image)

In the Post Office tab, name the connector and provide an administrative mailbox for the connector. The Administrator mailbox receives all administrative messages.
for the connector, such as message conversion problems or Non-Delivery Reports (NDRs). This can be a group account or a generic mailbox, such as Postmaster.

Next, provide the connector language, connector PO UNC path, and connection account information. The PO path must be a UNC name in the format of the server where the PO resides. You can also use the Browse button to browse to the PO path instead of typing the information manually.

The Connect As information specifies the account and password used to connect to the file share where the PO resides. The account must have rights to connect to the Post Office share.

The Allow ADE to propagate synchronized entries option permits Exchange to participate in ADE exchange with the cc: Mail environment. It is on by default, but should be used only on a single cc: Mail connector in the organization. When configuring additional connectors, clear the check box. The check box has no effect if the PO is not configured for ADE propagation.

So far, you have created a basic connection to the cc: Mail PO. To specify the scope and cost of the connector and addresses it will send to, switch to the Address Space tab. In this screen, choose a cost (1–100), scope, and one or more valid addresses that this connector will send to. The * and ? wildcards can specify multiple recipient addresses. Configuring the address space effectively adds this connector to the Exchange organization’s message-routing topology. Figure 18-12 shows how a connector configured to send to all users with the “Username at Corp” address format. All Exchange users can use this connector, since it is scoped to the entire Exchange organization. If you want to be able to send to any cc: Mail address by using this connector, simply enter * for the address space.

![Figure 18-12: Configuring the cc: Mail connector address space, cost, and scope](image-url)
If you are going to synchronize the Active Directory with cc: Mail, you must configure the Import Container and Export Container tabs. In the Import Container tab, specify the Windows 2000 container where cc: Mail directory additions and changes are imported. In situations of long-term coexistence, you may want to create an Organizational Unit (OU) for this purpose. In Figure 18-13 shows a container (called cc: Mail recipients) created to hold the imported accounts. cc: Mail users are added as Windows 2000 contacts by default.

![Figure 18-13: Configure the Import container for cc: Mail directory updates and create an OU.](image)

The Export Container tab enables you to specify one or more Windows 2000 containers for directory replication to cc: Mail. Add all containers that contain accounts that cc: Mail users need access to. In large or complex environments, many containers will need to be added to this list for all mail-enabled accounts to be available to cc: Mail users. Any time a new container is added to the AD that you need to replicate, you must add it manually.

Once you have configured the import and export containers, go to the Advanced tab of the connector to set a directory replication schedule; this schedule is for replicating changes from the AD to cc: Mail. The cc: Mail updates to Exchange occur according to the schedule set in cc: Mail. Make sure your connector directory replication schedule makes sense in the context of the greater ADE replication topology.

To enable Exchange accounts to be replicated to cc: Mail, you must enable the cc: Mail address type in the recipient update policy. The cc: Mail address type is not available until the connector is installed. To enable the cc: Mail address type, go to...
the System Manager and expand the Recipients container. In the Recipient Policies container, by opening a policy and going to the E-Mail Addresses tab, you can specify which types of addresses are automatically generated for the accounts that meet this policy’s criteria. In Figure 18-14, the cc: Mail address type is enabled on the default recipient policy.

Consider bulletin board replication strategies if you are planning a long-term coexistence scenario or if your migration process will take a while. You can configure coexistence in two ways, depending on whether you want Exchange or cc: Mail to have the master copy of the shared information.

To use cc: Mail as the master location, configure bulletin board propagation from the master bulletin board to a corresponding Exchange public folder. The address you will propagate to can be in the format `ExchPF at ConnectorPO`, where `ExchPF` is the directory name of the Exchange public folder and `ConnectorPO` is the name of the cc: Mail PO used by the cc: Mail connector. Exchange should post information to the master cc: Mail bulletin board by mailing to a contact created with the bulletin board address. As information is added to the bulletin board, it is propagated to Exchange via the cc: Mail bulletin board propagation mechanism. Note that this technique does not address deletions from either system.

If you want Exchange to house the master copy, create a directory entry in cc: Mail that points to the Exchange public folder. cc: Mail users can use this address to mail to the public folder. Next, configure a rule on the public folder to copy messages to a contact created with the bulletin board address. In this scenario, users post only to the Exchange public folder, and the data is copied to the corresponding cc: Mail bulletin board. This scenario does not propagate deletions either.
**cc: Mail migration tools**

Migrations from cc: Mail can be complex, because of the different types of information to be handled. You may want to migrate the following cc: Mail components:

- User messages
- Directory information
- Bulletin Board data
- Mailing List information
- Lotus Organizer calendar information
- Priv.ini private directory information

In order to migrate from cc: Mail, you may use several tools, depending on which components you intend to migrate, your migration methodology, and the configuration of your existing environment. Some of the tools you can use are listed in Table 18-2.

<table>
<thead>
<tr>
<th>cc: Mail Migration Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exchange Migration Wizard</td>
<td>Enables you to migrate user data, directory information, and bulletin board information from your cc: Mail environment into Exchange</td>
</tr>
<tr>
<td>Lotus cc:Mail Migration Directory Toolkit for Outlook 98 and Outlook 2000</td>
<td>Used to migrate cc: Mail directory information (including mailing lists and user private directory information) to the Exchange 2000/Outlook 2000 environment</td>
</tr>
<tr>
<td>cc:Mail Archive Importer (Ccmarsh.exe)</td>
<td>Used to convert cc: Mail archive files into Personal Folder (.pst) format</td>
</tr>
<tr>
<td>Outlook 2000 Import/Export Wizard</td>
<td>Used to import Lotus Organizer information into a user's Calendar</td>
</tr>
</tbody>
</table>

Your primary tool for migrating from cc: Mail is the Migration Wizard. Before running a migration, make the following preparations:

- Run a trial migration against PO data restored from backup. This will ensure that your migration process works and minimize risk from database corruption.
- Choose the machine you will migrate from — the Exchange server that you are migrating to, or a separate workstation. Make sure that the machine you run the migration from is robust and has fast access to Exchange and the cc: Mail PO. If possible, move the PO to the Exchange server itself and run it from there.
Prepare the migration machine. Whichever machine you choose to run the Migration Wizard on will need the cc: Mail Export.exe and corresponding language resource file somewhere in the system path. If you have been running the Connector for cc: Mail, these files are already available on the Exchange server.

Make sure users rename folders with default Outlook folder names. If user mailbox data is being migrated into the Information Store, users will want to rename personal folders named Tasks, Contacts, Calendar, and so on before the migration. If they do not, Outlook will have problems creating the default folders during the first time setup.

Make sure the database is cleaned up and backed up. Back up the PO and run cleanup utilities and regular maintenance to ensure database integrity. Back up the PO again immediately before the migration.

Make sure nobody is logged on to the PO, and make sure all routers, gateways, and maintenance utilities are turned off. The Migration Wizard should have uninterrupted, exclusive access to the PO during the migration.

If you have done your planning, preparation, and trial migration, the actual migration is fairly straightforward.

**Microsoft Mail coexistence and migration**

Before the release of Exchange 4.0, Microsoft’s premier messaging application was Microsoft Mail for PC Networks (MS Mail). The architecture of Microsoft Mail is very similar to that of cc: Mail. A passive, LAN-based Post Office database resides on a network share, and front-end applications read and write to the Post Office for user interaction, message transfer, directory transfer, and administration.

Because of its LAN-based, shared-directory architecture, Microsoft Mail suffers from many of the same problems as cc: Mail. It does not scale well to large organizations and does not offer the security and reliability expected for business-critical applications. In addition to architectural issues, support for MS Mail is limited and new development has ceased. For these reasons, most organizations currently using Microsoft Mail are migrating to Exchange.

In many environments, Exchange must coexist with an existing MS Mail application during migration. In order to enable its customers to upgrade to Exchange, Microsoft provides superb tools for both coexistence and migration. The backbone of any coexistence strategy involves the Connector for MS Mail, used for message transfer and directory synchronization between the MS Mail environment and Exchange. Exchange server supports a single connector per server.
The Connector for MS Mail consists of three basic components:

- **The Connector Post Office**: Sometimes called the Shadow PO, this component acts as a temporary storage area for data being transferred to and from the MS Mail environment. Located on the local Exchange computer in the \Program Files\Exchsrvr\Connect\msmcon directory, the Connector PO is very similar to a standard MS Mail PO. Unlike the MS Mail PO, it does not contain any recipient mailboxes. It is accessed by the Connector Interchange (described below) via the Maildat$ hidden share.

- **The Connector Interchange**: This component uses the Connector PO to transfer data from Exchange to the MS Mail environment. The Interchange component runs as a Windows 2000 service and retrieves and delivers data to the Connector PO according to a schedule that the Administrator chooses. The Interchange only utilizes the Connector PO and does not interact with other MS Mail Post Offices.

- **The Connector MTA**: Sometimes called PCMTA, this component is responsible for transferring messages between the Connector PO and other MS Mail Post Offices. It runs as a Windows 2000 service and can be used to connect to multiple MS Mail POs in a hub configuration. Since the Connector MTA communicates directly with MS Mail Post Offices, it can supplement or replace existing MS Mail MTAs, depending on your MS Mail topology.

Every Connector PO has a serial number of 22-28798. MS Mail Post Offices use this serial number to identify themselves to other POs. Because the number is not unique, you cannot use two Exchange Mail Connector POs as a backbone for Exchange. If you attempt to, the connector MTA will not be able to pass any mail and will produce an error in the event log. Even if it could be done, it should not be done. Because Exchange has more backbone features, it will always make more sense for Exchange to be the backbone for MS Mail.

When data is transferred between Exchange and MS Mail, each of the above three components is involved. When an Exchange user sends a message to a Microsoft Mail user, the message is routed to an Exchange server with the Connector for MS Mail. When the message arrives it is converted to MS Mail format, and the Connector Interchange service delivers the message to the outbound queue of the Connector PO. The message is retrieved from the Connector PO by the Connector MTA and is then routed within the MS Mail MTA topology to the destination PO where the recipient is located. Messages are routed from MS Mail to Exchange in the reverse order. Figure 18-15 illustrates the message flow between Exchange and MS Mail in a typical environment.
Preparing for the Connector for MS Mail

Before you connect Exchange to your MS Mail environment, you should do some discovery work. The three main areas to become familiar with are the Routing Topology, the Directory Synchronization Configuration, and the individual Post Office characteristics. Knowledge of these areas helps you determine how Exchange can best be integrated into the current messaging system. All of the information you need is available through the Admin.exe MS Mail administration program.

✦ **Routing Topology**: Review and understand the existing message transport systems topology. Make sure you know where all the MTAs are located on the network, what transport they use to connect to other post offices (LAN, ASYNC, X.25), and the protocols used. You will want this information when you configure the connector MTA.

✦ **Directory Synchronization Configuration**: Make sure you understand how Dirsync is configured in the MS Mail system. Make a note of where the Dirsync server resides, where the requestors are located, and what the schedule is. Understanding the current Dirsync configuration helps you design your directory synchronization strategy for Exchange and MS Mail.
**Post Office Characteristics:** Survey the existing Post Office databases to gather statistics about the number of users, Post Office size, PO version, mailbox size limits, regular maintenance performed, known issues, and so on. Compiling this information will prove to be an extremely valuable exercise in both coexistence and migration situations.

If it is feasible, clean up the existing environment before connecting Exchange. Microsoft provides diagnostic tools that can help ascertain the health of an MS Mail PO. In some cases, these tools can fix the MS Mail PO database. They are invasive, however, so make sure you back up the PO prior to performing any repairs. The tools for Versions 2.x and 3.x can be obtained from Microsoft, and the tools for 3.5 are available on Version 3.5 PO installation media. For more information on the PODIAG tools, search the TechNet Web site at http://support.microsoft.com/servicedesks/technet/.

The TechNet site also provides valuable information about how to troubleshoot Dirsync and message transport. Make sure your current MS Mail environment is functioning as well as possible before you connect Exchange to it. Address any known issues while you still have your MS Mail environment segregated.

**Connecting to Microsoft Mail for PC Networks**

Once you have performed your discovery and addressed outstanding MS Mail issues, you are ready to connect Exchange to MS Mail. To connect to the MS Mail environment in a coexistence situation, you must configure the Connector for MS Mail. If you have not installed the connector, do so now using the procedure described in the “Installing a connector” section of this chapter. Make sure that the server you install the connector on has LAN or dial-up access to the MS Mail PO database shared directory (often shared as MAILDATA) and is running a compatible protocol to the server where the PO share resides.

You can find the connector — essentially the same one as in earlier versions of Exchange — in the Connectors container in the Exchange System Manager snap-in. To configure the connection, highlight the connector and view the properties. The first screen you will see is the Interchange tab, as shown in Figure 18-16. Here you identify an administrative mailbox to receive administrative notifications, specify language and message-tracking preferences, and can choose the Maximize MS Mail 3.x Compatibility option. The Maximize option forces Exchange to convert any embedded OLE objects to a format that an MS Mail client can read. Enabling this option also roughly doubles the size of the converted OLE-embedded message going into MS Mail.
Once you have configured the Interchange properties, go to the Local Post Office tab and make a note of the values there. The Network value is configured by default as the name of your Exchange organization. The default values for the Postoffice and Sign-on Password fields are EXCHPO and EXCHADMIN, respectively. MS Mail POs use these values when you identify the connector PO in their directory, so note them for later reference.

If you change the Network or Postoffice values in this connector tab, you must also change those values in the appropriate account recipient policies in Exchange 2000 and regenerate those addresses.

If you switch to the Connections tab of the connector, you can configure connections to MS Mail POs—besides a default connection to the connector PO that you do not have to configure. In this tab, you can create LAN, modem, or indirect connections to an MS Mail PO by clicking the Create button. When you specify an indirect connection, you are configuring a connection to an MS Mail PO via another PO, as shown in Figure 18-17.

If you are configuring a LAN connection, you must specify a network URL for the PO, and a Connect As username and password. Once you have connected to the PO via the LAN, you can upload routing information about other POs that are known to the connected PO. You can upload the information by simply clicking on the “Update Routing” button in the properties of the LAN-connected PO.
Once you have configured your connections, go to the Connector MTAs tab, where you configure connector MTA services on the Exchange server and specify which MS Mail POs they serve for message transport, as shown in Figure 18-18. This procedure configures the message routing topology between Exchange and MS Mail. Each MTA service that you configure is created as a separate Windows 2000 service, so that each MTA can have its own configuration parameters for message polling frequency, message size limits, and so on.
So far, you have created basic connections to the MS Mail system. To specify the scope and cost of the connector and addresses it will send to, switch to the Address Space tab. In this screen, choose a cost (1–100), scope, and one or more valid addresses that this connector will send to. The * and ? wildcards can specify multiple recipient addresses. Configuring the address space effectively adds this connector to the Exchange organization’s message routing topology. If you want to be able to send to any MS Mail address by using this connector, you can simply enter * for the address space. If you have already defined connections to MS Mail POs, the address space information is automatically configured to include users at those PO addresses; so you may not need to configure anything here.

The MS Mail POs must be configured to recognize the Exchange Connector PO as an external PO, adding it to the MS Mail directory as a valid PO to interact with. To accomplish this, work with your MS Mail Administrator to use the Admin.exe MS Mail administration program. Depending on which version you have, the process may be slightly different, so consult your MS Mail documentation for directions. When you configure the External PO values in MS Mail, use the values you wrote down while viewing the Local Postoffice tab of the connector. Every MS Mail PO connected to Exchange must have the connector PO configured as an external PO in the Admin.exe program.

**Synchronizing the Active Directory with the MS Mail Global Address List**

After you have configured a connection to MS Mail, messages can be transferred between the two systems. At this point, however, the systems do not share directory addresses. If you are coexisting with MS Mail, synchronize the Exchange 2000 Global Address List (GAL) with the MS Mail GAL. Then Exchange and MS Mail users can send messages to either system by choosing addresses from their respective GAL.

Synchronizing the two systems means setting up Dirsync between Exchange and MS Mail. MS Mail Dirsync works as follows. In MS Mail, a Postoffice can play two roles in the Dirsync process: the Dirsync Server and the Dirsync Requestor. The relationship of the Dirsync Server to its requestors is one-to-many, so there is only one Dirsync Server per MS Mail Network. The Dirsync Server maintains the master list of addresses compiled from all participating Postoffices (requestors). Changes to the master GAL are periodically sent to all requestors, which then incorporate those changes to their local copy of the GAL. The process happens in three stages: Time 1, Time 2, and Time 3 (T1, T2, and T3).

- **T1:** The Requestor PO sends a message to the Dirsync server containing all local directory changes since the last Dirsync. The message is transported using the standard MS Mail PC MTA message transport mechanisms.
- **T2:** The Dirsync server compiles the GAL from all of the updates it has received and sends directory update messages to each of the requestors. It is important that all requestors send in their changes before T2, or their changes will not be propagated to other POs until the next cycle.
- **T3:** Each requestor PO rebuilds its local copy of the GAL by incorporating the update message sent from the Dirsync server during T2.
Exchange uses a Windows 2000 service called Microsoft Exchange Directory Synchronization (DXA.exe) to manage directory synchronization with an MS Mail network. You must decide whether to configure Exchange as a Dirsync Server or as a Dirsync requestor. There is no single way to fit Exchange into the existing Dirsync topology; instead, base your decision on the existing configuration of Dirsync, how well it works, and the impact of your decision on the migration path.

### Configuring Exchange as a Dirsync requestor

If everything runs smoothly in your current Dirsync process and you do not want to make any major changes to the Dirsync configuration, configure Exchange as a Dirsync requestor. This is a relatively easy configuration with the least impact on existing systems. When Exchange is configured as a Dirsync requestor, it sends incremental GAL changes to the Dirsync Server on the MS Mail Network. In this configuration, Exchange also receives updates to the MS Mail GAL and incorporates them into the Exchange GAL.

To create a Dirsync requestor in Exchange, use the System Manager snap-in. Highlight the Connectors container, choose New, and choose Dirsync Requestor from the Action menu. You will be prompted to name the Dirsync Requestor (for display purposes) and specify the Dirsync Server Postoffice from the list of connected MS Mail Postoffices. After you have identified the Dirsync Server, you will see the General property page of your new Exchange Dirsync requestor. This property page is shown in Figure 18-19.

![Figure 18-19: Configuring General properties of an Exchange Dirsync Requestor](image)
In the General property page, verify that the appropriate language and address types are chosen. By default, only the MS Mail address type will be synchronized during Dirsync. You can specify additional address types by choosing the appropriate check boxes.

In the Import Container tab, specify the Active Directory location in which MS Mail recipients are placed when they are replicated from the MS Mail GAL. Determine which type of AD object is created during synchronization (usually contacts).

In the Export Containers tab, specify all of the AD containers that you want to add to the MS Mail GAL, as shown in Figure 18-20.

![Figure 18-20: Export Containers tab of an Exchange Dirsync Requestor](image)

In the Schedule tab of the object’s properties, set the Dirsync schedule—that is, schedule for when the requestor sends in its updates to the Dirsync Server for all containers that are specified in the Export Containers tab. Send updates daily, before the T2 set by the MS Mail Dirsync Server.

In the Settings tab, specify a password for sending update messages to the Dirsync Server. You can request a full import or a full export for the next Dirsync cycle and choose options for template replication. Templates are similar to custom forms in Exchange and can be replicated if you want to. During the first successful Dirsync, a full import/export is automatically processed.
Configuring Exchange as the Dirsync Server

Some organizations choose to change their Dirsync topology, configuring Exchange as the Dirsync Server. As a Dirsync Server, Exchange can provide better reliability, easier management, and more flexibility. For example, in MS Mail, updates are sent only once a day; more frequent updates must be done manually. Exchange does not have this restriction, so you can configure the T2 schedule as often as you like, keeping tight synchronization with the MS Mail Postoffices.

The downside of configuring Exchange as the Dirsync Server is that you have to reconfigure all of your MS Mail Postoffices. If there are many Postoffices or no centralized management, this can mean a lot of work. However, if you plan on leaving users on MS Mail for a long time, you should probably move the Dirsync Server to Exchange.

To configure Exchange as the Dirsync Server, open the Exchange System Manager snap-in and navigate to the Connectors container. With the container highlighted, choose New and Dirsync Server from the Action menu. You arrive at the General property page, as shown in Figure 18-21.

First, specify the display name of the object in the Active Directory. Then, as in the MS Mail connector, specify an Administrator’s mailbox to receive administrative notifications. In troubleshooting situations, you may want to mark one or both of the check boxes for sending incoming and outgoing Dirsync messages to the administrative mailbox. Doing so will give you a chance to see the actual messages being sent to and from the requestor Postoffices.
In the Schedule tab, configure the T2 time of the Dirsync process — the time at which the Dirsync Server sends update messages to all the requestor Postoffices. Adjust the schedule according to how often changes should be replicated to the MS Mail network. In steady-state coexistence scenarios, you may want to run updates only daily, whereas active migration or other directory-dynamic situations may require more frequent synchronization. The Details tab is for administrative notes.

Once you have configured the Exchange Dirsync Server, add every MS Mail Postoffice participating in Dirsync as a remote Dirsync requestor. Setting up a remote Dirsync requestor is very similar to setting up an Exchange Dirsync requestor; it defines an MS Mail Postoffice as a requestor to the Exchange Dirsync server. From the System Manager snap-in, highlight the Connectors container. From the Action menu, choose New, and choose Remote Dirsync Requestor. The property pages of the Remote Dirsync Requestor, shown in Figure 18-22, are essentially the same as the Exchange Dirsync Requestor object.

Every MS Mail PO needs to be configured individually as a remote requestor. For instructions for configuring the property pages, see the Configuring Exchange as a Dirsync Requestor section earlier in this chapter.

Once you have configured the requestors in Exchange, reconfigure the MS Mail environment to support a new Dirsync Server. Work with your MS Mail Administrator to accomplish this task, following these basic steps:
1. Connect to the existing MS Mail Dirsync Server with Admin.exe. One of the options in the Dirsync menu for a PO designates the PO as the Dirsync Server. On the current MS Mail Dirsync Server PO, this option is set to Yes. Set the option to No to decommission this server as the Dirsync Server.

2. While connected to the old Dirsync Server PO in Admin.exe, configure it as a requestor to the Exchange connector PO Dirsync Server. Exchange appears in the list of POs only if you have previously defined it as an external PO. After you define it as a requestor, set a schedule compatible with the schedule you have set in Exchange, and check your Dirsync Requestor options to make sure they are correct.

3. Using Admin.exe, reconfigure all other MS Mail POs to be requestors to Exchange. Check requestor options and Dirsync T time schedules to make sure they are appropriate.

4. To ensure a clean slate for your new Dirsync Server, it is recommended that you reset MS Mail synchronization numbers after changing the Dirsync topology. A utility called Listds.exe can be run against the PO to reset these numbers; you can obtain this utility, and instructions for resetting Dirsync synchronization numbers, from Microsoft. As always, run a full backup of the PO before using any of the post office tools. Once you have reset the synch numbers, run a manual Dirsync export to give the new requestor PO fresh synch numbers.

To enable Exchange accounts to be replicated to MS Mail, you must enable the MS Mail address type in the recipient update policy. The MS Mail address type is not available until the connector is installed. To enable the MS Mail address type, go to the System Manager and expand the Recipients container. In the Recipient Policies container, open a policy and go to the E-mail Addresses tab to specify which types of addresses are automatically generated for the accounts that meet the policy's criteria.

**Migrating from MS Mail**

Migrating from MS Mail to Exchange involves moving a variety of data from the MS Mail systems to your Exchange organization: messages, user account information, personal address entries, appointments, and shared folders. The primary migration tool is the Exchange Migration Wizard. To migrate MS Mail, use an approach similar to the one described in the “cc: Mail migration tools” section earlier in this chapter. The Migration Wizard migrate directory data, mailboxes and shared folders, and calendar data.

Before using the Migration Wizard, the following conditions must be met:

✦ You know the path to the PO from which you are about to migrate.
✦ There is a network connection between the PO and the Exchange Server.
You can place the PO on the local machine for quicker migration.
✦ You have the Administrator account and password on the Mail PO.
✦ There are no users using the PO from which the migration process will happen and any MTA or Dispatch instances are closed. This will prevent any conflicts caused by open files.

Outlook can also help in an MS Mail migration. Using the Import/Export features in Outlook, you can migrate data from an MS Mail MMF (Mail Message File) to personal folder format.

For more information on Outlook and the Import feature, see Chapter 12.

Lotus Notes coexistence and migration

Lotus Notes is one of Exchange’s chief competitors. Many companies are interested in using some of the capabilities of Exchange but also have invested considerable resources in developing their Notes platform for messaging and applications. Exchange 2000 server can now coexist with the Notes/Domino platform more easily. Starting in Exchange 5.5 SP3, Microsoft made the Connector for Lotus Notes available, has continued to develop the connector for integrated messaging and directory replication, and has improved application support.

The Lotus Notes connector uses the Notes API to connect to the Notes database. Exchange uses Notes client .dll files to interface with the Notes databases, so that you must install the Notes client software from IBM as well as the Exchange connector software. The Connector for Lotus Notes provides excellent interoperability with Notes Versions 3.x, 4.x, and R5/Domino. The connector has the following functionality:

✦ Enables messaging exchange between Exchange and Notes servers
✦ Supports calendaring between the two systems
✦ Synchronizes directory information
✦ Supports Doclinks (hyperlinks that bring you to another section in a Notes doc, another Notes doc, or another Notes database)
✦ Supports OLE attachments and rich text format messages

The connector runs a number of different processes that facilitate transfer of information between Exchange and Notes. Four executables that control these processes: lsmexin.exe, lsmexnts.exe, lsmexout.exe, and lsntsmex.exe. The lsmexout.exe and lsmexnts.exe processes handle messages being transferred from Exchange to Notes, using temporary storage (MTS-OUT/READY-OUT) in the information store for message conversion. Messages leaving Exchange are placed into a Notes database called Mail.box. The lsmexin.exe and lsmexnts.exe processes
handle message flow from Notes to Exchange, also using a temporary storage location (MTS-IN/READY-IN) in the Information Store for message conversion. Messages are periodically picked up from a Notes Database called Exchange.box.

Using the Information Store for message conversion means that the processes benefit from transaction logging and diagnostic logging capabilities of the Store. Figure 18-23 illustrates data transfer between Exchange and Notes.

![Figure 18-23: The processes of the Connector for Lotus Notes](image)

### Configuring a connection to Notes

Communication with the Notes server requires preparations both on the Exchange Server and in the Notes environment. This section concentrates on the preparations needed to facilitate communication on an application level. To prepare network connectivity to a Notes Server, consult your Windows 2000 and Notes documentation.

Preparing the Notes environment for a connection from Exchange is complex, so you should work with the Notes Administrator. There are three steps to prepare the Notes environment, and you will be required to have administrative privileges to the Notes server in order to complete them:

1. **Configure a Notes ID to be used by the connector:** Have your Notes Administrator create an ID file with a blank password. Make sure the ID is not included in the address book, and use a name that identifies the ID as a connector ID (ExConnect). Place the ID file in a location available to the Exchange server where the connector will be installed.

2. **Grant the Notes ID necessary permissions on relevant databases:** The connector will retrieve messages, reference DocLinks, perform database maintenance, and perform directory synchronization using this ID, so it needs special privileges. To deliver messages from Exchange, the ID requires Depositor access to the Mail.box database. The ID requires Reader rights to any address books that will be used as sources for Notes-to-Exchange directory replication. In the Notes address book where Exchange recipients will be
replicated, the ID requires Editor access as well as the privilege to delete documents. The connector can initiate database maintenance on Notes databases that is similar to the Exchange IS maintenance. In order to take advantage of this functionality, list the connector’s Notes ID as an Administrator on the server to which it will connect.

3. **Define the Exchange environment as a foreign domain:** In the Name and Address Book, define the Exchange organization name as a Foreign Domain, specifying Exchange.box (by default) as the Gateway mail file name.

Once the Notes preparations are complete, prepare the Exchange Server. Assuming you have installed the connector software, now install the Notes client software. Depending on which version of Notes you are connecting to, you can use Version 4.x or Version 5.x of the client software. Work with the Notes Administrator to install the software. Once it is installed, verify that you can connect to the Notes Server with the client using the Notes ID created for the connector. Make sure that the path to the Notes client software (%installdrive%\Lotus\Notes) was added to the system environment variables, and make a note of the location of the notes.ini file. You will need to know the location of the notes.ini file when you set up the connector.

Once you can verify a client connection to the Notes Server, configure the connector in Exchange. In the System Manager, open the properties of the connector in the Connections object. In the General tab, configure the basic connection information for this connector. Figure 18-24 shows a connection with a server named NotesSvr01 in a Notes organization called MergeCo, with the connector mailbox information and the Notes.ini file path also filled in.

![Figure 18-24: Configuring general properties for the Connector for Lotus Notes](image-url)
Make sure that the connector mailbox name you enter here is the same as the one that you specified as the Gateway mail file name when you set up Exchange as a foreign Notes domain.

The polling interval configures the period used to poll the Exchange.box for messages awaiting delivery to Exchange.

The Address Space tab is similar to any of the other connectors. Use this tab to scope the connector, and specify valid addresses to be routed through the connector. Wildcards are enabled.

The Import Container tab specifies the container to which Notes addresses are replicated during directory synchronization. You should probably create an OU to place these recipients in, instead of putting them into one of your default object containers.

The Export Containers tab specifies all containers that are used for directory synchronization from Exchange to Notes, as shown in Figure 18-25.

![Diagram of Connector for Lotus Notes](image)

**Figure 18-25:** Configuring the Export container of the Connector for Lotus Notes

Any recipients that will be replicated to the Notes address book must have valid Notes e-mail addresses for replication. Enable the Notes address type in the appropriate recipient policies.
Once you have specified the Import and Export containers, set a Dirsync schedule in the Dirsync Options tab of the connector, as shown in Figure 18-26. Set your schedule to run during off-hours if directory replication latency is not an issue. The Dirsync schedule is completely customizable to fit your organizational and technical requirements. You can manually force an incremental or full directory replication in either direction in the Dirsync Options tab.

![Figure 18-26: Configuring the Dirsync schedule for the Notes Connector](image)

In the Advanced tab, you can specify important configurations such as message size limits, the delivery order of messages (FIFO, Priority, Size), and the routable downstream domains, as shown in Figure 18-27.

In the Routable domains dialog box, specify all Notes domains to which you wish to send messages. By default, you can send only to the domain to which you are directly connected.

**Tools for migrating from Notes to Exchange**

Besides the connector, two main tools help you migrate from Notes to Exchange: the Migration Wizard and the Application Converter.

The Migration Wizard helps you migrate messages with attachments, calendar information, and document links. This section describes the Migration Wizard in terms of Notes-specific information; more information appears in the *Using the Migration Wizard* section earlier in this chapter.
The Migration Wizard migrates user messages based on the default Notes message templates. If you are using custom templates, consider using the Exchange Application Services for Lotus Notes, described later in this section. Message attachments are migrated with the messages. Depending on how you configure the Wizard, you can convert DocLinks to URL format or attach them as RTF or OLE objects during the migration process. You can also migrate Calendar-type data such as appointments and tasks.

Connecting to a Notes server with the Migration Wizard requires preparations in both the Notes and Exchange environments, similar to the preparations for configuring the Notes connector. Before using the Migration Wizard, complete the following preparations:

1. Have your Notes Administrator create a Notes ID for the migration and give it Reader permission to all mailboxes that need to be migrated. Place the Notes ID file on a location available to the machine from which the Migration Wizard will be run (you can put it on a diskette).

2. Install Notes client software 4.x or 5.x (whichever you use to connect to your Notes server) on the machine that will perform the migration. Make a note of the notes.ini file path, and make sure that the path to the client application files is added to the system environment variables (for example, E:\Lotus\Notes).

3. Test the connection to the Notes server by configuring the client to connect to a database on the server.
Once you have connected to the server and been given permissions to the mailboxes you intend to migrate, you are ready to run the Wizard.

The Migration Wizard does not migrate encrypted messages or custom applications. If you need to migrate applications with scripting or user-defined fields, consider using the Exchange Application Services for Lotus Notes. The application converter component in Application Services enables you to migrate some types of applications (including MAPI and CDO) from Notes to Exchange.

The first step in converting applications is to interview the developers, read the application documentation, and explore the workings of the application. Once you have discovered all you can about the application, use the Application Converter to migrate the structure of the application—including views, forms, custom fields, scripts and so on—to an Exchange folder-based application. As soon as you are confident about the new Exchange application, use the converter to copy the actual data to the Exchange Information store. The Application Converter can also replicate data between Notes and Exchange in a coexistence scenario.

Of course, the application development environments of Exchange and Notes are noticeably different. Some applications may not convert directly to Exchange and it sometimes takes work—tweaking permissions, writing custom scripts, or possibly even changing the logic of the application—to get a converted Notes application to work in Exchange. Read the readme.txt file in the \Migrate\asn\setup\i386 folder on the Exchange CD for system requirements and setup information.

Coexisting and Migrating Novell GroupWise

GroupWise is a common messaging system in the Novell environment. Some organizations are choosing to migrate from Novell to Windows 2000 for file and print, directory, or application services such as SQL. In these cases, it often makes sense to migrate to Exchange at the same time. Other organizations are choosing to migrate to Exchange to get additional features, scalability, reliability, or management capabilities that are unavailable in GroupWise. Many tools can help you integrate Windows and Exchange into the Novell/GroupWise environment and migrate Novell resources if you choose.

If you plan to operate Exchange 2000 in a GroupWise 4.x or 5.x environment for an extended period, set up the Connector for Novell GroupWise, which offers the best interoperability and the most functionality of all the GroupWise tools available. The connector supports message transfer, directory synchronization, and calendar-item transfer between the systems. Calendar Free-Busy sharing via the connector is not available in the initial release of Exchange 2000, but is planned for a future service pack. In the meantime, a Calendar synchronization tool available on the Exchange 2000 CD can synchronize the Free-Busy information on Exchange 2000 and GroupWise servers.
The Connector for Novell GroupWise works in a way similar to the Connector for Lotus Notes. The main difference is that the Exchange server does not connect directly to the GroupWise server, as it connects to a Notes server. When Exchange connects to a Notes server, it uses client DLL code to access the database directly. When Exchange connects to a GroupWise system, it uses an API gateway.

Novell provided the API gateway to facilitate connections to GroupWise from other applications. Exchange uses the API gateway to exchange data of all types with the GroupWise server—messages, task and appointment data types, attachments, directory queries, and directory updates. Both the Exchange and GroupWise systems have to convert data to the API format before transferring it to the gateway. Converted data destined for GroupWise is placed in the API-IN (messages) and ATT-IN (attachment) directories in the gateway. Data sent outbound from GroupWise is placed in the API-OUT and ATT-OUT directories in the gateway.

Transferring data from Exchange to GroupWise involves several steps. This example assumes that the message is addressed to a recipient, but the basic process is the same for directory messages:

1. The Exchange server identifies the recipient as a GroupWise recipient by its address.
2. The routing engine directs the message to a server that has the GroupWise connector. Once at the connector server, the message is passed to the connector.
3. The connector service formats the message to the API gateway standard and passes the message to the Exchange router service for GroupWise.
4. The router service for GroupWise connects to the API gateway and places the message and any attachments into the inbound queue (API-IN/ATT-IN).
5. The API gateway converts the message to the Novell native format and passes the message to the GroupWise MTA to be routed to the destination GroupWise server.

Messages traveling from GroupWise to Exchange follow the same path in the opposite direction. The GroupWise MTA converts the message to the API format and passes the message to the API gateway outbound queue (API-OUT/ATT-OUT). The Exchange router service for GroupWise periodically polls the gateway and picks up any new messages that have arrived. Once picked up, the messages are passed to the GroupWise connector service to be converted to Exchange format and passed to an inbound queue of the Exchange routing engine. The message is then routed to the recipient’s home server for delivery. Figure 18-28 illustrates the basic message processing that occurs when a message is sent round trip from Exchange to GroupWise.
Preparing the GroupWise environment

Install the GroupWise API Gateway NLM Version 4.1 on the GroupWise domain that you will communicate with. After the gateway installation is complete, install Patch 2 for the API gateway, enabling GroupWise to expand distribution lists sent from exchange clients.

Configure the API gateway, according to the gateway documentation, specifically for the version of GroupWise that you use. You should also configure the gateway to enable distribution lists. Once it is configured, start the gateway NLM.

GroupWise must be configured to use the API gateway for any messages destined for Exchange recipients. To do this, use the GroupWise Administrator (Version 4.x) program or the NetWare Administrator program (Version 5.x) to define Exchange as an External Foreign Domain in the GroupWise system. Once it is configured, configure a Gateway Link for the gateway. The procedure and exact configuration of the Foreign Domain and gateway link depends on which version of GroupWise is being used; the important thing is to make sure that the Foreign Domain name in GroupWise matches the GWISE-type address value defined in the E-Mail Addresses tab of the recipient policy in the Active Directory. You can access the recipient policy in the Exchange System Manager snap-in.
The Exchange Router service for GroupWise, to connect to the API gateway, needs access to the gateway directories. Using the NetWare Administrator program, create an NTGateway group, and give the group Read and Write access to the queue directories of the API. Create an account, place it in the NTGateway group, and note the password. The Exchange Router Service for GroupWise uses this account to connect to the gateway.

**Configuring Exchange to connect to GroupWise**

Assuming you have installed the connector, use the Exchange System Manager to configure the connection. Begin by viewing the properties of the Connector for Novell GroupWise, which is located in the Connectors container. In the General tab of the connector, enter the UNC path to the API gateway on the NetWare server. Provide the NetWare account name and password that has directory access to the gateway—that is, the account placed in the NTGateway group during GroupWise preparation.

Configure the Address space for the connector as you would for any other connector. Use the tab to scope the connector, and specify valid addresses to be routed through the connector. You can use wildcards such as * to specify the range if valid addresses for the connector.

In the Import Container tab, specify the container to which GroupWise addresses are replicated during directory synchronization. You should probably create an OU (for example, GroupWise Recipients) to place these recipients in, instead of putting them into one of your default object containers. As with the other connectors that support synchronization, you can choose which types of objects (default is contact) are added to the AD. You can also apply an include filter or an exclude filter to the addresses being replicated from GroupWise.

In the Export Containers tab, specify all containers used for directory synchronization from Exchange to GroupWise. You can also specify whether to export Windows 2000 groups and contacts.

Once you have specified the Import and Export containers, set a Dirsync schedule in the Dirsync Schedule tab. If directory replication latency is not an issue or you can choose one of the schedules in the drop-down list, set your schedule to run during off-hours; nevertheless, the Dirsync schedule is completely customizable to fit your organizational and technical requirements. To set a custom schedule, click on the Customize button. You can manually initiate an incremental or full directory replication in either direction.

**GroupWise migration to Exchange 2000**

The Migration Wizard is the best tool for performing GroupWise migrations. To connect to the GroupWise domain with the Migration Wizard, make some preparations in the NetWare environment. These preparations are identical to the preparations described in the *Preparing the GroupWise Environment* section earlier in the chapter.
The migration account, which is specified in the Migration Wizard, must have rights to all data that is to be migrated. Depending on which data you choose to migrate, the rights can include access; delete; and modification to messages, tasks, posts, and appointments. Normally, these procedures fall to each individual user. A macro is provided in the Migrate\Tools\GWise directory on the Exchange server CD to help you make these changes en masse.

The Wizard converts messages and their attachments, tasks, appointments, posts, and the underlying mailbox folder structure. Most data is migrated to its Exchange equivalent; GroupWise tasks are migrated to Outlook tasks, GroupWise appointments are imported into the Outlook Calendar, and phone messages are migrated to Outlook messages. During the migration, some information has to be interpreted and altered slightly. For example, tasks that have not been accepted are migrated as messages.

Other options for coexistence

This chapter has focused on connectors designed to operate in long-term coexistence. Compared to a generic connector such as X.400 or SMTP, these connectors — Notes, cc: Mail, MS Mail, and GroupWise — are designed for more functionality. You should evaluate the functionality carefully and decide whether your organization wants or needs one of these connectors. You may decide that features like calendar synchronization, native addressing, and advanced directory synchronization are not required for your environment. Also, there may not be a specific connector for the system you need to connect to. In such cases, there are other strategies you can use to enable more basic connections to other systems.

The SMTP connector was built in compliance with RFC 821 and 822. Therefore, it can interoperate with almost any system that has an SMTP service, including LAN-based systems with SMTP gateways, Unix SendMail systems, and other IP-based mail systems that support the RFC standards. SMTP is a generic connector that affords great performance as well as a standardized and familiar configuration.

Like SMTP, the X.400 standard was developed for international messaging applications. The X.400 connector can connect Exchange to systems that support the 1984, 1988, or 1992 X.400 standard.

Consider using SMTP or X.400 in two situations:

✦ You need to provide only basic messaging functionality to another system.
✦ The messaging system you will coexist with does not have a system-specific connector in Exchange.

For details about the X.400 and SMTP connectors, see Chapter 10.
Also useful for migration is the ADClean utility, which helps clean up directories during. Although single-step migrations (migrating users and their mailbox data to Exchange 2000 in the same process) can be performed in small environments, in many cases this strategy is not feasible. In coexistence situations, it is common to configure directory synchronization to create contacts in the AD, with which Exchange users can address messages to recipients in the foreign messaging system. When the users in the foreign system are migrated, they need to have mail-enabled accounts created for them, and doing so can cause duplicate accounts. ADClean identifies duplicate accounts, assisting in account information merging and cleaning up the Active Directory. It can assist in migrations from foreign systems and upgrades from previous versions of Exchange.

Tip
Visit the Microsoft Web site for Exchange 2000 to find the latest tools and white papers available for migration. The URL for Exchange deployment and migration is www.microsoft.com/exchange/techinfo/deploymigrate.htm

Summary
This chapter focuses on tools and techniques for achieving productive coexistence with other messaging and collaboration applications. It explains the preparation, installation, and configuration of connectors that most coexistence situations require and examines internal messaging handling for each connector. Beyond coexistence, the chapter looks at some of the tools available for planning and performing a migration to Exchange 2000. Whether you are planning long-term coexistence or a migration to Exchange 2000, applying the information in this chapter will help ensure that your project is successful.

✦✦✦
This chapter discusses the principles of configuring hardware for an Exchange 2000 Server installation, reviewing a number of basic principles and several possible configurations. Exchange 2000 is installed on Windows 2000 Server. An excellent companion reference book referred to in this chapter is *Windows 2000 Server Secrets*, published by IDG Books, Inc.

Creating the hardware server platform, on which you will load Windows 2000 Server and Exchange 2000 Server, requires three basic steps:

- Choosing the hardware, basing your choice on hardware specifications, expected usage, and budget
- Configuring the hardware components to maximize the performance and fault tolerance of the server
- Mapping the software components of Windows 2000 Server and Exchange 2000 Server onto the hardware to optimize performance and fault tolerance

### Choosing Hardware

For a supportable system, your production Exchange 2000 Server hardware components must be included on the Microsoft Hardware Compatibility List (HCL), which can be found at [http://www.microsoft.com/hcl](http://www.microsoft.com/hcl)
If you are assembling server hardware yourself, ensure that all the components are on the list. If your hardware is not on the HCL Exchange, Windows 2000 may behave inexplicably and fail during or after installation. You will probably have to get new hardware.

**Microsoft hardware requirements**

The Exchange Server administrative manuals list minimum requirements for hardware:

- **Memory:** 128MB or more
- **Processor:** 166 MHz Pentium or faster
- **Hard Drive Storage:** NTFS partition with at least 4GB of available disk space for Exchange 2000 Server

For more information on hardware, see http://support.microsoft.com/support/kb/articles/q262/0/68.asp

This list is the absolutely minimal configuration on which Exchange 2000 Server will install and run, assuming nothing else at all is running. But it is not recommended (or endorsed by Microsoft) for even the most minimal serious-production server. You will not be able to take advantage of the utility provided in Exchange 2000 Server with so little horsepower. To put it another way, no one except Microsoft has ever built Exchange on the minimal server-hardware profile, and they did it only to answer the inevitable question from customers and the press, “What is the minimum hardware configuration to run Exchange server?” A better question is, “What are the minimum practical requirements?” The answer to this question is a bit more complex.

**Minimum practical requirements**

According to the minimum hardware configuration specified by Microsoft, the hard drive storage space required is 4GB in an NTFS partition, implying that a single hard drive of that storage capacity is sufficient to load and run Exchange 2000. But for Exchange 2000 to run with performance and data protection, the minimum requirement is to have two separate physical hard drives. This configuration will both increase performance and, if configured as shown in Figure 19-1, protect your data in case of a failure of one of the hard drives.

It is important to distinguish between two logical drives contained on a single physical drive and two logical drives each contained on a separate physical drive. The former does not improve performance or data protection, whereas the latter does.
If you have a single-hard-drive system and the hard drive physically fails, you will be unable to recover any data since the last full backup of the Exchange data stores; you will have lost both the current information store and the logs required to roll forward the previous full backup. In small systems with a limited number of users, relatively low e-mail traffic, and frequent full backups, you may not need multiple physical drives to achieve acceptable end-user performance or data integrity. However, separating the logs from the information store on different physical hard drives increases the performance significantly in an active system. The higher the activity, the greater the difference in performance will be when the logs are separated. Although low performance may be acceptable in small systems with low utilization, it is often unacceptable in terms of fault tolerance and data integrity. In a system of at least two drives where the logs are on a separate physical drive from the data stores, the following failure and recovery modes can occur:

- **The hard drive with the information stores fails:** In this instance, the last full backup is restored, and the logs which are secure on the separate drive are used to roll forward the information stores to the time of failure. This assumes that circular logging is turned off, that the logs have not been deleted since the last successful full backup, and that there is a successful full backup of the information stores.

- **The hard drive with the logs fails:** In this instance, the drive should be replaced and an immediate full backup performed. This response protects you in case the hard drive with the information store fails later. Make certain circular logging is turned off.

In a two-drive system, you can mirror the drives to obtain a higher level of fault tolerance; if one drive fails, the second drive will continue operating. This configuration does not give the performance of the previous configuration, although it is the right choice for you. It requires you to monitor the health of the drives closely. If the first drive fails and the failure remains undetected and repaired, then you have the same problem as in a single-drive system. Exchange 2000 stops functioning, and Windows 2000 ought to be configured to alert the Administrator.

A four-drive system addresses the need, in case of failure, for continued functioning and the ability to recover the data. In this instance, there are two mirrored sets of drives (RAID 1), as shown in Figure 19-1. The system (Windows 2000 Server) and the logs are on one mirrored drive set and the data stores on the second. This configuration provides good performance; even if one drive fails, the other will continue to function. You should still monitor the health of the drives, ideally with a system that sends an automatic e-mail alert.
Choosing a physical drive

You have two major choices for hard drives: Small Computer System Interface (SCSI) and Intelligent Drive Electronics (IDE). Just a few years ago, most people said that you should never build a server with IDE drives, but should use SCSI. Nevertheless, IDE has improved greatly. For smaller systems, ATA 100 drives, cables, and Enhanced IDE controllers can provide snappy I/O. More expensive EIDE controllers can also do RAID.

Avoid older, slower IDE ATA 33 and 66 drives and controllers; I/O is a large part of the performance bottleneck for almost any Exchange server.

SCSI continues to improve as well. For medium-sized and large systems, SCSI is usually preferred. Buy SCSI controllers and drives fast enough to gain the performance you will undoubtedly need. Advanced SCSI controller cards can do on-board RAID and relieve the CPU from the overhead of software RAID.

Memory

Microsoft’s minimum recommended memory configuration is 128MB, but a minimum production server should probably have 256MB or more of memory. It is important to minimize the number of times that the system page file (virtual memory) is used.
When that happens, the hard drive is used as memory to supplement the physical memory of the server. The time to access physical memory is measured in nanoseconds \((10^{-9})\), whereas the time to access disk drives is measured in milliseconds \((10^{-3})\) — between 100,000 and 1,000,000 times slower. The cost of memory has come down significantly in the past few years, and adding more will dramatically increase performance.

**Tip**

When buying memory, buy the largest DIMMs or SIMMs you can, in order to leave banks open for subsequent low-cost and easy expansion.

Typical servers for medium-sized environments should have 512MB of memory and larger environments 1GB. Although this amount of memory increases the cost of the server, it also substantially increases performance.

**Processors**

Although the minimum hardware configuration specifies a Pentium 166 MHz, this level may not provide reasonable performance. In a small environment (under 50 users), a well-priced Pentium III with 1MB or more on-board cache is sufficient. In larger environments, faster processors and multiple processors are needed to provide adequate performance.

**Note**

Buy the processor with the best cost/performance ratio; usually this is not the latest processor, but one slightly older.

**Multiprocessor motherboards**

Determining whether to buy a motherboard with multiple CPUs is complicated. The benefit obtained from additional processors, although significant, is not a linear increase in performance. This complication can be due to several factors:

- The factor limiting performance may not be processing capacity but memory availability, disk I/O, or NIC congestion.
- Where server performance is limited by CPU processing power, a certain amount of system overhead must be allocated to each additional processor. Windows 2000 has worked to minimize this overhead, making the real increase in processing power more closely approach the theoretical limit.

Adding a second processor does, however, increase the fault tolerance of the machine by providing a redundant processor in case one fails. As SMP has matured, new chipsets have significantly reduced the cost of adding multiple processors, making dual- and quad-processor designs very cost-effective. At the same time, after a second processor has been added, you must decide whether adding more processors is as advantageous as adding another server, and whether that server should be added to a Windows 2000 Server cluster. Windows 2000 provides an operating system in which the increases in computing power can be small, medium-sized, or large, and in which scalability is extremely flexible.
To determine the number of processors to have on a machine, look at what the Exchange Server is doing. It is not the number of mailboxes (that is, users) on an Exchange server, but the mail traffic, usage patterns (such as users using server-based inbox rules), and server services that increase the load on the processors. Thus, the appropriate questions to ask are, “On average, how many messages will be processed per day?” “What services will be running on the server?” and “What other CPU intensive tasks will be using up those MIPs (Million Instructions Per Second)?” You may find that answers vary within your organization, with some business units having much higher average message traffic, different patterns of usage and needs for different services on their Exchange servers.

If the server is already in production, use Task Manager to look at the processor utilization. This will give you a good idea of whether that is a bottleneck. If it is over 50 percent, it may sometimes spike high and become the limiting step in processing messages. If so, you should probably add another processor.

When purchasing a new server, you may want to purchase additional processing power. If your budget is limited, a cost-effective way to provide expansion room while limiting up-front cost is to purchase a motherboard or chassis that is two- or four-processor capable, but with a single processor installed. Later, you can add processors as needed. This provides a relatively inexpensive upgrade path for future processing power, instead of replacing a whole server or adding a new server.

If your model includes an extensive pilot program, you may be able to defer additional processors until you get into real production, perhaps months after you start your pilot.

**Disk storage systems**

If you have an existing e-mail system, a good step in determining the size of disk storage is to examine your current system and answer the following questions:

- What are the current space requirements — both private (individual e-mail storage) and public (public folders in a Microsoft system) — for all messaging in the organization?

- What was the rate of growth in storage requirements over the past year, six months, three months? Has the rate of growth increased over those times? The rate of growth of e-mail storage can be accelerating (it rarely declines) including:
  - E-mail is achieving greater acceptance and thereby utilization within your organization.
  - Applications, both at the corporate and at the business unit level, are being written that utilize e-mail as both a transport and storage facility. This is particularity so of hybrid workflow applications.
  - Knowing the expected usage is key to planning for storage requirements several months to a year out from installation.
What new applications will use your e-mail system as a storage facility? Which ones does your company plan on implementing in the next year? What are the estimated delivery dates? What are the storage requirements estimated by the application development team? Answers to these questions will help you better gauge the requirements for the successful rollout of these applications and assess their effect on your deployment plans.

Does the company have a policy limiting e-mail storage by individuals and retention of information in Public folders? Are they enforced? Many companies do not have policies in these areas, or have them and do not enforce them. This question is not generally an IT question. It is more properly a business question for the business management, but which you need an answer to size the Exchange 2000 Server properly.

When you have answers to these questions or reasonable estimates, combine them with some of the information below to calculate disk storage requirements:

- The system software for Windows 2000 server and Exchange 2000 Server takes approximately 1GB of space.
- The page file for a Windows 2000 server should be equal to 50 to 75 percent of memory size.
- Each message sent or received from the last full backup is recorded in the transaction log (with circular logging off).
- A message sent to multiple users is stored once in each database in which one of the recipient users is found.

This approach may appear different from that of previous versions of Exchange, where the system stored only one instance. In reality, it is the same — except that now you are allowed more than one database and get one instance per database.

For example, suppose you have an Exchange 2000 Server system with four databases, but with six users. Each user is receiving a message. Three users store mail in one database and the other three in three separate databases. The total number of instances for which that message is stored is 1 instance (DB with 3 users) + 3 instances (1 user in each of 3 DBs) + 1 instance in the transaction log = 5 instances.

The figures used here demonstrate how to calculate the storage requirements and should not be used as a formula for your actual calculations. To produce that formula, review the assumptions and historical data for your company.

If you have a company of 500 mailboxes that will have 3,000 e-mails per day with an average size of 100KB, 50 percent of which are sent to users assigned to two different databases, and which has a full backup each week, then the total storage required per week is as follows:
First week:

\[(1,500 \times 100\text{KB}) \text{[50 percent of e-mails sent to one database]} + (1,500 \times 2 \times 100\text{KB}) \text{[the 50 percent of e-mails sent to two databases]} + (3,000 \times 100\text{KB}) \text{[e-mail stored in the transaction logs]} = 750\text{MB}\]

Second week (assuming a successful backup that clears the old logs):

\[(1,500 \times 100\text{KB}) \text{[the 50 percent of e-mails sent to one database]} + (1,500 \times 2 \times 100\text{KB}) \text{[the 50 percent of e-mails sent to two databases]} = 450\text{MB} \text{[no additional storage requirement for the transaction log, since it is cleared with the weekly backup]}\]

To assess the amount of storage needed, you need to assess the retention rate for e-mail. The retention rate can be affected by maximum-storage rules of your company, historical e-mail–retention patterns of end-users, or the requirements of e-mail–based application. Once you have reviewed and analyzed these factors, you can make some assumptions regarding the amount of e-mail retained and factor these assumptions into the storage requirements. For example, suppose that 40 percent of all e-mail received is read and discarded within a week after receipt, that 60 percent is retained for a period of 12 months, and that only 10 percent of the 60 percent is retained indefinitely. Suppose also that the growth of e-mail by the existing employee base is 10 percent annually and that the company will add 15 percent more employees in the next 12 months. You should consider all of these factors in order to get an estimate of the storage required in the next 12 months. The calculation is as follows:

Total amount of e-mail received in 12 months:

\[(1 \times 750\text{MB}) \text{[the first week’s storage]} + (51 \times 450\text{MB}) \text{[remaining 51 weeks’ storage]} \times (1 + (0.1/2)) \text{[reflecting the increase in the e-mail traffic of 10 percent annually, yielding an average of 5 percent increase for the whole year]} \times (1 + (0.15/2)) \text{[reflecting an increase in employment of 15 percent for the entire year, yielding an average of 7.5 percent for the year]} = 26,751\text{MB}\]

The amount retained the first year would be

\[26,751\text{MB} \times .6 = 16,050\text{MB}, \text{or approximately 16GB}\]

Next, calculate the amount of storage required for public folders. This amount will be dictated to a great extent by the rules and access rights you give to people accessing public folders. If you have a current e-mail system and plan to maintain the same rules, then historical data growth should be relevant. Assume that your current public folders are 4GB in size and have historically expanded at a net rate of 15 percent per year (accounting for additions and removals). The amount needed is

\[(1 + .15) \times 4\text{GB} = 4.6\text{GB}\]
The total size of the e-mail and public folders stored is 20.6GB. In general, both for performance and in order to account for the uncertainties contained in the calculation, the estimated storage requirement should be no more than 75 percent of the disk storage capacity. Allow for the reduction in disk capacity due to the use of RAID. After deducting for the implementation of any RAID storage, the amount of storage is be calculated as follows:

\[ 20.6GB \div 0.75 = \text{approximately } 27.5GB \]

The CD-ROM contains a checklist and Excel spreadsheet to help you with your calculations.

**RAID disk storage**

Each type of RAID configuration has different cost points, levels of fault tolerance, and performance. Windows 2000 allows software RAID, but for performance reasons (lower CPU and I/O overhead) hardware RAID provided by the manufacturers of the controller cards is better. The most common types of RAID used in configuring a Windows 2000 and/or Exchange 2000 server are RAID 1, RAID 0/1, and RAID 5.

It is important to distinguish the hardware configurations most suitable for Exchange 2000 and Exchange 2000 Enterprise Edition. Each version has different capabilities, especially in the size and configuration of the information store. The difference affects the design and size of the hard disk storage that is suitable for each.

Exchange 2000 allows for multiple databases on an Exchange 2000 server. These databases can be backed up and/or restored independently or concurrently. Although separate databases need not be contained on separate RAID structures, there are distinct advantages to doing so. For example, suppose multiple databases are present on the same physical RAID 5 or RAID 0/1 structures. While data is backed up or restored to one dismounted database with the other(s), mounted e-mail is concurrently being written and retrieved to the mounted database(s). Backing up or restoring a database inhibits performance. If a RAID 5 structure had eight hard drives, it is far better to split them into two RAID 5 sets of four drives and mount two databases, one on each. That way, when a backup or restore is being performed on one database, the second database will not suffer in performance — especially if the databases can be split onto separate I/O channels on the RAID controller card.

Exchange 2000 permits up to five databases per storage group and up to four storage groups per Exchange 2000 Enterprise server. By contrast, Exchange 5.5 and earlier versions allow only a single storage group with two databases (public and private). Each storage group has a single transaction log, which should be located on a separate physical hard drive.
Backup: tape drives and WAN/LAN backup

Since Exchange 5.5 did not allow concurrent backup of components within an Information Store, additional tape backup devices on a single server could not be used efficiently. Thus, although Exchange 5.5 Enterprise server removed the 16GB limit on the size of the information store, practical considerations (that is, the time required to back up or, especially, restore an information store) came into play. Exchange 2000 Server Enterprise Edition significantly removes this practical limitation by allowing clustering, and by allowing concurrent backup and restore of databases within multiple Exchange 2000 storage groups. If more than one storage group is located on a server, multiple tape drives can be a significant advantage in both backing up and restoring an information store. In the case of WAN/LAN backup, it is often desirable to have a separate backup network with separate NIC cards, wiring, hubs switches and routers, if backup is being performed during periods of moderate to high user traffic. In the case of a LAN where the servers are centrally located, a separate backup network is particularly cost-effective and can often be accomplished with an additional NIC card per server and an additional hub or switch.

Add multiple tape drives to a server and make multiple information stores to create two or more databases. One database from each information store can be backed up or restored simultaneously. Tape drives are inexpensive compared to the increase in performance during backup and restore. Even most tape changers have a single read/write station, which allows only one tape to be read or written to at a time. Two tape drives are better than one.

Disk I/O channels

When configuring a server to host Exchange 2000, try to balance the data throughput across I/O channels. Many RAID controllers today have two or more I/O channels. Balancing the data transfer across these will help maximize performance of any given hardware configuration. Often, adding an I/O channel, even if it means purchasing an additional controller card, is a wise investment in terms of the increase in performance. You can use the Windows 2000 Server performance monitor to determine the load on each of the I/O channels, giving an idea of whether redistribution of that load will increase performance. If one channel is consistently running at greater than 90 percent, then the load should be redistributed to other channels, or a new I/O channel added.

Network interface cards

In a small system, an inexpensive 100BaseT Ethernet card is sufficient to handle the network traffic coming into the server. If traffic is greater and/or fault tolerance is required, multiple 100BaseT or GB Ethernet with PCI bus mastering cards can be added. If the Exchange 2000 Server is being used as a connection to the Internet in a Windows 2000 cluster, Windows 2000 will not stop broadcasting the DNS entry associated with the TCP/IP address for a NIC on a server if the NIC fails. Some third-party
software programs provide that functionality and are probably good investments. If they are not added, however, and the failed NIC address is broadcast, any user presented with the failed NIC’s TCP/IP address will fail to make a connection. With the third-party software, when there is a failure, the physical TCP/IP address of the failed NIC is removed from the addresses assigned to the virtual cluster server. The user is automatically routed to one of the functioning cards and makes a good connection.

**Redundant power supplies**

Redundant power supplies remove a possible failure point in the server, an important function as the servers get larger and handle a larger percentage of the total user population. This function has been a continuing trend in larger corporations, to reduce the cost of administration.

**Uninterruptible power supply**

An Uninterruptible Power Supply (UPS) allows the server to continue processing in a momentary power outage, or to shut down without data loss in an extended power outage. Many units also have brownout- and gradient protection. Where continued operation is critical, natural gas–operated or diesel-operated generating can provide extended periods of power beyond the capability of the UPS. These extended power systems must supply power to all the components necessary for the Exchange 2000 Server system, including hubs, routers, modems, and firewalls. If the hub connecting the router and the server loses power, the system will not function even if the server is running.

**Building Large Systems**

Organizations wanting to build large systems can take advantage of new features in Exchange 2000, or improvements in the old features designed for exactly that purpose.

**Four-way active clustering**

Windows 2000 Server allows you to cluster four Windows 2000 servers in an active/active mode to create a virtual server. Thus, four Exchange 2000 servers can act as a cluster. Since an Exchange 2000 Server can have a maximum of four data storage groups, it is critical when designing the Exchange 2000 server storage group architecture to examine what would happen in the case of a node failure (that is, failure of the physical server). In that event, the remaining nodes take over the processing of the data storage groups for the failed node. If the nodes that continue to operate have to serve more than four storage groups, then a failure would occur. The possible storage group configurations for a two-node system are shown in Table 19-1.
Table 19-1
Possible Information Store Configurations on a Node (Server) in a Two-node Exchange Server Cluster

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

Scalability using front-end/back-end architecture

The flexibility of the Windows 2000 Exchange 2000 architecture allows the distribution of the processes to multiple servers. This distribution allows the hosting of connectors on one or more servers, hosting of the Internet e-mail connection on others, and the message stores on several more. This type of architecture is commonly referred to as a front-end/back-end architecture, as shown in Figure 19-2. The servers hosting connectors (including the Internet connection) and receiving all inbound calls are the front end. They communicate with the back-end servers, which serve as message store servers and never receive inbounds directly from users. In larger environments, the back end is often a Windows 2000 Server cluster with shared disk storage.

A good example of a front-end/back-end configuration designed for a two million–user environment is found in the white paper located at http://www.microsoft.com/Exchange/techinfo/2millionconfig.htm

Figure 19-2: Front-end/back-end server architecture
Microsoft BackOffice Resource Kit
LOADSIM.exe

The load simulator is a very useful tool for testing a specific hardware configuration against the design parameters of your e-mail system. There are a number of specific tasks that are options in LOADSIM.exe. The tester decides the mix, quantity, and frequency with which these various tasks are to be performed. The validity of the test is (to a great extent) determined by the care and accuracy with which the tester simulates the actual conditions that will be encountered when making these selections. The following tasks are available:

- **Inbox task:** User reads a new e-mail.
- **Browse task:** User reads an old e-mail.
- **Send mail task:** User sends an e-mail.
- **Schedule+ task:** User schedules a task in Schedule+.
- **Public folder task:** User accesses a public folder.

You can also vary the number of days over which LOADSIM.exe will run. These are virtual days and take somewhat less than a real day to execute. In order to perform this test, you will need a server with the hardware configuration you are considering using. By running LOADSIM.exe against that server, you will be better able to determine whether or not it will meet your needs.

**Summary**

For good cost/performance value from your Exchange servers, you need to be astute about selecting hardware and hardware components, configuring the components, mapping software to hardware, and upgrading inexpensively. You need to be knowledgeable on these topics—even if you are buying complete servers from your vendor. Servers are not all the same. A bit of attention to the areas covered in this chapter helps ensure that your system will deliver, today and as it grows.

✦✦✦
Performing a Smart Installation

In Chapter 16, we presented a walk-through install for a very simplistic Exchange implementation. However, the quick install does not address basic configuration issues with most test, pilot, and production environments. When you ignore basic configuration issues, you have a less than smart installation. We recommend that you read Chapter 17 before you read this chapter, since it describes the planning and preparation upon which smart installation depends. During the planning process, you will define many of the requirements for your Exchange installation and arbitrate the various ways to achieve them. Arbitration entails an understanding of configuration options, as sometimes these will constrain or offer insights into the best choices.

In addition, there are server-specific items that need to be considered and directory preparations that should be completed before you begin the installation process. If you have done your planning and preparation properly, and then configure properly, the installation should be successful. We call it a smart install because the likelihood of success and lower costs are increased with planning and preparation. We believe increasing the odds for success and reducing cost is a positive manifestation of applied IQ. You’ll want to take a survey of your work when you’re finished to make sure that your installation was completed properly and to keep track of what has changed.

If your organization is deploying many Exchange servers, or needs to ship servers to remote locations to be installed, you may want to automate your setup process. We will discuss some of the tools and techniques you can use to accomplish this task and leverage the thinking and effort that went into your smart install.
Focusing on Installation Scenarios

This section will discuss some common reasons for undertaking an Exchange installation and the high-level items to be concerned with to ensure success. If you have done planning, as suggested in Chapter 17 you will have a good idea of how these issues would be arbitrated in your organization. If not, you may understand well the implications of improper planning as you try to discern the right answers for your organization without the information to come to an informed conclusion.

Installing Exchange Server in a new messaging environment

We define this type of installation as—no existing messaging system in place and Exchange 2000 will be the only system used. Planning for this type of installation is usually easier than environments with legacy messaging systems or operating systems because there are no older systems to contend with and interface with. Much of your planning will revolve around the Active Directory considerations, server specifics, and third-party software choices. You’ll want to define sensible naming conventions for Exchange entities. Any production installation should include an enterprise virus protection solution and Exchange backup solution, at the very least. More often than not, production installations will also include Internet mail connectivity, firewall, and various types of client connectivity.

A new messaging environment will require more consideration of standards and policy than in other scenarios. In order to ensure easier support, maintenance, and upgrades add standards and policy creation planning to your technical and organizational planning. In particular, you’ll want to define size limits for messages passing through gateways and connectors, as well as for mailboxes. You’ll want to think about how groups will be used, and who will be allowed to use them. You’ll also want to think about how public folders will be used, who will use them, and who will maintain them. The use of contacts should be considered, too. Where will they be put in the Active Directory and which address books will they show up in? Who should be adding them to the Directory, and will there be an approval process for adding a contact? These are just a few of the things you should think about before putting Exchange into production, but you’ll want to make sure that you are defining and communicating policy early in the production implementation.

Installing an Exchange 2000 server in an existing Exchange 5.5 environment

Exchange 2000 Server can be installed to coexist with Exchange 5.5 servers running in a production environment. This type of installation requires minimal planning for
coexistence and, in almost every case, eventual migration. In this situation, we recommend you install the Active Directory Connector in order to achieve directory coexistence between Exchange 5.5 and Windows 2000. The Active Directory Connector can be installed, configured, and tested before you install the first Exchange 2000 machine. In addition, you should be planning the upgrade of Exchange 5.5 client and server systems or planning to migrate your Exchange 5.5 equipment and users to Exchange 2000 machines.

You can introduce Exchange 2000 into an Exchange 5.5 environment in a couple of ways: by upgrading an existing server to Exchange 2000, or by installing a new server. Both methods will have the effect of introducing Exchange 2000, but the process and potential risks are much different. While there is no standard way to upgrade your Exchange organization, we prefer a conservative plan that introduces Exchange 2000 in a deliberate and controlled manner. This means installing Exchange 2000 on fresh boxes rather than upgrading existing machines. In larger organizations, the migration process may be easier to manage by introducing Exchange 2000 on separate machines, though keep in mind that in such organizations, the Exchange servers may be large and relatively expensive—a factor that complicates the idea of using fresh equipment. On the other hand, this strategy allows you to move Exchange components one at a time, which would benefit teams that have little experience with Windows 2000 and Exchange 2000 in production environments.

If you choose to do an in-place upgrade, you need to make sure that the existing Exchange 5.5 server is ready to be upgraded—in other words, that there are no issues with the existing Exchange 5.5 hardware, configuration, software version levels, and so on. Make sure you have read through the release notes for potential problems and preparations required for upgrade from whatever you are running. Also, make sure you have a rollback plan in case the upgrade fails.

See Chapter 18 for more information on doing an in-place upgrade.

Install a new Exchange organization intending to coexist with or migrate from an existing messaging system

When Exchange is coexisting with other mail systems, such as MS Mail, cc: Mail, GroupWise, or Lotus Notes, additional connectors will be required to support the environment. Routing group connectors can only be used to connect Exchange routing groups within the same Exchange organization. Some connectors are not installed in a typical setup. So, you’ll have to choose them specifically in a custom mode installation. The two main functions of most connectors are to exchange messages and to aid in synchronizing the coexisting directories.
Protocols other than TCP/IP may be required in a coexistence scenario. For example, if you have cc: Mail post offices running on Novell file servers, you will probably need to install the IPX/SPX protocol in order to connect to those systems for message transfer and directory synchronization.

**Creating a test or pilot environment for one of the above scenarios**

A testing environment requires detailed planning and coordination in order to achieve meaningful testing results. Too often, testing is unplanned, lacks the proper resources, and does not add any tangible benefit to the implementation. Testing should be controlled, precise, and well thought-out. You’ll want to make sure that tests show clear results and reveal tangible information that is useful to the Exchange implementation effort. Even if you are installing Exchange to get a feel for it, don’t perform the installation and leave it at that. Take note of the new services that are added and the new file types that are used, and try to do some of the common administrative tasks by using the new administrative tools.

Often, setting up a proper test environment is the most difficult aspect of the testing process. If you are testing a particular feature of Exchange, or finding out if it will coexist in your existing environment, try to emulate your environment as closely as possible. Make sure you use the same software (including revisions and service packs) and hardware that is used in production. Due to hardware requirements, you may not be able to test the end-to-end system. In such cases, test individual components with the resources available and carefully document testing results.

A pilot is very different from a test, since it is, essentially, the beginning of a production implementation. Pilots tend to go in phases with server-side implementation and testing, followed by the workstation rollout, testing, troubleshooting, and documentation. It is crucial to think about the pilot in terms of a production system when doing your planning. Make sure you include backup and recovery planning, downtime scheduling, and a clearly communicated support plan when you do. You should have a pilot plan that outlines the roles and responsibilities of the people involved.

Once you are into the pilot, have change control policies and procedures firmly in place. Without control over requested and required changes, the roles of the team can get randomized, which in turn can cause your pilot environment to get randomized because your pilot team is responding to support issues and user functionality requests rather than continuing deployment.

**Preparing for the Installation**

Once you have the high-level decision making done and are ready to actually install Exchange, you’ll need to make a few additional preparations.
Hardware preparation and planning storage

Check to make sure that your server is on the Windows 2000 Hardware Compatibility List (HCL). The HCL is a list of Windows 2000 certified hardware that Microsoft provides on their Web site. Non-certified hardware may work just fine, but we have found that in many cases, using questionable hardware results in downtime. Find links to the Hardware Compatibility List on the CD that comes with this book.

Driver compatibility can be an issue with Windows 2000. If you have a standardized hardware platform that is known to be compatible with Windows 2000, you should have no problems. If not, make sure you check out any nonstandard hardware and make sure you obtain the most recent Windows 2000-compatible driver. Windows 2000 has been out for some time, but hardware vendors release new drivers on their Web sites periodically. Often, new drivers will optimize your hardware and fix known problems.

New drivers occasionally introduce new problems, but you still want and need to be current. You should only reinstall older drivers when you run into a problem.

Carefully plan your database and storage group strategy. If you have large mailbox servers, separate users into multiple databases. While this will increase administration slightly, using multiple smaller databases makes restoring databases much quicker. Smaller databases thereby reduce both user downtime and the scope of the application outage in situations where a single database needs to be recovered. At a minimum, place data with specific recovery requirements in a separate database.

Another technique you can use is to physically separate databases by placing them on different storage groups. Storage groups have their own transaction logs, and multiple storage groups can be backed up and restored in parallel. Combining this technique with the use of multiple drive controllers can be an effective way to increase performance and reduce exposure to hardware failure.

Each storage group runs under its own Store.exe process. Since each Store.exe process uses considerable server resources, using multiple storage groups will have serious performance implications. There is very little effect on performance if you add databases to a single storage group.

You will want to separate transaction log files from their databases onto physically different storage media. Having dedicated log file drives will increase server performance and make your system easier to recover in certain failure scenarios. A simple rule of thumb is that you should have a dedicated physical disk for each storage group’s transaction log.

Plan the server hard drive capacity to support the log files required during a restore scenario. Make sure you have enough space for a full week’s worth of log files plus the weekend full backup.
When a restore is not effective, it may be necessary to perform offline repair on the Exchange database. This requires free space roughly equal to the size of the database plus 10 percent. In order to avoid copying the data over the network, it may make sense to have multiple smaller databases and leave enough free space on the local server to perform the operation locally. You’ll want to make sure that the size of your databases or the availability of local disk space does not prohibit you from meeting your SLA for server recovery.

Physical installation media

You’ll want to make sure you have the correct version of the software, service packs, and license codes. There is nothing worse than traveling to another office, only to find out that you forgot the service packs software license code. Did you remember your antivirus and backup software? Did you remember to bring hot-fixes and driver updates for hardware?

Choose the right version of installation code for both Windows 2000 and Exchange 2000. Many countries have restrictions on encryption types. If you’re installing servers to be used abroad, find out if you need to install the international English version of the software to enforce these restrictions. Also, find out if you require a local language version of Exchange.

Consider whether or not you need the Enterprise version of Exchange for your installation. Table 20-1 illustrates some of the basic differences between the Standard and Enterprise Editions of Exchange 2000.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Standard Edition</th>
<th>Enterprise Edition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mailbox Store Size</td>
<td>16GB</td>
<td>Unlimited storage</td>
</tr>
<tr>
<td>Number of Mailbox Stores per Server</td>
<td>One (multiple application stores are supported)</td>
<td>Multiple (See Chapter 8)</td>
</tr>
<tr>
<td>Front-end/Back-end functionality</td>
<td>Cannot be front-end server</td>
<td>Can be front-end or back-end server</td>
</tr>
</tbody>
</table>

Don’t use “home-burned” CDs for copies of software. More than once, we’ve seen companies mistakenly install beta, evaluation, or limited license versions of software this way. Using “home-burned” CDs could also be a violation of licensing agreements. Suffice it to say that “home-copied” software can often prove to be unstable down the road and cause serious and embarrassing problems that you don’t want to expose your organization to. Helping your organization minimize risk is part of the job of a great Administrator.
Some companies successfully use a “Gold tape or disk”—a DLT tape, CDR, or something similar that includes all the approved and certified software required for an installation, as well as batch files and other scripts for automating the install process. Our suggestion about avoiding “home-burned” CDs refers only to ad hoc unauthorized copies in companies without sophisticated licensing management and copying policies.

Rather than carrying the physical media around, you may want to make the installation code available in a shared folder on the network. Installing “over the network” may cause unwanted network traffic, but in high-speed switched network server backbones, this is not normally an issue. The installation files on the Exchange 2000 CD are located in the I386\Setup folder.

**Windows 2000 preparations — local machine**

The choice of file system is important on an Exchange machine. In almost all scenarios, you should run NTFS on all of your volumes, including your transaction log drives.

Make sure you have all the required services installed. These services might include RRAS for remote users, Services for Macintosh if you will have Mac clients connecting to your Exchange machine, or Certificate server if you plan on running the Exchange Key Management service. However, be careful about running unnecessary services on your Exchange machine, which can slow performance considerably.

As mentioned in Chapter 16 on Quick Installs, the Internet Information Server must be running with the NNTP option before you begin an Exchange installation. We also recommend, for security and performance reasons, that you use a separate machine for remote RRAS connectivity.

Make sure you have all the required protocols installed and configured in Windows 2000. Use TCP/IP only if there’s a compelling reason to use multiple protocols. Not only do you need TCP/IP, but Exchange 2000 must be able to communicate with a DNS server for many of its functions. Make sure that there is access to good DNS resolution for the installation domain.

Once you have all the required Windows 2000 software on the server, install the latest service pack(s) and search for new hot-fixes and patches that might apply to your environment.

Service packs may need to be installed sequentially or may include earlier service packs and patches. Read the notes and readme.txt files to ensure you know exactly what you need to install and in what order!

Special accounts, installation rights, and permissions

Depending on the installation scenario, one or more security accounts will be required to complete the installation. It is recommended that you use purpose-made accounts.

One account you’ll want to create is the service account, which is used to run the Exchange services on the local machine. In order for the Exchange services on the local machine to communicate with other Exchange servers and Active Directory components, all servers in a routing group must have the same service account—a process similar to that found in an Exchange 5.5 site.

If you are installing into a site with Exchange 5.5 servers, you’ll use the existing site services account. Otherwise, you should create a special domain account that is only used to run Exchange services. When you create this account (for example, ExchService) in Active Directory Users and Computers, make sure you clear the “User must change password at next logon” check box. This account does not need special rights when you create it, as it will be granted those rights when it is identified in the setup process.

You will want to create a special account for installation and administration as well (for example, ExchAdmin), which must have local administrative rights on the install machine. The property page for creating an additional Exchange service account is shown in Figure 20-1. If you are installing in a forest with Exchange 5.5 servers, this account must also have full Exchange rights on the Site and Configuration containers in Exchange 5.5. If possible, make this account part of the Schema Admins and Enterprise Admins groups. When Exchange is installed for the first time, you are prompted for an Exchange administrative account to be granted full Exchange-related permissions. Once the install is completed, the administrative account that you specify is also able to delegate Exchange permissions to other accounts.

Figure 20-1: Creating the Exchange service account
The Exchange services account and installation accounts are granted wide-ranging permissions on the local machine and on the network. Create and use an internal “best practice” to keep these accounts secure.

If it is not possible to give Enterprise Admin and Schema Admin rights to the Install account, you will have to coordinate with the Enterprise Active Directory team to run ForestPrep. ForestPrep, which only needs to be run once, will perform the following actions:

✦ Extend the Active Directory Schema to include the object classes and attributes that are used by Exchange 2000.

✦ Add the organization object with the organization name you specify. Because you cannot change the organization name, make sure it is correctly specified during the ForestPrep installation. If you are joining an existing Exchange 5.5 organization, ForestPrep will create the organization object based on the existing Exchange 5.5 configuration information.

✦ Grant Full Exchange permissions to a named account. This allows the account to install Exchange throughout the forest and to delegate Exchange permissions to other accounts. Make sure you give the Enterprise Administrator the name of the account you have created for installations.

To prepare the forest for an Exchange installation, run the install from a command line with the /ForestPrep switch. For example, you would run F:\setup\i386\setup /ForestPrep where F: is your CD-ROM drive letter. ForestPrep should be run in the domain where the schema master computer is installed. Normally, this will be the forest root domain. A ForestPrep setup can take hours to complete on slower hardware, depending on the hardware and network conditions. Those hours may not include the time that the re-replication of the schema will take to propagate through your organization.

When you modify the Active Directory schema, the entire schema must replicate to every domain controller. Extending the schema is an irreversible process. You can disable object classes and attributes, but you cannot remove them.

**Active Directory considerations**

The Active Directory does not need to be completely rolled out to begin an Exchange 2000 installation, but you should have at least one Windows 2000 Active Directory domain implemented. The server you are installing Exchange on must be able to communicate with a Windows 2000 domain controller that is a Global Catalog server. As we mentioned earlier, the server must also be able to successfully communicate with a DNS server.

The Windows 2000 machine you are installing on cannot be part of a workgroup. The installation machine needs to be part of a domain prior to installation.
Do not install the Active Directory without first planning your forest and domain structure. The first Windows 2000 domain becomes the forest root domain, which is given special rights and roles that cannot easily be moved. Sometimes, companies create an administrative domain that is only used for enterprise administration (sometimes called a “dedicated root domain”). Make sure you read through Chapters 6 and 23, which cover topology and design, and the Windows 2000 Active Directory.

**Naming conventions**

Now’s the time that the careful planning you did for the Exchange naming standards comes into play. You should have already developed naming standards for organization, servers, routing groups, storage groups, contacts, public folders, and so on. Keep in mind that you have to get two naming standards right the first time: the Exchange organization name and Exchange server names. These cannot be easily renamed without removing and reinstalling the software.

Most other Exchange entities can be renamed through the standard management snap-ins, but it is better to define the naming conventions before you begin your implementation. Redundant work is avoided, and you will quickly get a sense of whether your naming standards make sense so you can adjust them if necessary. Aside from the Organization and Server names, objects that have naming restrictions include the following:

- Chat communities
- Mailbox stores
- Public folder stores
- Public folder hierarchies
- Policies
- Address lists
- Offline address lists
- Routing groups
- Instant messaging virtual servers
- Instant messaging user address
- SMTP message format domain display name

In general, plan on using fewer than 64 characters, and do not use special characters such as the following: [#;“", /<>+*]. Chapter 23 has an excellent section on developing sensible naming conventions.
Clustering considerations

How important is Exchange as a business-critical application? Is down time unacceptable? If a high-availability system is required, you will probably start thinking about clustered hardware. Exchange offers four-way active/active server clustering using the Windows 2000 Cluster service, as discussed in Chapter 19. Exchange clustering solutions have improved over the last couple years, but require special consideration for installation, maintenance, upgrades, and backup and recovery. For example, you will be required to install the Advanced Server version of Windows 2000 in order to use server clustering.

If you plan to install clustering, you need to ask yourself a few questions. Do you have the in-house expertise to support it, or are you relying on vendor support? Does the vendor have either the experience or the ability to work through a first-time clustering install with some deftness? Remember, a high-availability system needs to be set up and tested properly and must be maintained and monitored diligently in order to be effective. This requires a fair amount of expertise, so make sure you do not underestimate the resources required to support a clustered solution.

Coexistence with other mail systems

If you have to coexist with another system, you must prepare some components of your existing environment before you start installing Exchange 2000. More information on coexisting can be found in Chapter 18. Pay special attention to making sure your existing systems have the required software revision levels to support connectors.

Tip

Clean up the directories and directory replication/synchronization topology on existing systems before you connect Exchange 2000 and start replicating directory information. If you don’t, you will not have to re-replicate after subsequent repairs. Worse, changes in the dramatic topology could cause vast re-replication.

Licensing

Ask yourself whether you have the client and server licenses required to support your installation. You’ll need a single CAL (Client Access License) for each authenticated user connecting to the Exchange system, in addition to the Windows 2000 CALs. A single server license for each server you install is required. Of course, upgrading from Exchange 5.5 licenses will be cheaper than purchasing new licenses.
Performing an Installation

Once the preparation is done, the installation should be a snap. Most installation problems result from poor planning and preparation. For the purposes of this installation, we will assume that

- You’ve checked to make sure your server is on the HCL for Windows 2000.
- Windows 2000 is installed, including all the necessary protocols and services to support your environment.
- The most recent Windows 2000 Service pack is installed, including relevant hot fixes.
- The correct Exchange 2000 base installation media (known good) is available.
- The most recent Exchange 2000 service pack is available and installed, including relevant hot fixes.
- You’ve chosen an installation drive and names for the Exchange organization and server.
- Special user accounts have been created for the Exchange services and for logging on to perform the installation.
- A ForestPrep installation has been run naming the installation account as the Exchange administrative account (if the installation account you are logging on with is not a member of the Schema Admins and Enterprise Admins groups and this is the first time you’ve installed Exchange 2000).

Before you begin an Exchange installation, make sure your server is in a proper state for an Exchange install. Also, make sure you are logged in as the right account.

In Exchange 2000, administrative privileges will be given to the Windows 2000 user account you’ve used to log in.

Make sure all applications have been properly exited and that no users are connected to the machine, as the installation process can tax resources and stop and restart services.

The installation process is very simple once you’ve done all your planning. If you’re running from a network share, type `setup.exe` (with options) to initiate an installation. Figure 20-2 shows the setup options.
Performing a Smart Installation

Check out the options for creating unattended setup files and for running in unattended mode. We cover automation concepts later in the chapter.

If you plan on running from the CD, simply pop it in, and the Autorun feature will present you with the Install-shield setup.

Once you’ve started the installation, there are really no choices to make until you get to the Component Selection screen. To get there, click Next at the Welcome screen, Choose “I Agree” after reading the EULA (End-User License Agreement), and fill in the License Code that came with the CD. As noted in Chapter 16, generic license codes will not work, and you should never use anything but the license that came with the product.

We suggest you write the CD key on the CD with permanent marker so that if you lose the case, you can still install. Also, make a paper copy of the CD with the key and keep it with your licensing file.

At this point, you are presented with the Component Selection screen, where you will have to decide which components to install on your Exchange server. There are three modes of installation: Typical, Minimum, and Custom. Table 20-2 lists the components that are available for installation, along with a description of each component and the mode of install you can use to install the component.
<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Typical, Minimum, or Custom Installation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Exchange 2000</td>
<td>This component must be marked for the install action in order to install any Exchange components. Use the “Drive” drop-down to change the default location in which other components will be installed.</td>
<td>Typical, Minimum, Custom</td>
</tr>
<tr>
<td>Microsoft Exchange Messaging and Collaboration Services</td>
<td>This component includes basic Exchange messaging services such as the Routing Engine, the Information Store with related services, and the System Attendant. In order to install the Instant Messaging and Chat components, you'll need to install this component.</td>
<td>Typical, Minimum, Custom</td>
</tr>
<tr>
<td>Microsoft Exchange MSMail Connector</td>
<td>This component allows communication to a Microsoft Mail messaging system for coexistence and migration purposes.</td>
<td>Custom</td>
</tr>
<tr>
<td>Microsoft Exchange cc: Mail Connector</td>
<td>This component allows communication to a cc: Mail messaging system for coexistence and migration purposes.</td>
<td>Custom</td>
</tr>
<tr>
<td>Microsoft Exchange Connector for Lotus Notes</td>
<td>This component allows communication to a Lotus Notes messaging system for coexistence and migration purposes.</td>
<td>Custom</td>
</tr>
<tr>
<td>Microsoft Exchange Connector for Novell GroupWise</td>
<td>This component allows communication to a Novell GroupWise messaging system for coexistence and migration purposes.</td>
<td>Custom</td>
</tr>
<tr>
<td>Microsoft Exchange Key Management Service</td>
<td>This component integrates with a certificate server to provide advanced security (signatures and encryption) for your messages.</td>
<td>Custom</td>
</tr>
</tbody>
</table>
### Component Description

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Typical, Minimum, or Custom Installation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Exchange System Management Tools</td>
<td>This component installs the Exchange 2000 MMC snap-in for managing Exchange-related systems and services. It can be installed by itself on a system running any version of Windows 2000.</td>
<td>Typical, Custom</td>
</tr>
<tr>
<td>Microsoft Exchange 5.5 Administrator</td>
<td>Use this component to install the administration tool for managing Exchange 5.5 computers in an Exchange 5.5 or mixed environment.</td>
<td></td>
</tr>
<tr>
<td>Microsoft Exchange Chat Service</td>
<td>This component allows users to have real time, shared online communication framed by a chat client window.</td>
<td>Custom</td>
</tr>
<tr>
<td>Microsoft Exchange Instant Messaging Services</td>
<td>This component allows users to engage in real-time online text exchanges using an Instant Messaging client application.</td>
<td>Typical, Custom</td>
</tr>
</tbody>
</table>

Installation components are organized in a hierarchical fashion. A parent component has a plus/minus sign next to the component name. Sub-components appear below their parent, and the subcomponent name is indented relative to the parent. In order to install a subcomponent, you must choose the “Install” action on the parent component first. If you do not, you will be reminded with a message box.

To choose a different drive in which to install the files, click on the down arrow in the Drive column. Changing the drive letter for one component also changes the drive letter for all other components. Make sure you check the hard disk space statistics at the bottom of the dialog box to ensure you have enough space.

As we discussed in Chapter 19, if you want performance, try to map different storage groups to different physical spindles. Also, try to map your logs to a drive that is not the same as one with an information store.

Choose the type of installation you want. The Custom installation mode can be selected, as shown in Figure 20-3, and allows you to specify which components you wish to install.
Once you choose the components to install, click Next to choose the Organization options. In our case, we’ve decided to “Create a new Exchange organization,” as shown in Figure 20-4. Click Next to move to the next box.
You are now presented with a dialog box for entering the name of your new Exchange organization, as shown in Figure 20-5. If you decided to join or upgrade an existing 5.5 organization, your server will be added to your 5.5 organization. You may want to choose this option if this installation is part of your plan to upgrade your existing 5.5 installation. We choose to create a new organization, though for illustration purposes Figure 20-4 shows joining an existing organization.

The organization name cannot be changed without reinstalling Exchange.

[Figure 20-5: Entering the new organization name]

Make sure the name is typed in correctly, and click Next. You’ll be presented with the Exchange server license agreement, where you will agree to have a Client Access License (CAL) for every machine that has authenticated access to your Exchange server. Click Next, and you will see the Component Summary dialog box, as shown in Figure 20-6. Here, you get a chance to review the components you have chosen to install, and the installation drive.
Once you’re satisfied with your choices, click Next to begin the actual installation process. You will see a screen indicating the tasks as they occur and the overall progress of the installation. The phase where the Active Directory Schema is being updated may take considerable time. If you have prepared the Active Directory by using a ForestPrep installation, the installation will breeze past this step. However, if the Schema needs to be extended, it’s time to get a cup of coffee. Figure 20-7 shows the setup process during the Schema update phase. Schema updates can take anywhere from half an hour to two hours. That does not include the time it takes to re-replicate the Schema across your organization, but you can continue your install without waiting for that replication to complete.

When the installation is finished, click Finish and your installation is complete. You will want to survey some of the changes made to the environment and check to make sure your installation was successful.

**Note**

The process of installing a second Exchange server is much the same as installing the first. Since it will be picked up from the Active Directory, you will not have to specify an Exchange organization name. If there are multiple routing groups in the organization, you’ll have to decide which routing group to put the server into. Otherwise, the installation is the same.
Surveying Server Changes and Verifying the Success of the Install

Several items are changed and added to the machine during an Exchange installation. This section will survey some of the changes that are made and look at ways to verify that the installation completed successfully.

New folders and system software are added to the server during an install. You’ll notice that Exchange has added an M drive (or the next available logical drive letter). This is a virtual drive that corresponds with the installation drive you chose. If you look in this drive, you’ll find a folder that corresponds with the name of your domain (in our case, ilmarin.com). Inside the folder, you’ll find mailbox and public folder information.

If you look at the installation drive, you’ll see a `\Program Files\exchsrvr` directory that contains most of the Exchange system files. Some files are also added to other directories, such as the System32 directory.
Network Shares are added when you install Exchange. You’ll notice that the \Exchsrvr directory contains two folders that have been shared out. The tracking.log shared folder is used for message tracking, if enabled. The Address shared folder is used by the system to access .dll files for creating various address types, such as X.400 or SMTP. Figure 20-8 shows the additional network shares that appear after you’ve installed Exchange.

![Figure 20-8: Network shares created during setup](image)

The Exchange installation adds multiple services as well. If you look in the Services container on your server, you’ll notice several services that are Exchange-related. The exact services you have will depend on which Exchange components were installed. At the very least, you’ll notice things like the System Attendant, the Routing Engine (formerly the MTA), and the Information Store. If the installation was successful, all the Exchange-related services should be started. Figure 20-9 shows the Exchange services that were added during our installation.

Directory Schema changes have been made to accommodate Exchange object classes and attributes. A simple way to see this is by looking at the properties of a user account in Active Directory Users and Computers. You’ll now see an Exchange General tab that was not available prior to the install. Figure 20-10 shows some of the new Exchange objects and attributes that were added to user account properties in the Active Directory.
Figure 20-9: Examining the Exchange services

Figure 20-10: New User properties in the Active Directory
The easiest way to verify the success of your installation is by checking out the Event Viewer. Look in the application and system logs for any warnings or errors that might indicate problems with the installation. Figure 20-11 shows the application log of our Exchange Server after we completed the installation and the setup program automatically started the Exchange services.

![Image](image.png)

Figure 20-11: Examining the Application Event Log

Ways to Automate

If you are rolling out many Exchange servers for a large organization, it might make sense to automate the installation process. Automating your installations will benefit your organization by

- Drastically reducing resources (time and personnel) required for installations
- Ensuring a consistent server built on all machines in the environment

The trade-off is that you have to develop and test the automation process. There are several automation mechanisms that give you varying levels of control and
automation during an installation. The following sections discuss some of the tools and techniques that you might consider using to automate various pieces of your Exchange 2000 rollout.

**Unattended setup of Windows 2000**

A Windows 2000 installation can be automated by using the Unattend.txt file in combination with the “unattend” setup switch. The Unattend.txt file can be created manually or can be generated by using the Setup Manager utility available in the Windows 2000 resource kit. The Setup Manager is a GUI application that allows you to customize a variety of answer files, including those for Windows 2000 professional, Windows 2000 server, and Sysprep (described later in this chapter). Among other things, the Setup Manager allows you to

- Define computer-specific information, such as computer name, user name, organization, and time zone.
- Configure networking and related settings.
- Create distribution folders.
- Kick off application installations and run custom commands.

Once you’ve created a master unattended answer file, you can combine it with a /UDF file for defining a database of machine-specific information. This combination will allow you to completely automate the setup of Windows 2000 and most applications.

The syntax for an unattended setup will look like this: `Winnt(32) /unattend <answerfile>` where `<answerfile>` is the custom answer file you created to provide setup information to the installation. Using this method, you have a good degree of control over how much you will automate. You can provide some of the default information, everything except computer-specifics, or fully automate the process.

**Integrating Windows 2000 service packs**

Rather than run the Windows 2000 installation and then run the latest service pack separately, you can integrate both into a single installation folder. To integrate a service pack with the base installation code, do the following:

1. Simply copy the Windows 2000 installation folder (I386) with all of its subdirectories to a shared distribution folder you’ve created on a Windows 2000 machine.
2. If you’ve downloaded the compressed service pack, expand the service pack files according to the instructions. Locate the update.exe file.
3. Run `update.exe -s:<InstallShare>` where `<InstallShare>` is the shared distribution folder you copy the original install files to. For Example, `update.exe -s:C:\W2Kinstall`.

Now when you run the installation, the service pack is included.
Unattended setup of Exchange Server

It’s possible to use an Unattend.ini file to customize and automate the installation of Exchange 2000 Server. The Exchange Server uses the Unattend.ini file to control the installation, and it is based on the choices you make during the installation. The Exchange Server CD contains some sample files that you can use to develop an automated Exchange Server setup. Look in the \Support\Batsetup folder on the CD for the tools you need to create a custom batch script for installation.

You can also create a custom unattended answer file during setup if you use the /CreateUnattend switch during install. Unattended Exchange installations can be used very successfully in conjunction with Windows 2000 automation methods such as SMS, Sysprep, and unattended installs.

Using Sysprep and imaging

Sysprep is used for preparing a Windows 2000 machine to be cloned using imaging software, such as Symantec’s Ghost. It can be used to automate the core operating system, OS service pack, language packs, and some applications. In addition to automating, it speeds the installation time considerably, since much of the detection and file copy process is bypassed. The HAL, mass storage device controllers, ACPI (Advanced Configuration and Power Interface) support must be identical on cloned machines. Anything that is plug and play, such as video cards, network cards, and so on, will be detected and enumerated during the installation process.

You begin the cloning process by installing the core OS, service packs, and hot fixes on the master server. Next, install applications that you would like to have on all machines. Applications installed before disk duplication cannot depend on the Active Directory or run as a service. You can run setup of those applications, such as DCPromo, by using the [GuiRunOnce] section of the Sysprep answer file. Once you’ve prepared your machine, run Sysprep and then use a third-party disk imaging utility to create the installation image.

Once you have the image, you can work on creating custom answer files (Sysprep.inf) for the installation, which will supply the server name, time zone, domain affiliation, and so on. There’s also an option for automating application setup using the [GuiRunOnce] section of the Sysprep.inf file. This mechanism can be used in combination with the unattended mode of Exchange 2000 setup to automate much of the Exchange Server installation, but will require development and testing.

Tip

See your Windows 2000 documentation for more information on using Sysprep for disk duplication.
Client automation

There are many ways to automate the installation and setup of client software. Listed here are two of the more common ways to automate the deployment of your Exchange client applications.

SMS

Microsoft Systems Management Server is a popular application for deploying and managing software distributions, including Outlook 2000. Outlook 2000 can be deployed by itself or as part of an Office 2000 installation, but the process of using SMS for deployment is the same. The following high-level tasks are associated with an SMS deployment:

✦ Define computers that will be appropriate for the Outlook 2000 SMS deployment.
✦ Create a distribution package by using the source software and a package definition file.
✦ Distribute the package to installation sites by using SMS distribution points.
✦ Test the advertisement on a pilot group of machines and troubleshoot issues.
✦ Advertise the package to the appropriate production computers, initiating the installation.
✦ Monitor the progress of installations, user-installed hardware and further troubleshooting.

An SMS Administrator would want to hurt us for simplifying so much. If you are planning on using SMS for your Outlook 2000 deployment, we are assuming you have an SMS infrastructure in place and have personnel with SMS expertise available to assist with the rollout.

Active Directory distribution

The Active Directory can also be used to deploy Outlook 2000 and other applications to Windows 2000 machines. To advertise installable applications, Active Directory uses the Windows Installer, which is available on all Windows 2000 machines in conjunction with group policy.

Applications that will be deployed by using the Active Directory need to have Windows Installer-type package files. These files have the .msi extension, and work with the Windows Installer. The package is placed on the server in a shared folder, and a group policy object is given permissions to use the application. There are two basic ways to make an application available:
**Publish the application:** This makes the application available for install but does not add any shortcuts or menu items for the application. The application is not installed unless the user requests it. If the user is allowed to install the published application by group policy assignment, they can install it by using Add/Remove programs or by opening a document that requires the published application.

**Assign the application:** If an application is assigned to a user, the application shortcuts, menu items, file associations, and registry changes are added to the user’s machine, but the application code is not installed until the first time a user opens the application. If a user only uses 4 out of 12 assigned applications, a lot of disk space is conserved, since the other 8 applications are available but not actually installed. If an application is assigned to a machine, the application is installed the next time the machine is started.

There is a lot more to using the Active Directory, including version upgrades, self-healing applications, and clean software removal. Consult your Windows 2000 documentation for more information on using the Active Directory for deploying Outlook 2000.

**Summary**

Hopefully, it is now clear that a quick Exchange installation is much different from a smart installation. We have covered some of the best practices for ensuring success for your Exchange 2000 installation. You learned how to prepare for your installation, including preparations on the local machine and in the Active Directory. You were led through a sample install, saw what was added to the machine, and verified that the installation was successful. In addition, some of the tools and mechanisms you can use to help automate the deployment of your Exchange 2000 implementation were examined.
This chapter introduces the Exchange 2000 Conferencing Server. Have you ever considered what life would be like if you did not need to fly all over the world to have meetings? Wouldn’t it be nice if you could conduct or participate in a meeting right from your desktop? A meeting in which you could see and hear all the other participants and share information and documents? Conferencing Server is available today and makes it all possible.

What Is the Microsoft Conferencing Server?

Microsoft Conferencing Server is part of the Microsoft BackOffice collaboration platform. Exchange 2000 is the cornerstone of the collaboration platform; Conferencing Server is part of Exchange and a major new addition to the collaboration platform. This section explores the features and components of Conferencing Server. With Conferencing Server, you can schedule and conduct online conferences with live audio and video, and use additional features such as whiteboard drawings, chat, application sharing, and file transfer. You can structure online conferences by reserving conferencing resources, inviting conference participants, and setting the properties of each conference. Conferences can be public or private and can accommodate two or more attendees. The centralized reservation system of Conferencing Server enables you to schedule and join conferences from the Outlook calendar. If you are not an Outlook user, you can join conferences from a Web browser by using the conference location URL included in the conference request message.
Conferencing server features

In the past, online conferencing applications were difficult to use. The Conferencing Server has changed all that and offers the features described in the following sections.

Coordinating conference invitations

The Conferencing Server is fully integrated into Outlook 2000 calendaring. If you have Outlook 2000 and receive an invitation to a conference via e-mail, that e-mail invitation will include all the information necessary to connect each participant to the conference—time, date, and connection information. All you have to do is accept the invitation, show up for the virtual conference, and join in.

To connect, conference organizers and participants do not need information about the servers or the underlying technologies that the conference uses. Conferencing Server and Outlook 2000 together make connecting very user-friendly.

Finding a conference host

There are two different architectural models for conferencing: peer-to-peer and client-server. In peer-to-peer conferencing, conference participants establish direct links between their desktops; one or more desktops serve as hosts of the conference. A disadvantage of this model is that the conference host must be available for the duration of the conference. If the top-node conference host leaves, the conference ends.

In client-server conferencing, conference participants interconnect through a central server. A disadvantage of this model is that if the server is unavailable at the scheduled conference time, the organizer must schedule a new conference on another server.

Conferencing Server addresses both disadvantages by providing client-server conferencing with a load-balanced, best-connected server at the beginning of a conference. This approach eliminates the problem of server unavailability that existed in client-server conferencing applications, as well as problem of a host peer leaving the conference and terminating it.

Creating the conference topology

In the past, hosting successful conferences required you to understand conference topologies; you might have needed to know how networks deliver data and how to minimize the number of data copies sent over the network, or you might have needed to host the conference in more than one location and interconnecting the groups of participants.

Conferencing Server, by contrast, builds the topology according to the network location of the client at the time the client joins the conference.
Providing security and encryption

Even when you use Conferencing Server on a secure intranet, you may still want to ensure that only invited people can access the information discussed in the online conference. When scheduling an online conference from Outlook 2000, you can choose one of three levels of security: public, public with password, and private. You can access a public conference by typing the URL in the address window of your web browser. You can access a public-with-password conference by typing the URL of the conference in the address window of the web browser, after which you are prompted for a password. You can create a private conference by setting up the Conference Management Service to require that users authenticate themselves before accessing the conference.

Conferencing across a firewall

To meet with people online, you must share a common network such as the Internet. Even when you want to share information with others, you must still prevent unauthorized access to your company’s resources. Firewalls prevent unauthorized access, but they also prevent the flow of information in online conferences.

With Conferencing Server, you can host secure online conferences with attendees outside your organization. Conferencing Server is designed to make each conference attendee’s participation seamless, regardless of the attendee’s location on the Internet.

Even with seamless access, latency in transmission can present problems. Latency could reside in your Conferencing Server hardware but more likely involves your LAN, your connection to the Internet, your ISP, the Internet between you and the server, or any combination of these.

Controlling system and network resources

Most companies have a limited number of meeting rooms. Typically, meetings in these rooms are scheduled ahead of time, so the organizer knows that the room is reserved. Similarly, although online conferences do not require a physical location, they do place demands on the company’s network and server infrastructure.

The cost of a virtual meeting depends on the type of online conference you are hosting. An online telephone conference uses different resources from an online video conference. Conferencing Server bases its resource model on the status of the conference technology providers and their scheduled resources. Using the Conferencing Server resource model, you can define and manage the conferencing infrastructure to support the demands of online conferences within your organization.

Limit the use of Conferencing Server until your company recognizes its value. Once they do, upgrade bandwidth — on the box you use for Conferencing Server, on your LAN backbone where the Conferencing Server is, on your Internet connection, and so on.
Using Conferencing Server in your organization

Conferencing Server allows members of widely dispersed teams to meet online, removing geographical barriers and improving productivity in the process. Online conferences can build team interaction and encourage collaboration across great distances. Team members with specialized knowledge can participate in conferences on the other side of the country or on the other side of the world, without the expense and time required to travel to a physical location.

Scheduling a conference

You schedule a conference by creating a meeting request and inviting one or many conference resources. If the associated conference technology providers accept the conference request and successfully communicate their acceptance, Conference Management Service, a component of Conferencing Server, stores conference details in the conference calendar mailbox. Conference Management Service maintains acceptances, rejections, and changes to conference definitions. It processes updates and publishes them to the associated Exchange server free/busy information.

Scheduling conference resources

You can schedule a conference resource in two ways:

✦ Reserving a Conference Resource: You can directly reserve a conference resource using Outlook 2000. Outlook 2000 checks the free/busy information for the resource and adds the resource to the invitation if the resource is available. This procedure gives you a confirmed reservation. When you send an invitation, the conference resource is reserved and the URL for the conference is included in the invitations.

✦ Inviting a Conference Resource: If you are not using Outlook 2000, you can still schedule conferences and reserve conference resources. However, you must use an application that can send a meeting request, such as previous versions of Outlook, Outlook 2000 in offline mode, Exchange Server Version 5.0 client, or Outlook Web Access. Using one of these applications, you can send an e-mail with an invitation to the conference resource. If the resource is available, the application sends you an e-mail acknowledgment, accepts the reservation, and provides the conference URL.

Joining a conference

In a typical online conference, the user asks to attend by going to the location that the conference URL specifies. Every online conference is identified by a conference access URL. All conference participants use the same URL, regardless of the combination of conference technology providers the conference is using. A user can access the URL of an online conference before, during, or after the scheduled start time.
Conference Management Service receives the user’s request and attempts to find the associated conference in the conference calendar mailbox. After Conference Management Service locates the conference definition in the conference calendar mailbox, it determines whether a password or user authentication is required. Conference Management Service then works in conjunction with the conference technology providers to obtain the correct information for the user. During the conference, Conference Management Service permits the organizer to modify the end time and properties of the conference.

Joining a Video Conference

When a participant joins a video conference, a script runs on the client computer that investigates whether client configuration and network connectivity characteristics allow a direct connection to a multicast conference. If any of the prerequisites are not met, the Conference Management Service attempts to connect the participant to the conference over a unicast connection.

When a multicast connection is established, users can see a video of all participants, including any participants using unicast connections, and they can hear up to five active speakers simultaneously.

When a unicast connection is established, users can hear up to five speakers, but they can see only the current speaker.

Conferencing Server key components

Conferencing Server consists of three interrelated key components: Conference Management Service, Data Conferencing Provider, and Video Conferencing Provider.

Conference Management Service

The Conference Management Service (CMS) manages conference resources and supports conference technology providers. CMS provides access to online conferences through a Web site hosted by the Internet Information Services (IIS). As mentioned earlier, you can use CMS to configure a Conferencing Server to require user authentication on these Web pages, thus limiting access to private conferences.

Installing Exchange 2000 extends the entries in the Windows 2000 Active Directory to include entries specific for Exchange 2000. When you install the Conferencing Server, additional entries are stored in the Active Directory. In particular, user mailboxes are enabled for conferencing; that information is maintained and propagated in the ADSs database. CMS uses the Exchange 2000 free/busy public folder to publish availability of conference resources. The Outlook 2000 client accesses the free/busy folder for information on the availability of conference resources at specific times (for example, when it is accepting an invitation).
When you schedule a conference, CMS creates a URL that attendees use to access it. CMS stores all scheduled conferences in the conference calendar mailbox. This information is used to create a persistent representation of the format, structure, and any additional information associated with the conference. CMS also allows e-mail clients that cannot reserve conference resources to invite them anyway.

To conserve and manage resources, CMS controls the lifetime of conferences, working with each conference technology provider to ensure that the conference starts and ends on schedule.

CMS keeps a list of all scheduled conferences. Public conferences are listed on the conference access pages, and any user can participate. Private conferences are not listed; to join a private conference, participants must have the correct URL.

CMS allows the Conferencing Server Administrator to analyze both schedule and conference data in formatted, comma-delimited, audit log files on the server on which Conference Management Service is running.

**Data Conferencing Provider**

Data Conferencing Provider is a conference technology provider that enables conference participants to share applications, conduct whiteboard sessions, transfer documents, and chat. Data collaboration can be accomplished using tools such as Microsoft NetMeeting or other applications that support the T.120 network communications standard.

**How does a Data Conferencing Provider work?**

A Data Conferencing Provider creates a resource scheme based on its maximum permissible number of simultaneous conference-participant connections. This number is the physical resource against which the conferencing resource makes reservations when you invite it to a conference. Each conferencing resource has a size associated with it; Conference Management Service counts this size toward the cost of hosting the conference. A Data Conferencing Provider subtracts the resource cost from the maximum participation count. If the remaining available maximum participation count is greater than the resource cost, the resource is reserved for the time you request. If the available count is less than the size of the resource for any specific requested time, the Data Conferencing Provider prompts the Conference Management Service to publish a busy indication, and no additional reservations are accepted.

During a data conference, the each participant’s computer is connected to a T.120 multipoint control unit (MCU). On the conferencing site, you can install the T.120 MCU on multiple servers running Microsoft Windows 2000. The Conferencing Server groups MCUs to provide fail-over and load balancing across the conferencing site. Groups of MCUs provide the platforms on which each scheduled data conference is hosted. When a participant joins a data conference, a Data Conferencing Provider considers the following criteria to select the best MCU:
Which MCUs are available and in service?
Is the conference already running on an MCU?
What load do existing conferences place on the MCU?
Are there any administratively defined restrictions on which MCU can be used?
What is the network location of the participant?
Is the participant inside the local conferencing site?
Is the participant inside a site that is administratively defined as a local site?
Is the participant outside the organization’s network?

Using these criteria, a Data Conferencing Provider connects a conference participant to a load-balanced MCU that is closest to his or her network location, minimizing the number of data copies sent between these locations. Because a Data Conferencing Provider assigns an MCU when a participant joins a conference, the interconnection of MCUs in the conference is always dynamic and can optimize server availability.

You can install T.120 MCUs on any Windows 2000 Server, independent of other services. Each MCU automatically attempts to retrieve a machine certificate from Windows 2000 Certificate Services. The MCU can host private conferences only with valid certificates.

There are two different configurations for an MCU: peer-to-peer and client-server. In a peer-to-peer conference, such as a NetMeeting application with ten participants, ten copies of the information have to be forwarded across the network. If the participant is on a modem or at the other end of a slow WAN link, the response can be slow.

To avoid clients sending multiple copies of the conference data, the MCU can be located on a central server and each participant can connect to it. In addition, if the MCU server can bridge multiple networks, a conference can take place between participants on your intranet and participants on the Internet. When the conference is hosted on a server, the organizer is not required to remain in the conference until it ends.

A data conference has the following features: shared applications, shared clipboard, file transfer, whiteboard, chat, and an MCU.

Shared applications
You can share a program running on one computer with other participants in the conference. Participants can review the same data and see the actions while the person sharing the application works on the program; for example, you can edit a Microsoft Word document. The person sharing the application can choose to collaborate with conference participants, allowing them to take turns editing or controlling the application. Only the person sharing the program must have the application installed on his or her computer.
Shared clipboard
You can share data with other participants by using cut, copy, and paste operations. For example, you can copy information from a local document and paste the contents into a shared application. The shared clipboard provides an easy way for participants to exchange data between shared and local applications.

File transfer
You can send a file in the background to one or more conference participants using a file transfer. Because the file transfer occurs in the background, participants can continue sharing an application or using other data conferencing features.

Whiteboard
You can load or sketch diagrams and organizational charts, or display other graphic information in this multi-page, multi-user drawing application. Since Whiteboard is object-oriented (versus pixel-oriented), you can move and manipulate the contents using a click-and-drag operation. You can use a remote pointer or highlighting tool to point out specific content or sections of shared pages.

Chat
You can type text messages to share common ideas or topics with conference participants, to record conference notes and action items, or generally to communicate if audio is not available. The NetMeeting whisper feature allows you to have a separate, private conversation with one of the conference participants during a group chat session.

NetMeeting 3.01 or later is the recommended client software for a Data Conferencing Provider. It can run as an embedded control on the Web page, and it facilitates private, secure conferences.

Multipoint Control Unit
Multipoint Control Units (MCUs) are required for data conferences. MCUs interconnect multiple conference participants and synchronize and distribute conference data between participants. MCUs are the glue that holds a conference together.

Video Conferencing Provider
Video conferencing is a technology that allows two or more people in remote locations to exchange voice and video images in real time. A Video Conferencing Provider allows users to organize and participate in multiparty video and audio conferences. Video Conferencing Provider technology is based on the Internet Engineering Task Force (IETF) IP multicast standards.

How does a Video Conferencing Provider work?
When a conference with a Video Conferencing Provider resource is scheduled, a special multicast IP address is requested from a Multicast Address Dynamic Client Allocation Protocol (MADCAP) server and reserved for the conference. MADCAP, which is part of the Windows 2000 Dynamic Host Configuration Protocol (DHCP)
service, assigns a multicast group IP address to each videoconference. This multicast IP address cannot be used in subsequent reservations until the initial reservation expires.

The expiration time is defined as the scheduled conference end time plus the maximum videoconference extension time. You can configure the maximum videoconference extension time from the Video Conferencing Provider in Conferencing Manager. The maximum videoconference extension time is the maximum amount of time (in minutes) that the conference organizer can extend a conference in progress.

As a multicast client joins a Video Conferencing Provider conference, the ActiveX control accepts the multicast IP address reserved by the Video Conferencing Provider for the conference. The control then uses that IP address to subscribe to the multicast conference. Once subscribed, it retrieves and sends video and audio data.

In a pure multicast videoconference, no servers participate in the actual transfer of video and audio traffic. If routers are in the way of the multicast packets, they serve as replicators or multiplexers, which forward packets to the network segments on which the clients participating in the conference are located.

**MADCAP**

For a multicast videoconference to work, servers running MADCAP in Windows 2000 must be present on the network, and routers must be multicast-aware. Servers running Dynamic Host Configuration Protocol (DHCP) in Windows 2000 can assign multicast and unicast addresses. You can configure multicast scopes and corresponding multicast IP ranges with the DHCP Administration Tool.

**Tip**

Spend some time with the WAN people in your organization when you pilot and build your deployment plan for Conferencing Server to ensure that your routers and WAN bandwidth are up to the task.

**Automating Conferencing Topology**

The single-server deployment described in this chapter does not address the issue of multiple participants on the remote end of a WAN link, because the deployment still requires multiple copies of the conference data to be transmitted across the link. In this scenario, you could place an MCU at both ends of the WAN link and instruct participants to connect to the MCU closest to their physical locations on the network. Although this increases the complexity of participation in a conference, you can build a conference topology in this manner using peer-to-peer logic only. However, if you stretch the conference over multiple WAN links, and if multiple participants often travel, changing which MCU is closest, this manual solution becomes impossible to manage. Data Conferencing Provider solves these problems by creating an automatic topology that sends single instances of data across the WAN link. A persistent location for the conference allows participants to come and go. Data Conferencing Provider automatically directs a client to the closest server and also provides a high level of security.
**Auxiliary components**
Other items support the key components of Conferencing Server: Outlook calendar, conference window, conference calendar mailbox, and conference access pages.

**Outlook calendar**
You can schedule online conferences using Outlook 2000. You must invite a conference resource (the virtual meeting room in which the conference is held) for the meeting. Outlook 2000 reserves the conference resource before sending prospective attendees conference invitations. A link (URL) to the online conference is included in the invitations that conference participants receive. To join the conference, attendees can access this link with a browser. Outlook 2000 users receive a conference reminder that includes a Join Conference option, which users can click to join the conference. They can also join a conference by right-clicking the meeting in their Outlook calendar and choosing Join Conference.

**Conference window**
When a user joins a conference, the browser displays a conference window, customized according to the conference technology providers used and whether participants access the conference before, during, or after the scheduled time. The left frame of the conference window, the conference panel, displays general information about the conference. The conference window includes one or more additional frames, each of which is associated with a specific conference technology provider. The conference panel includes

- Conference schedule and amount of time remaining in the conference
- Mail-enabled link to the conference organizer
- Subject of the conference
- Conference privacy (attendance is restricted to the invitees)
- Access to online help for conference access pages
- Refresh button (you can refresh your window with the latest conference data)

The conference organizer can access two additional features: the ability to request an extension to the length of the conference in session and the ability to request to change the resource on which the conference is hosted (to allow additional participants to join the conference).

The conference window displays a warning to participants five minutes before the end of the conference. At the end of the five-minute period, participants are disconnected from conference resources unless the Administrator configures a grace period and the conference technology providers honor it. Grace periods allow attendees to arrive early or leave late. Grace periods apply to all scheduled conferences on the site.
Conference calendar mailbox
Each conferencing site requires a conference calendar mailbox assigned to CMS. You must assign the mailbox before a conference organizer can schedule an online conference using conference resources.

For better performance, place the conference calendar mailbox on a server running Exchange that is as close as possible to the conferencing server hosting CMS — on the same server if possible.

Conference access pages
Associated with each CMS is a Web site that hosts the conference access pages. All conference participants access these pages, which also include a list of links to public conferences. Users can select one of these links to attend a conference.

Because all conference participants must retrieve pages from this Web site, the site’s network address must be accessible to all users. If some users attend from the Internet, the conference access pages must be available over the Internet as well.

Figure 21-1 shows how the components discussed in this section relate to each other and provide the conferencing service.

Security
Conferencing Server provides security when it controls access to conferences and when users participate in a conference. CMS controls the security for creating and accessing conferences.

When you schedule a conference, you control access to it by specifying whether the conference is public or private. With Conferencing server, you can create the following types of conferences:

✦ **Public Conferences**: These are accessible to any user who has access to the URL. By default, conferences are public.

✦ **Public Conferences with Password**: You can create a public conference that participants can access with the correct password. The conference technology provider uses the password to further limit access. To create a public conference with a password in Outlook 2000, select the Allow external attendees check box and include the password on the conference invitation.

✦ **Private Conferences**: You can create a private conference so that only attendees you invite can participate. CMS, the conference technology provider, or both check the credentials of each participant for authorization to participate in the conference. If CMS performs the check, it uses IIS to check the user’s credentials. A digital certificate — an electronic document that grants credentials to the users or computers on the network — is required for private conferences.
In this section, you will learn how to configure, plan, and install a Conferencing Server and its components. Before you install a conferencing server, you must verify the system requirements and permissions. After you verify that your configuration meets the requirements, you must plan your Conferencing Server deployment. If you are satisfied with your conference site plan, then you can install the Conferencing Server and its clients.
System requirements
In this section, you will be introduced to the hardware and software requirements for both conferencing servers and conference clients.

Hardware requirements
This section explains the hardware requirements for servers on which you can install a Conference Management Service, a Data Conferencing Provider, and a Video Conferencing Provider. You will also learn about the hardware requirements for clients that will participate in data or videoconferences.

Servers
The recommended hardware requirements for servers on which you install Conference Management Service, Data Conferencing Provider, Video Conferencing Provider, or MCUs are as follows:

✦ 400 MHz Intel Pentium processor (or equivalent)
✦ 256MB of RAM

A server with this dedicated configuration can handle approximately 500 simultaneous user connections. But more powerful hardware is advisable. Consider all aspects that can choke performance: processor speed, ability to add a second processor, high-speed LAN, high-speed SCSI or EIDE (ATA 100), amount of memory, fast video, and so on.

Clients
There are two types of conferencing clients: data and video. For a data conferencing client, the minimum hardware requirements are as follows:

✦ For a Windows 95 client, an Intel Pentium 133 processor or better (or equivalent) with at least 16MB of RAM
✦ For a Windows NT or Windows 2000 client, an Intel Pentium 133 processor or better (or equivalent) with at least 32MB of RAM

For a video conferencing client, Microsoft recommends the following hardware requirements:

✦ 266 MHz Intel Pentium processor (or equivalent) running Windows 2000 Professional
✦ 128MB of RAM
✦ LAN connection
✦ Sound card with microphone and speakers
✦ Video capture card and camera
Use hardware as powerful as you can afford. When asked, “When is a good time to lime your fields?” a farmer will reply, “Whenever you have the time and the money.” The same goes for having more hardware for applications like this. It will run (with nothing else running) on the minimum configuration, but you will appreciate its power more if you have higher performance—part of which comes from the hardware.

**Software requirements**

This section details the software requirements for servers and for clients.

**Servers**

To install Conferencing Server, the following services and components (some of which come from Windows 2000) must be available on the machine where the installation is taking place:

- Microsoft Exchange 2000 Server
- Microsoft Windows 2000 Server
- Microsoft Internet Information Services (IIS)
- Multicast Address Dynamic Client Allocation Protocol (MADCAP)
- Certificate Services
- Active Directory Service.

---

**Additional Conferencing Server Requirements**

The Conferencing Server requires at least one server running Exchange 2000 Server on the same domain. You can install Exchange Conferencing Server components on the same server as Exchange or on another computer in the same domain. In addition, at least one of the servers running Exchange on the domain must include or have a replica of the Schedule+ Free Busy Information public folder.

The Exchange Conferencing Server can be installed only on a server running Windows 2000 Server or Windows 2000 Advanced Server.

The MADCAP service, which is a part of the Dynamic Host Configuration Protocol (DHCP) service configuration on Windows 2000, must be running on the domain. The MADCAP is used to configure Internet Protocol (IP) multicast address scopes for videoconferences.

To support secure data conferences, the Certificate Services must be installed (on at least one computer) in the Active Directory forest that contains the T.120 multipoint control units (MCUs) for secure conferences.

To host conference access Web pages, you must have IIS installed on the same site as a server running Conference Management Service.

To support conference resource objects and configuration objects, you must have Active Directory installed on your domain.
Clients
Users who organize conferences and want to participate in conferences must have Microsoft Windows 2000, Outlook 2000, NetMeeting 3.01 or higher, and Internet Explorer 5.0 or some other Internet browser (frames-capable, able to download ActiveX controls, and JavaScript-enabled).

Conference participants can have full access to all the conferencing features if they meet all the system requirements.

Depending on their role, the minimum requirements for clients are as follows:

- **For a conference organizer:** An e-mail application with a calendar, such as Outlook 97 (or later) or Outlook Web Access.
- **For a data conferencing client:** A frames-capable Internet browser with JavaScript enabled (such as Internet Explorer 4.01 or Netscape 4.5), a T.120-compliant client application (such as NetMeeting), and an operating system that supports your Internet browser and T.120-compliant application.
- **For a video conferencing client:** A frames-capable Internet browser that supports ActiveX controls with JavaScript enabled (such as Internet Explorer 4.01 or Netscape 4.5), NetMeeting 2.1 or later (if you are not on a multicast-enabled network), and an operating system that supports your Internet browser and T.120-compliant application.

A client who accesses an online conference from a computer with Microsoft Windows 2000 on an IP-multicast enabled network receives a multiparty video image of the conference participants. These video images are embedded on the conference Web page by using an ActiveX control.

Permissions
Each role involved in administering a conferencing server has a permission associated with it:

- **Managing Exchange Conferencing Server:** This role, which manages and configures site-wide conference settings, Conference Management Service, Data Conferencing Provider, Video Conferencing Provider, and all MCUs must have read and write permissions on the conferencing site.
- **Exchange Conferencing Server services:** This role, which starts and stops services, such as Conference Management Service, MCU service, or H.323 service, must be a member of the Domain Admins group or logged on with the domain administrator account.
Managing resources: This role, which can install the Conferencing Server components, and create and manage the conference calendar mailbox and any conference resources, must have permissions to create domain user accounts.

Planning a Conferencing Server site
The number of conferencing servers that your organization needs, and where to install them, depends on the network topology and demand for online conferencing. Several factors will help you decide how many conferencing servers you would need and where to install them.

Anticipated schedule load
Based on existing network patterns, you can decide on how your organization will use the Conferencing Server. For example, if you have e-mail servers at each of several data centers, you can install the Conference Management Service on each server.

Logical separation of conferences
Based on how you think your organization will use online conferences, you can create different sites for different types of conference. For example, you could create a conferencing site for conferences that are published to the Internet.

Multisite conference topologies
Since only one instance of a Conference Management Service is active on a Windows 2000 site, you can balance your user load across multiple sites by assigning users to sites according to their geographic location.

Single conferencing site
For an initial or pilot deployment, you can install all conferencing services on a single computer. If you create a virtual network address for the conference access pages, you can then expand this pilot deployment to meet the needs of your organization.

The first CMS installed on a Windows 2000 site becomes the active host server for the conferencing site. You can install additional instances of CMS on other servers, but only one at a time is active on a site.

A single conferencing site with conference access pages installed on the default host server is adequate for most companies. However, very large companies that place a heavy demand on Conferencing servers can designate a single computer or group of computers to process conference access requests.
Multiple conferencing sites
If your organization has groups of people in more than one geographic location and more than one Windows 2000 site you can create additional conferencing sites. Because each conferencing site has its own conference calendar mailbox, you must create conference resources and configure network addresses for the conference access pages for these additional sites. Creating additional conferencing sites helps balance network load and improve performance.

Deploying Conferencing Server best practices
The following is a list of best practices that you should follow when you deploy a Conferencing server:

✦ Network topology: For conferences with people outside your organization, place the IIS service on a server whose DNS host name can be resolved by both internal and external clients.
✦ Conference Calendar Mailbox: Keep the conference calendar mailbox on the same server as the active CMS.
✦ Limit conflict: The conference resources and the conference calendar mailbox should be the only mailboxes on the server running the CMS.
✦ Sites with multiple subnets: If your conferencing site has multiple subnets, place at least one MCU on each subnet.
✦ Phased rollout: Consider a phased rollout for the Conferencing Server. Deploy the service on one server and give access to a single group. Then add additional servers and groups to meet demand.

Installing and configuring Conferencing Server
You must take care of several tasks before you install a Conferencing server. This section outlines those tasks.

Verifying user account
The user account that you are logging on with must be a member of the local computer Administrators group and must be one of the following.

✦ Member of the Enterprise Administrators security group for all domains
✦ Member of the Domain Administrators security group for the local domain
✦ Member of Conferencing Administrators group
Creating a Conferencing Administrator
You can create a Conferencing Administrator user or group for people to use to install, configure, and administer Exchange Conferencing Server. Use this account as an alternative for users whose accounts are not members of Enterprise Administrators or Domain Administrators.

Delegating administrative permissions
You can use the Administration Delegation Wizard to delegate permissions to Administrators. If the account that installs Exchange Conferencing Server is not a member of the Enterprise Administrators or Domain Administrators security group, delegate the administrative permissions for the Exchange Full Administrator role to the account at the organization level.

Enabling computer and user accounts to configure trust for delegation
As part of installation, the Conferencing Server checks the rights on the account that is installing it. If the account has sufficient rights, the Conferencing Server sets the trust for delegation on the server. The trust for delegation setting allows any process or service running on the local server to request additional services on behalf of a remote client.

Use the following procedure to enable Exchange Conferencing Server to set the trust for delegation on the server. First, make sure that the hardware and software is available and configured correctly. Then install the Conferencing server. Figure 21-2 shows the initial screen for the installation process. You start the install process by calling Launch.exe.

![Initial Exchange 2000 Server Conferencing Server screen](image)

Figure 21-2: Initial Exchange 2000 Server Conferencing Server screen
After the initial screen, the following two screens, shown in Figures 21-3 and 21-4, prompt for the license agreement and for product identification.

Figure 21-3: License agreement

Figure 21-4: Product identification

After you enter the product ID, you are prompted to decide whether you want to have a complete installation or a custom installation, as shown in Figure 21-5.
Normally, you should select Complete. If you select Custom, several options present themselves, as shown in Figure 21-6.

Table 21-1 describes the components listed on the Custom Setup screen in Exchange Conferencing Server Setup.
### Table 21-1
**Custom Installation Components and Their Function**

<table>
<thead>
<tr>
<th>Component</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conference Management Service</td>
<td>Accepts reservations for online conferences, maintains conference schedules, and controls conference technology providers. When you install this component, Exchange Conferencing Server installs Data Conferencing Provider and Video Conferencing Provider.</td>
</tr>
<tr>
<td>T.120 MCU/H.323 Conference Bridge</td>
<td>Provides an MCU for T.120 data conferencing clients and a bridge that allows H.323 clients to participate in audio and videoconferences.</td>
</tr>
<tr>
<td>Conference Access Web Pages</td>
<td>A set of Web application pages hosted by IIS. Conference participants use these pages to participate in online conferences.</td>
</tr>
<tr>
<td>Conferencing Manager</td>
<td>A Microsoft Management Console tool for managing and configuring conferencing services.</td>
</tr>
</tbody>
</table>

After you select the appropriate components, select an administrative group, as shown in Figure 21-7.

![Figure 21-7: Selecting an administrative group](image)
The default administrative group is First Administrative Group.

The administrative group is selected for you if CMS is already installed on the site or if there is only one administrative group in your Exchange organization. Once CMS is installed in an administrative group, subsequent instances of CMS are installed in the same administrative group.

The final step is to invoke Install, as shown in Figure 21-8.

![Figure 21-8: Ready to install Conferencing Server](image)

After the initial installation of Exchange Conferencing Server, you can install additional copies of Exchange Conferencing Server components on other servers in your organization. For example, you can install additional copies of CMS and Conferencing Manager and create additional MCU servers. The procedure is the same as for a custom installation.

You can install MCUs on a Windows 2000 site without Exchange Conferencing Server installed. If you do, you must select another site with an active Conference Management Service to act as the conferencing host server for the MCU.

**Post-installation**

After you install Conferencing Server, you must perform certain tasks on the server and on the client computers. This section outlines these tasks.

**Server**

After you install Exchange Conferencing Server, perform the following tasks to prepare your conferencing site for online conferences.
Mailboxes
Create the following mailboxes for your conferencing site:
- Conference calendar mailbox
- Conference resources

Exchange Conference Resource mailboxes and user mailboxes on different servers
Exchange Conferencing Server uses free/busy information for scheduling conference resources. If the conference resource mailboxes and mailboxes for users who schedule and participate in conferences are on different servers, you need to do one of the following:
- Replicate free/busy information between the server with the conference resource mailboxes and the server with the user mailboxes
- Change the default public store on the mailbox store for each server with user accounts to a shared public folder store

Client
After you install Exchange Conferencing Server, create a conference calendar mailbox, where the information for all scheduled online conferences on a conferencing site is stored. You can use the conference calendar mailbox to schedule conferences. Before you can schedule conferences, you have to perform several tasks to prepare your users to participate in online conferences:
- Enable Microsoft Outlook 2000 users to schedule conference resources. In Outlook 2000, in the Calendar, choose Tools, and then choose Options. On the Preferences tab, choose Calendar Options, choose Resource Scheduling, and then choose Set Permissions. On the Permissions tab, in Calendar Properties, set the permission for the users that can schedule conferences.
- Provide certificates to your users for secure conferences.

Conferencing Server Administration
This section explains how to administer the Conferencing Server using the Exchange Conferencing Services snap-in. You manage each conferencing site within the organization separately.

CMS defines the top level of the Conferencing Server installation. Install each instance of a CMS on a computer on a Windows 2000 site. Only one instance of CMS is active on each site. An additional instance on a site is idle and acts as a backup host server for the conferencing site. The host server you configure defines which
server is active; the first instance of CMS that you install on the site is the default host server. The following steps will allow you to administer a Conferencing Server:

1. On the Start menu, choose Programs, choose Microsoft Exchange, and then choose Conferencing Manager.
2. In the console tree, right-click Exchange Conferencing.
3. Choose Manage; in the scroll box, select the name of the site you want to manage. Then click OK.
4. In the details pane, double-click <Site Name> Conferencing Site, where <Site Name> is the name of the conferencing site.

At this point, you can administer CMS, MCUs, conference resources, and conference technology providers.

**Administrative groups**

When installing CMS, you must specify the Exchange administrative group in which to install the service. These groups allow you to assign administrative permissions to specific users who then control different servers or services. When you install CMS from a specific Windows 2000 site in an administrative group, all that you install in that site are placed in the same administrative group; therefore, you can always manage the servers on a specific site from a single administrative group.

**Conference technology provider service management**

You can control each service installed as part of the Conferencing Server from the Conferencing Manager, Windows 2000 Services, or System Manager that is on the same domain.

**Logging reports**

CMS produces a report log for a number of operations. You can specify the events that you want included in the log using the following procedure:

1. On the Start menu, choose Programs, choose Microsoft Exchange, and then choose Conferencing Manager.
2. In the console tree, right-click Exchange Conferencing.
3. Choose Manage, and in the scroll box, select the name of the site you want to manage; then click OK.
4. In the console tree, double-click Exchange Conferencing.
5. In the details pane, right-click <Site Name> Conferencing Site, where <Site Name> is the name of your conferencing site, and choose Properties.
6. Click Logging, and select the events you want to include in the log.
Each event is written to a comma-delimited text file named for the date it was generated. These files are automatically purged according to the same schedule you defined for the Exchange 2000 message log files.

The log files can be found on the server running Conference Management Service in the directory \<servername>\<servername>.ecs.

**Summary**

Conferencing Server adds real functionality at relatively low cost to your company. For knowledge-workers and virtual teams, it is a great tool. The product has evolved to the point where resources, installation, and administration are well within the command of most Administrators and organizations. Refer to this chapter to learn about the product and about how to get it going, install it, and administer it.
Securing Your System

Protecting the corporate record is one of the mandates of all Exchange Administrators. The method to protect the corporate record that we focus on in this chapter falls under the heading of security. We tend to think of security as a layered process: you need to secure the passage of data as it transmits, be sure that all connections to the system come from appropriately authenticated users or services, provide ways that the data content can be secured, and finally, protect from viruses. To provide these capabilities, Exchange 2000 relies heavily on Windows 2000 features and services, adds some services and features of its own, and is augmented by third-party products.

Windows 2000 Security

The Windows 2000 security subsystem has been totally revamped with the introduction of the Active Directory Service (ADS), which replaced the Security Accounts Manager (SAM) in Microsoft Windows NT 4.0 and 3.x as a security database. It also replaced Exchange 4’s Directory and three-part security context model.

Active Directory Service

The Active Directory Service is part of the Windows 2000 security subsystem. All objects in ADS are protected and secured by Discretionary Access Control Lists (DACLs). Every access to any object or attribute in the Active Directory by
any user or service is validated against the DACL. There are also system-generated internal control lists called SACLs (System Access Control List). These provide rights and permissions for system functions. Every object has a SACL, DACL, or both.

When Administrators delegate permissions or rights to an object, they are writing Access Control Entries (ACEs) into the object’s DACL. The ACEs consist of a unique identifier for the user who is being granted permissions, along with the specific right or permissions to the object. Security Identifiers (SIDs) are used to designate the principals. In Exchange 2000, Security descriptors exist on folders and items. They control access to the folders and are used in the inheritance of subfolders. There are also security descriptors on Mailboxes that define mailbox control rights (such as deleting storage and reading or changing permissions). An example of administrative security descriptor controls would be the administrative functions on a folder to mail enable it, set retention control, expiration, and so on.

The Windows 2000 security system uses the Active Directory to store security policies and account information. It also implements security models for all objects and authenticates the access to any Active Directory object or attribute. Exchange 2000 modifies the ADS schema by adding attributes for recipients, e-mail address support, and Exchange configuration.

The Local Security Authority (LSA) is a Windows 2000 protected subsystem with the job of providing local computer security. LSA acts as a guardian to ensure users have appropriate access permissions to gain access to resources and objects on the local system. LSA is made up of several components that, together, manage authentication. LSA components are provided in Table 22-1.

<table>
<thead>
<tr>
<th>LSA Component</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net Logon service</td>
<td>Validates user credentials against a domain controller (DC), returning domain SIDs and user rights for the user.</td>
</tr>
<tr>
<td>NTLM authentication protocol</td>
<td>Authenticates clients that do not use Kerberos authentication.</td>
</tr>
<tr>
<td>Kerberos Key Distribution Center (KDC) service</td>
<td>Responsible for granting tickets to clients.</td>
</tr>
<tr>
<td>Secure Sockets Layer (SSL) authentication protocol</td>
<td>Provides authentication over an encrypted channel (alternative is less secure clear channel).</td>
</tr>
<tr>
<td>Kerberos V5 authentication protocol</td>
<td>Owns responsibility for Kerberos ticket authentication.</td>
</tr>
<tr>
<td>LSA server service</td>
<td>Security policy enforcer.</td>
</tr>
</tbody>
</table>
## Access control

Once LSA authenticates a user, the next step is to gain access to the network. Here again, the Windows 2000 security system is responsible for control. It owns all access to resources and objects on the network, including those in Exchange. Windows 2000 Server controls access using security descriptors, which are assigned to objects and stored in the Active Directory Global Catalog servers. (If you only have one domain, all domain controllers are effectively Global Catalog servers.) Global Catalog servers maintain a copy of the ADS that includes all information about the domain it is part of (after accounting for latency of replication) and stubs for all domains in the forest. In multi-domain environments, GCs are the only servers that have information about universal groups.

An easy way to think about the Global Catalog server is that it provides the Global Address List (GAL) for Exchange users.

Security descriptors list groups and users that are allowed access to an object. They dictate the user and group access level and permissions. A security descriptor may further be used to specify events that you (or someone else) want audited every time that object is accessed.

Log on to a Windows 2000 server, and your security identity becomes encapsulated in an access token associated with your logon session. Start an application, and that application runs as a process within your logon session. Whenever the application tries to open a file or access an object, it identifies itself as your agent by presenting your session access token.

For example, when your Exchange client tries to access the GAL, your client uses Lightweight Directory Access Protocol (LDAP) or Name Service Provider Interface (NSPI). (NSPI is used by earlier MAPI clients to gain access to ADS.) Your client provides your session access token to the ADS in the first case and to Exchange 2000 in the second. The rest of the story in the second case is that the Exchange Directory DLL on the Exchange 2000 server acts as a directory store proxy service. It intercepts and forwards the packets to ADS, including your session tokens.

### LSA Component

<table>
<thead>
<tr>
<th>Component</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Accounts Manager (SAM)</td>
<td>Location for storage of local security accounts. Also enforces locally stored policies and supports APIs.</td>
</tr>
<tr>
<td>Multiple Authentication Provider</td>
<td>Holds LSA components together.</td>
</tr>
</tbody>
</table>
Auditing

In order to detect intruders and attempts at compromising data on your system, Windows 2000 allows you to audit all security-related events. (This is why it is so important that Exchange Administrators have adequate permissions as Windows 2000 Administrators, or very close ties to that community.) In addition, Windows 2000 maintains a Security Log containing security events, as shown in Figure 22-1.

![Figure 22-1: The Security Log in Event Viewer](image)

You can specify which events should be written to the security event log. The audit entry into the log shows the action performed, the user who performed it, and the date and time of the action. You can audit both successful, as well as failed, actions. In Table 22-2, we list settings from Microsoft’s Secure Domain Controller security template. Later on in the chapter, we show you how to use the Security Configuration Tool Set and Security Templates snap-in to apply these changes to a machine.

<table>
<thead>
<tr>
<th>Security Event to Be Audited</th>
<th>Typical Security Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Logon Events</td>
<td>Failure</td>
</tr>
<tr>
<td>Account Management</td>
<td>Success/Failure</td>
</tr>
<tr>
<td>Directory Service Access</td>
<td>Failure</td>
</tr>
<tr>
<td>Logon Events</td>
<td>Failure</td>
</tr>
<tr>
<td>Security Event to Be Audited</td>
<td>Typical Security Setting</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Object Access</td>
<td>Not Audited</td>
</tr>
<tr>
<td>Policy Change</td>
<td>Success/Failure</td>
</tr>
<tr>
<td>Privilege Use</td>
<td>Failure</td>
</tr>
<tr>
<td>Process Tracking</td>
<td>Not Audited</td>
</tr>
<tr>
<td>System Events</td>
<td>Not Audited</td>
</tr>
</tbody>
</table>

Enabling auditing and monitoring security logs does not guarantee that you will catch a rogue Administrator. A person with administrative rights, some forethought, and creativity can hide their actions by turning off auditing, clearing security logs, or creating an alias to disguise their actions. For this reason, we recommend that you audit the success and failure of policy changes and account management.

**Kerberos**

Windows 2000 introduced a new authentication protocol called Kerberos V5. (The original Kerberos V4 was designed and deployed in Project Athena at MIT back in 1987 and is considered a classic.) In Windows 2000, Kerberos supersedes the NTLM authentication protocol, which is a challenge/response authentication protocol. NTLM was the default for network authentication in Windows NT version 4.0 and earlier. NTLM continues to be supported in Windows 2000 but is no longer the default.

We can use Figure 22-2 to explain how mutual authentication using Kerberos works.

**Figure 22-2: Mutual Authentication using Kerberos**

The term “mutual authentication” refers to the need for both the recipient of a service and the provider of a service to authenticate themselves to each other. Mutual authentication must be performed before the service is rendered or accepted by the provider of the service and the recipient of the service, respectively.
Since both the provider and the recipient of a service cannot trust each other, there must be a mechanism by which they can authenticate each other. This is where Kerberos comes in. Kerberos allows a service to authenticate a recipient so that access to the service is protected. It also allows a recipient of a service to authenticate the service provider so that there is protection against “rogue” services.

The following steps provide a birds-eye view of how Kerberos facilitates the mutual authentication between a service and its recipient.

1. The Kerberos mechanism begins when a service provider is installed. You register the service and give it a Service Principal Name (SPN) using the Setspn utility. The SPNs are entered in the Active Directory under either a user account or a computer account that the SPN will run under.

2. When a recipient wants to request service from a service provider, the recipient creates another SPN that includes information about the requestor.

3. The recipient uses the Security Service Provider Interface (SSPI) to present its SPN to the Key Distribution Center (KDC) for its domain account.

4. The KDC for the recipient’s domain account searches the forest for an account on which the SPN is registered. When it finds the account, the KDC creates an encrypted message using the password of the service provider’s account.

5. The KDC sends this encrypted message to the recipient of the service.

6. The recipient of the service passes the message it receives from the KDC to the provider of the service it is requesting.

7. The service can now authenticate the recipient, since it can validate its own password.

8. In order to authenticate the recipient, the service sends the message to the client’s KDC, the originator of the message, to validate itself and thus close the loop on the mutual authentication.

When the mutual authentication mechanism was designed, much thought was given to each step in order to make sure that a “rogue” service or an “impersonating recipient” couldn’t defeat the integrity of the process. By having two KDCs involved in the process—one on the recipient side and one on the service provider side—it is impossible to violate the integrity of the mutual authentication mechanism.

**Certificate Services**

Windows 2000 includes a new service called Certificate Services, which provides facilities for issuing and managing certificates used in software security systems that use public-key cryptography. This is used in conjunction with the Exchange 2000 Key Management Service. Certificate Services performs a central role in the management of software security systems to enable secure communications across the Internet, corporate intranets, and other non-secure networks.
Certificate Services works like this:

1. A request for a certificate is generated by a user or service and transmitted to the Certificate Server via one of a number of transports, including RPC and HTTP.

2. The Server issues a certificate after it compares the request to policies (both custom and on-site) and deals with setting any optional certificate properties.

Certificate Services enables Administrators to manage the certificates by adding elements to the Certificate Revocation List (CRL) when required and by publishing a signed CRL regularly.

Certificate Services has features that make it a development platform. If you or your programming staff want to provide support for other transports, policies, certificate properties, or certificate formats, programmable interfaces are provided. If your development team wants to build a certificate authority, they can do so using Certificate Services to issue, track, manage, and revoke certificates. External applications interface with Certificate Services using the Component Object Model (COM).

Certificate Services consists of the server engine, the server database, and a set of modules and tools. Together, they provide the functionality of a certification authority.

To set up a certificate server, you can use the Windows Components wizard in Windows 2000 to configure your server screen. You can find it under Advanced/Optional Components, as shown in Figure 22-3. If you have IIS running on the machine you install it on, it will have to stop IIS. Since this is a Windows 2000 application, consult the Windows 2000 Server Bible for more information on installation.

Figure 22-3: Installing a Certificate Server
Encrypting File System

To protect your messaging and other data while it resides on the server hard disks, Windows 2000 includes a new file system capability: the Encrypting File System (EFS). Since Exchange allows access to files over the Internet, EFS is very important. EFS allows you to encrypt your data directly on volumes that use the NTFS file system and uses certificates that are based on the public-key cryptography and that can be issued by the Certificate Services.

Clients procure certificates by opening your browser and typing `\yourserver\certsrv`, as shown in Figure 22-4. This brings you to the certificate Web screen wizard, which provides you with a certificate.

![Microsoft Certificate Services - Microsoft Internet Explorer](image)

**Figure 22-4:** Procuring an EFS certificate

Internet Protocol Security

Windows 2000 utilizes an Internet Protocol Security (IPSec) framework for ensuring private, secure communications over IP networks. Like EFS, IPSec uses public-key cryptography. Unlike Certificate Services and KMS, which provide security on the application layer, IPSec provides security on the IP transportation layer. IPSec also provides protection for the TCP/IP protocol stack, such as Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP), and other protocols that send traffic at the IP layer.
IPSec is especially important in an Exchange 2000 environment. In the days of Exchange 5.5, servers encrypted their communications with one another using RPC encryption. Now that SMTP is used as the native server-to-server protocol, communications are no longer encrypted. One way of protecting the data stream from server-to-server is by using IPSec.

If you choose to enable IPSec, you must enable it by using an IPSec policy. IPSec policies can be accessed through the properties of the network connection that you wish to secure. Go to the Network and Dial-up Connections icon in Control Panel and then go to the advanced properties of the IP protocol. In the Other tab, you will find the option to enable IP security by creating a policy. There are three policies you can use:

♦ **Client:** When using the client policy, the machine will not encrypt traffic unless requested by another machine. This is the lowest form of security.

♦ **Server:** When using the server policy, the machine will request that any machine communicating with it use IPSec. If the machine does not support IPSec, communication will be unsecured. Communication with any IPSec-capable machine will be encrypted.

♦ **Secure Server:** This setting forces all communication with this server to be encrypted. This is the most secure policy but also incurs the most overhead on the system and has the most potential administrative issues. If a machine cannot use IPSec, it will not communicate with this machine. This includes non-IPSec enabled machines, NT 4.0 DNS servers, and most client workstations.

*Caution*

Keep in mind that encryption can add substantial system performance overhead.

### TCP/IP filtering

Another important Windows 2000 security feature that is heavily utilized in an Exchange 2000 environment is TCP/IP filtering. TCP/IP filtering allows you to specify exactly which types of incoming IP traffic are processed for each IP interface. You access IP filtering by modifying the properties of a network connection. To access this function, go to the Network and Dial-up Connections icon in the Control Panel and open the properties of the connection you wish to modify. Highlight the IP protocol and choose Properties. Once in the TCP/IP properties screen, choose the Advanced button, and then switch to the Options tab of the Advanced properties. In the Options tab, highlight IP filtering and choose Properties. The screen shown in Figure 22-5 will appear.

Enabling IP filtering allows you to do configure protocol and port filtering. Since you might disable ports required for critical network services, you should be extremely careful when port filtering of any kind.
Security Configuration Tool Set


The Security Configuration and Analysis snap-in, as shown in Figure 22-6, uses a database to perform analysis and configuration functions.
With the snap-in, you configure local system security. The snap-in uses personal databases, which enable you to import security templates created with the Security Templates snap-in and apply these templates to the Group Policy object for the local computer.

You must configure security settings in all appropriate group policies in order to effectively enable auditing on target machines. For example, if you enable auditing via a local policy on a domain controller, but do not enable auditing at the Domain Controller OU, auditing will not occur. This is because group policy inherited from the Domain Controller’s OU conflicts with the local machine policy. Make sure you test your security policies to ensure that they function as you expect.

The Security Templates snap-in, shown in Figure 22-7, provides a centralized method of defining security. With the snap-in, you can view the full range of system security settings, which are adjusted and applied to a local computer or imported to a Group Policy object.

The Security Configuration Tool Set is especially helpful when you have upgraded a domain controller from Windows NT 4.0. Upgraded domain controllers do not have the same security settings applied as freshly installed Windows 2000 domain controllers, because Microsoft did not want to change any custom security settings that might be required for applications or services. This application is great for pointing out those differences.

Now that we have reviewed the important security features of Windows 2000, we can turn our focus to the security features provided by Exchange 2000.
Exchange 2000 Security

While the security features of Windows 2000 are mainly focused on context-authentication and access control, the Exchange 2000 security features focus on content-securing messages.

Key Management Service

Exchange 2000 uses the Key Management Service (KMS) to provide additional security tokens to users to afford them the ability to secure message content. KMS is based on the Windows 2000 Certificate Services (unlike the Key Management Server in Exchange version 4) and utilizes public-key cryptography.

To install Key Management Service, you must have a Certificate Authority (CA) available. An example is Certificate Server running with a valid certificate. If you use Certificate Server, you must have it installed with policies for “Enrollment Agent (computer),” “Exchange User,” and “Exchange Signature Only” or you will receive an error dialog box that tells you the installation cannot be accomplished without them. With the CA installed properly, invoke the Exchange 2000 Installation Wizard and choose to install Microsoft Exchange Key Management Service, as shown in Figure 22-8.

![Microsoft Exchange 2000 Installation Wizard](image_url)

Figure 22-8: Installing Key Management Service

Next, you will be prompted for password handling. The KMS requires a key to start. This helps you securely authenticate the manager that provides security. It is something like having a lock on a key cabinet. Your options are to see the password and write it down (or commit it to memory) or put the key on a diskette or hard drive...
location. If you use a diskette, you will have to insert it every time you want to start the service. If you place the key on a hard drive, you will have to point to it each time (and hope someone does not delete the file). We listed these options in order of best to worst security in most environments.

1. Paper secured in a safe.
2. Diskette secured in a safe.
3. A key left on a hard drive.

The requirement for a key to start the service means that it will not start automatically if the server is rebooted.

On the next screen, you choose the location where the startup password and the backup password are written to. Default for location is A. The file will be called KMServer.pwd and can be read with any text editor, such as Notepad.

Open your Exchange System Manager. You will see a new object called Advanced Security. Highlight it, then highlight Key Manager. Right-click. Choose Start the service and you will see the Start Key Management Service dialog box shown in Figure 22-9.

![Figure 22-9: Starting the Key Management Service](image)

Right-clicking on the KMS while it’s running enables you to stop the service, enroll users, revoke certificates, recover keys, export users, import users, save KMS certificates, and change the startup password. Enroll users here as the first step in enabling
them for signing and sealing messages and postings. That process starts with the Key Management Service Login box, shown in Figure 22-10. This box displays the Domain\username and has a place for the Key Management Service password.

The default password for KMS administration is “password.” The default account is the one you were logged in as when you installed it.

Once you’re logged in, another box shows you the total number of Administrators entered and the total needed for this activity. Some KMS activities may require more than one Administrator, and each will have to enter their own passwords. As Administrators are added, each is given their own password.

The General property page displays the Certificate Servers it knows about and when their certificates expire. You can view the details on any Certificate Server by clicking on the View Details button. The Administrative property page requires a password for the Administrator user who is trying to access it. Type your password in here. Again, the default is “password.”

Never delete Windows 2000 accounts without first removing them from the KMS Administrators, and be cautious not to let your certificates expire before renewing them.

The next property page is the Passwords page, as shown in Figure 22-11.
Figure 22-11: Passwords page of the Key Manager Properties box

Again, the properties page requires your administrative password. Here, you specify the number of Administrators, each with their own password, required to

✦ Add/delete Administrators, edit multiple password policies
✦ Recover a user’s key
✦ Revoke a user’s keys
✦ Change certificate versions
✦ Import/export user records

The default for each of these tasks is 1. You can access the last property page, Enrollment, by entering your password. Here, you can specify whether to send the password via e-mail. (The alternative is hand delivery.) E-mail can be a security breach if you are using clear text and do not have your transmissions protected with some encryption. Voice mail would be even worse to use. If the token is known to nefarious characters, the value of signing and sealing is nil. Protect this process. If you choose, you can customize the message that will be sent in e-mail, as shown in Figure 22-12 in the Welcome Message page. If you select the e-mail option to send the tokens, this is the message each end-user will receive via e-mail when a token is generated. Also note that if you have older clients, you can check to have X.509 V1 certificates generated to accommodate them.
Enabling a user on the server and client

Right-click on Key Manager again and select Enroll Users. You will be requested to enter your password. Then you will see a dialog box requesting you to specify whether you want an alphabetical list from the global address book or a display of the mailbox store, Exchange servers, and administrative groups of eligible users. Make your choice and click OK. You will see a list, as shown in Figure 22-13. Select Enroll Users.

With the option we selected, an enrollment e-mail is sent to the selected users. They need the token to procure a certificate, otherwise known as a digital ID, before the security options will activate on Outlook other e-mail clients. Once the token is received in the mailbox (this could take a while if you have a lot of latency in your system), the user should select the Tools → Options → Security property page. Click on the Get a Digital ID setting in the lower right. Then select Set up security for me.
on the Exchange Server. This causes the server to generate a certificate. The certificate itself looks like what appears in Figure 22-14.

![Figure 22-14: A digital certificate](image)

In the window where Exchange prompts the user to enter the security token provided by the Exchange KM Administrator, you must specify the location of the file to store private security information. You must also give consideration to the location of this file, because if you lose it, the Exchange KM Administrator has to recover the key. Additionally, the security file has to be accessible from the Exchange client program. If the user is going to be a roving user, the security file must be on a network share or a home directory that is accessible to the user from whatever workstation he or she logs on to and from which the user runs Exchange.

The user will be asked for a token and the user name. She will enter her name and the token provided orally, in a written document or in the e-mail. This screen appears slightly different in each version of Outlook. Figure 22-15 displays the Outlook 98 version. Click OK.

![Figure 22-15: Entering the token](image)
The contents of the digital ID e-mail message to the user include:

- The X.509 sealing certificate
- The X.509 signing certificate
- The certification authority’s certificate
- The user’s private sealing key, generated when security was enabled for the user

When you click OK, your client will communicate the request to the KMS via e-mail and will notify you that the request has been sent and is awaiting a reply. Assuming the KMS is running and all else is well, a reply will be received from the KMS. To open it, you must have the password your user created while generating the request. Then the user is prompted to specify whether they want to add the certificate to the IE Root Store. (The Root Store is user-specific, so there is no issue with confusing or sharing keys.) In Outlook 2000, the user will receive an e-mail stating that the new certificate has successfully been installed. By default, the client is set up for S/MIME messaging and 40-bit encryption, as shown in Figure 22-16.

**Figure 22-16: S/MIME as default setup**

If the user hits Cancel either time they are asked to enter their password, the process is aborted. They will need to contact the KMS Administrator, who will revoke the earlier certificate and issue a new one.

The Security property page has two parts. The first part is the Secure e-mail. It allows check boxes for “Encrypt contents and attachments for outgoing messages” and “Add digital signature to outgoing messages” in the “Secure e-mail” area of the
property page. If the user does not select those check boxes, they can still create a signed or sealed e-mail when they create a new e-mail. To do this you select New to create a new e-mail, select File and properties and then the tab for security. On the Security page, choose to Encrypt or Add a Digital signature.

The second part of the Security Property page is for procuring your digital ID. You can import it or export it using the button. You can get a Digital ID by clicking on the Get a Digital ID button. This will cause the Get a Digital ID property dialog box to show up as shown in Figure 22-17. Your options are to get an S/Mime certificate or request that security be set up on the Exchange Server (the option selected in Figure 22-17).

![Figure 22-17: Adding a digital signature](image)

Every time your Exchange server is restarted, you must restart the Key Management Service, which requires a password. This is a security feature. Similarly, every time you want to invoke an action in Key Management Service Administration, you will need to enter at least one password and be an approved Administrator.

Before we leave the Administrator, there are a couple more items of importance to discuss. You are allowed to revoke certificates and choose the reason for the revocation from a list. In general, you do not want to revoke unless it is permanent. It is better to reclaim a certificate than to revoke and create a new one. All certificates, even revoked ones, stay in the database forever. So revoking 1,000 certificates and then creating new ones when you could have simply reclaimed the tokens is not a good idea. Figure 22-18 shows the revocation options.

You can also administer keys using Active Directory. Select a user, then go to the Exchange Features property page. After CA and KMS are installed, you should see an option for E-mail Security. Highlight it and click on Properties. You will be asked to enter the Key Manager Service password. You will only be allowed in if you know the password and are an Administrator of KMS. You will see a screen for allowing advanced security, recovering tokens, and revoking security, as shown in the Exchange Security page in Figure 22-19. Figure 22-19 also shows the dialog warning for creation of a temporary key.
The following sections will provide a more detailed discussion of why users want to Encrypt (Seal) or Add a Digital Signature (Sign) and what is going on when they do.
Protecting content by sealing messages

Some users want to protect the content of a message by encrypting it. Once encrypted, the message content is protected from anyone who does not have the means to decrypt it. Even a messaging Administrator who can gain access to a mailbox cannot readily decrypt individually sealed messages. Outlook 2000 clients, in conjunction with Exchange 2000, provide the means to encrypt the message content. This is called sealing the message. The process encrypts as specified in the KMS. Together, the keys and associated algorithms transform the body part of the message and attachments into data readable only by someone with a key or a very large computer system to crack the key. (The P1 envelope is not encrypted, as it needs to be read by messaging system components like MTAs in order for the message to be routed properly.) For the most part, the sealing process means the contents (including attachments) are readable by the intended recipient and originator only.

An encrypted message can only be sent to other users that have advanced security enabled. When you attempt to send a sealed message, the Outlook client initiates a call to determine

- Whether advanced security is enabled for each user or all the members of a DL to whom you addressed the message. If one or all of the addressees does not have advanced security enabled, Outlook will warn you.
- The level of encryption for each message recipient, such as DES, CAST 40, or CAST 64.

If you have advanced security enabled, Outlook will

1. Retrieve a copy of the user’s sealing certificate.
2. Use the sender’s security password to extract the stored sealing certificate.
3. Generate a bulk encryption key for each message recipient.
4. Encrypt the bulk encryption key with each recipient’s public sealing key that was retrieved earlier in the process into what is called a lock box. If the message has multiple recipients, there is a lock box created for all recipients.
5. Submit the encrypted message, lock box or boxes, and the sender’s sealing certificate to the Exchange Server Information Store through normal message delivery procedures.

When you receive an encrypted message, it appears in the inbox like a normal message, but an icon shows an envelope with a lock on it. That lock indicates the message was generated by the user that chose to encrypt the contents and attachments. They did this by a selection on his or her message options page while preparing to send the message, as shown in Figure 22-20.
When a user attempts to open this message, Outlook prompts the user to enter the private password in the next screen that the system presents before he or she can view the message. Once you enter the security password, Outlook lets you see the contents. Additionally, you can specify to remember the secure password, and Outlook will not prompt you for it again until your next logon.

When you click on an encrypted message in an attempt to read the contents, the Outlook client performs the following actions:

1. Prompts the user for the secure password.
2. Extracts the private sealing key from the .EPF file.
3. Decrypts the lock box that Exchange created when the sender submitted the message.
4. Uses the decrypted bulk encryption key from inside the lock box to decrypt the original message.
5. Displays the message contents on the screen.

**Signing a message**

When you sign a message, the message is created in clear-text. KMS then calculates a hash for the message. The hash or message digest is a 128-bit number that uniquely identifies the contents of the message. The same message will always result in the same digest. If the message is altered in any way, it will derive a different digest value.
When you digitally sign an e-mail using Exchange 2000 KMS Security, Exchange performs the following actions:

1. Hashes the original unencrypted message to obtain the message digest.
2. Prompts the user to enter his or her security profile password.
3. Extracts the private signing key.
4. Encrypts the message digest with the sender’s private sealing key, or digital signature.
5. Submits the signing certificate, the digital signature, and the message to the local Exchange Server Information Store for delivery.

When the message arrives at its destination, the recipient can open the sender’s certificate that includes the sender’s key. Using the key, you can decrypt the message as well as the hash. You recalculate the hash, and if it matches the hash that was sent with the message, you know that the message was not altered.

Figure 22-21 illustrates the Change Security Settings dialog box.

**Figure 22-21: Client security settings**

**Verifying a signature**

Signature verification is simple. While a secure e-mail is open, the user must click on the signature verification button that calls up the window. Digital signal verification shows the full X.500 distinguished name of the sender, as well as information indicating if the contents were altered after the message was sent, if the signature is suspended or has expired, or if an unknown authority issued the signature.
To verify the signature of a message signed using Exchange server advanced security, the Exchange client has to hash the message contents and compare the message digest to the message digest created at send time. If the message digests are the same, the digital signature guarantees that the message is from the sender and also that it was unaltered in transit. Digital signatures are not checked unless specified through the method demonstrated above.

Since you can decrypt the message with the sender’s key, you also know that the message was sent from the sender’s account using their security token. (You really never know that it was a specific person who sent it, but you do know whose credentials were used.)

For more on encryption and signing or how Exchange 4 and 5 worked, see Exchange Server 5.5 Secrets, published by Hungry Minds, Inc., formerly IDG Books Worldwide, Inc.

**Algorithms and keys**

Exchange 2000 uses three different hash algorithms: MD-4, MD-5, and SHA-1. It also uses four types of encryption algorithms: symmetric, asymmetric, block, and stream.

The *symmetric algorithm* uses a single key, which is known by both the sender and the recipient to encrypt and decrypt a message. This algorithm requires that the sender communicate the key to the recipient in a secure way.

The *asymmetric algorithm* uses two keys: one for encryption and another for decryption. The public key is used for encryption and the private key is used for decryption. Using these two keys ensures privacy and provides authentication. The public key is usually placed in a directory, or a location available to other users. However, the private key is kept in a secure location and is available only to the owner of the key pair.

A *block algorithm* uses shared-key encryption. Every message is broken down into fixed-length blocks, and a shared-key is applied to each block. In most cases, this block size is 64 bits. The decryption operation decrypts each block with the same shared-key and rebuilds the original message.

A *stream algorithm* is a variation on the block algorithm. A stream algorithm can operate on small units of plain text, usually bits. A stream algorithm is generally much faster than a block algorithm and can be applied to data as it is sent or received “on-the-fly.”

Exchange 2000 uses seven different encryption algorithms: CAST-64, CAST-40, DES, 3DES, RC2-128, RC2-64, and RC2-40.

Figure 22-22 shows the property page on the server where you specify which type of algorithm your system will use. The Exchange 4/5 encryption default is DES for North America and Cast-40 for Other. S/MIME encryption defaults for North
America are 3DES and RC2-40 for Other. Note that you can also set the security message format to Exchange 4/5 or S/MIME. The default is S/MIME.

![Algorithm property page](image)

**Figure 22-22**: Algorithm property page

There are many laws that cover aspects of exporting and using encryption products. These laws keep changing. If you are involved in security, make sure you know what local laws apply. If your project is international, also make sure you know what laws apply in all the countries you will be involved in and that you know the details of import and export laws on encryption products.

**Offline Usage**

Using advanced security to encrypt and send a message requires the Outlook client to open a conversation with the directory, both to check the directory to see if the recipient is advanced security enabled and to retrieve the recipient’s public key. Therefore, encryption of messages offline is not possible. You can still decrypt messages sent to you, sign messages to send to others, and verify signatures on messages, but you cannot originate new encrypted messages.

**Virtual Server security**

You can control connections to your Exchange 2000 server by restricting the IP addresses that can connect. The IP addresses that you can restrict may reflect individual nodes, subnets, or entire domains. You can set the Virtual Server security using the Internet Information Services (IIS) snap-in for the Microsoft Management Console (MMC), as shown in Figure 22-23.
You can right-click on the Virtual Server and select the Default SMTP Virtual Server Properties dialog box, then the Access tab, as shown in Figure 22-24.

To set up virtual access control, select the Connection button in the Connection control section. The Connection dialog box, shown in Figure 22-25, appears.
In order to restrict which computer, subnets, or domains can access the Virtual Server, click on the Add button. The Computer dialog box, shown in Figure 22-26, appears. This dialog box allows you to specify a single computer, a group of computers, or a domain that can be restricted.

Controlling connections to virtual servers can prevent many possible network attacks, such as unauthorized access. You can selectively include or exclude single computers, subnets, and entire domains from accessing a virtual server.

**Permissions**

When you install Exchange 2000, the installation process sets up several roles with permissions for administering the Exchange 2000 organization. The roles that are set up include:
The Exchange Full Administrator: Grants each member of the role permission to administer Exchange system information and modify permissions. Users assigned to this role have full control of the Exchange organization.

The Exchange Administrator: Grants each member of the role permission to administer Exchange system information but not modify permissions. This role is targeted for a support staff member that has to administer the Exchange organization but does not need to modify permissions.

The Exchange View Only Administrator: Grants each member of the role permission to view Exchange configuration information.

Development security
Microsoft has given special attention to the power of Event Sinks and to preventing their possible misuse. To provide security, Event Sinks can only be registered by someone with physical access to the server, so you want to protect that access. Only the owner of a folder can create an item with content class “urn:content-class:storeeventreg.” Further Event Sinks may implement the interface, ICreateRegistration, to validate a registration request. Your Event Sink is prohibited from accessing anything using the security credentials of “Euser-EXSTOREEVENT,” and they are prohibited from accessing anything outside the logon of the passed session.

Outlook 2000 security highlights
To ensure your Exchange messaging environment is secure, you need to spend some time focused on your clients. Clients, such as Outlook 2000, provide vast utility for end-users. They can also provide access points for those who wish to do your data harm. Here are the areas to focus on.

Secure e-mail content
To secure the content of your messages, you should use a combination of digital signatures (signing) and encrypted messages (sealing). Encryption can protect the message data while on local, network, or any other storage medium. Encryption also can protect a message during transport, because even if it is captured, it cannot be decrypted. Still, you should also make sure that the communication over your transport and your authentication method are as protected as possible. Techniques for increasing transport security and authentication include VPN, TSL, Kerberos, SSL, and IPSec.

Macros and attachments
Macros that are run from a message can expose your system to harm. Outlook 2000 includes security levels for macros. The Security level defaults to medium. Other options are
✦ Low: No security
✦ Medium: Signed Add-in runs, user prompted when unsigned Add-in is loaded
✦ High: Add-ins must be signed, unsigned Add-ins won’t load

The exception is that all COM Add-ins are trusted by default.

Macro exposure can be introduced through attachments or embedded code. E-mail attachments can also introduce viruses.

**Virus protection**

There are three areas of attention for virus protection in Exchange messaging systems. The first is the client’s security. We believe very strongly that not having virus protection running on every client and configured properly is a serious error. There are many quality vendors out there that have products that can help you. Even after you procure the product and install it on every machine, you still need to address configuration. In our opinion, proper configuration includes regular updates of the virus definitions, heuristics set on, e-mail, and Internet access protection selected, as shown in Figure 22-27.

![Figure 22-27: Client virus protection configuration](image)

Many products reset the configuration to the defaults when updated, and many users change them if allowed. Make sure the settings on your clients are what you expect.
But as an Exchange Administrator, don’t stop there. Client protection should be the last line of protection, not the first. Before a virus coming through your messaging system ever hits your client, it should be caught by one of the third-party packages running on your Exchange servers. Many of the major virus manufacturers make products that scan messages and attachments and deal with them as they go through your Exchange system and before they get to the clients. Each Exchange server should also be installed with the same protection loaded on any client in your organization as it may be used as a client workstation and may have a floppy in it with a virus.

Lastly, since the preponderance of e-mail viruses comes to us from the Internet, we believe you should also have an SMTP virus protector running in front of your SMTP connectors.

Why so many lines of protection? If you have ever had to clean up viruses in large environments where companies fear that they may have inadvertently proliferated the virus and may have damaged their relationship with customers or vendors because of the proliferation, you already understand the value proposition of pre-empting viruses with a multifaceted defense. In addition, when new viruses come out, not all manufacturers get updates out at the same time, and often they come out for the client product first. Once again, this multifaceted, layered approach to protection makes it more likely that you’ll have the protection you need more quickly than if you relied on a single product.

Summary

You have now been exposed to the many layers of Windows 2000 and Exchange 2000 security. You see how Exchange 2000 security builds on the foundation of the Windows 2000 security subsystem and how third-party products such as virus protection fit into the system. You should now have a better understanding of what your mandate for securing the corporate record means and how you might do it as an Exchange Administrator. You know that setting up and maintaining a secure electronic messaging system that will authenticate and protect the content of electronic mail throughout your organization requires knowledge and vigilance.
In this chapter, we highlight and discuss major topology and design considerations that should be part of your Exchange implementation planning effort. Even if you already have a well-thought-out Windows 2000 topology and design, you will want someone astute in the considerations that are specific to Exchange to review and adapt the plan, paying special attention to the areas highlighted in this chapter. Your particular situation may call for other considerations not covered here, or draw you to a different conclusion about how to handle unusual issues based on information we provide elsewhere in this book. The purpose of this chapter is to help you understand how to extrapolate the concepts we discuss elsewhere in the book and apply them to your design and especially your topology.

Your Business Organization

Windows 2000 and Exchange 2000 are two sides of the same coin. The good news is that you now have the flexibility to ensure that your messaging and computing implementation correctly reflect your company’s business organizational structure. The bad news is that many organizations do not have a comprehensive model for their company’s organization—or they may have many different competing models simultaneously and therefore struggle with getting their hands around implementations of Windows 2000 and Exchange 2000. It is difficult, maybe even impossible, to reflect a nebulous organization in a tangible messaging system. Of course, instead of trying to determine the structure
at the beginning, Windows 2000 and Exchange 2000 enable you to start with one
structure and change it later with some technical ease. Unfortunately, trying to
change your directory structure can easily delay your project and cause large
amounts of angst in the process because of business organizational concerns and
conflicts. A practical approach to this problem is needed.

Take advantage of the flexibility that Windows 2000 and Exchange 2000 provide. We
recommend that you spend only some of your precious planning time trying to get
all the organizational decisions the affect your design made now. If you can, try to
define the likely outcomes of issues in question so that you can build in that direc-
tion, but do not get bogged down in internal organizational dilemmas or spend an
inordinate amount of time figuring out the organization. To make progress, you
want to highlight options, press for decisions, illuminate implications, and imple-
ment as much as can be decided definitively, while taking your best shot at the rest
and relying on the flexibility of the system to allow adaptations later. When all is
said and done, you want your messaging environment to reflect your business orga-
nizational structure. To the degree that your business structure is unknown or at
odds, you will have to implement an arbitration of the organizational structure. In
some cases, planning the messaging and information infrastructure can serve as a
catalyst for understanding, or even modifying, the organizational structure.

You should focus particular attention on the administrative model, the message
routing model, and the physical placement of devices and services, all of which can
and should reflect your organization. The administrative model defines who has per-
missions to do what. The message routing model defines how messages should
move across logical and physical boundaries (your LANs, WANs, sites, domains,
external connectors, and so on). The physical placement of devices and services is
self-explanatory, but should be deeply tied to the first two models.

If you have an existing Exchange 5.5 environment (with the latest service packs and
Windows 2000) and do not plan to eliminate it at the time you create your Exchange
2000 environment, some of your decisions will be dictated because you will be
forced to run in mixed mode. In that scenario, there are certain restrictions, includ-
ing the prohibition from creating multiple routing and administrative groups, until
you get rid of your last Exchange 5.5 server and form a native Windows 2000/
Exchange 2000 topology. Consequently, while you may have a very good idea what
your administrative and routing structure should look like to emulate your business
organization, you may be limited in putting it into effect in the first phase if you are
coexisting with Exchange 5.5.

In a migration scenario, where you are going to remove all the 5.5 servers rather
than coexist, you can get started by creating a routing group and administrative
group for each existing Exchange 5.5 site. Then you can evolve to a more sophisti-
cated structure, with administrative assignments that are more appropriate as you
gain familiarity with Exchange 2000 and how to use it to reflect your organizational
structure.
**Naming Conventions**

As with any implementations or migrations, your deployment of Exchange 2000 can be a good time to review your naming conventions and update or alter them rather than continue with the old conventions or with the chaos that results from no conventions. Exchange 2000 and ADS have namespace differences from the world of NT and Exchange 5.5, which means that certain savvy organizations will jump way ahead in the script and implement a common namespace for Windows 2000 and Exchange 2000 early in their migration effort. In Windows 2000, namespace and network topology are distinct entities, making a common namespace possible. In a perfect world of common namespace, I would be able to log on to the Windows 2000 domain using elements from my e-mail address. For instance, if my e-mail address is Robert.Guaraldi@Ilmarin.com, I would want to be able to log on to my Windows 2000 Ilmarin.com domain controller using the name Robert.Guaraldi. Because both login names are the same, users would have less to remember to log in and be forever grateful. Common namespace is where most Windows 2000/Exchange 2000 implementations will ultimately end up. However, we expect few will try to implement a common namespace initially, due to legacy and migration complexities (like running in mixed mode rather than native mode). Rather, we expect them to continue using legacy geographical and hierarchical name structures (for example, a login Robert.Guaraldi to NorthAmerica.Ilmarin.com, where the NorthAmerica designation is geographical).

By default, Windows 2000 ADS allows one e-mail address for each object. Exchange 2000 expands the ADS schema when the Exchange version of ADS is loaded or when you run forest preparation, providing much more design flexibility. This schema expansion allows multiple e-mail addresses for each object, a feature that designers and administrators usually take advantage of. An example is allowing Robert.Guaraldi@Ilmarin.com as well as others like RGuaraldi@Ilmarin.com, BGuaraldi@Ilmarin.com, Bob.Guaraldi@Ilmarin.com for legacy messaging system compatibility.

With regard to namespace, Exchange 2000 differs from Exchange 5.5 significantly, as shown in Tables 23-1 and 23-2.

<table>
<thead>
<tr>
<th>Designation</th>
<th>Exchange 5.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>o=&lt;organization name&gt;</td>
<td>o=Ilmarin</td>
</tr>
<tr>
<td>cn=&lt;site name&gt;</td>
<td>cn=HQ</td>
</tr>
<tr>
<td>cn=&lt;recipients container&gt;</td>
<td>cn=Recipients</td>
</tr>
<tr>
<td>cn=&lt;unique alias in the container&gt;</td>
<td>cn=Rguaraldi</td>
</tr>
</tbody>
</table>
In Exchange 5.5, sites are an important part of the namespace. In Exchange 2000, sites are not important, but Windows 2000 domains define the namespace. While each object in the directory still has a unique DN that includes the site name, ADS does not use that name to identify objects. ADS allows multiple sites in a domain and uses a DN that does not include the site name to identify the objects.

<table>
<thead>
<tr>
<th>Designation</th>
<th>Active Directory Service (ADS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cn</td>
<td>Robert Guaraldi</td>
</tr>
<tr>
<td>Cn</td>
<td>Users</td>
</tr>
<tr>
<td>Dc</td>
<td>Ilmarin</td>
</tr>
<tr>
<td>Dc</td>
<td>Com</td>
</tr>
</tbody>
</table>

Whether or not you are implementing a common namespace, we have some suggestions for establishing naming conventions:

- Do not use function (for example, Messaging1) or geography (Freinzeltaly01) for server names. Many companies are moving to names that are theme-based, such as tree names like Maple and Aspen. The problem with functionality is that it can easily change during the life of a server. You may move BackOffice components from machine to machine as hardware needs change. Having a machine named SQL running Exchange is undesirable and mismatches functionality with name. The problem with geography-based naming is that in a virtual world, the servers can be anywhere and often move around. A Boston server that is physically located in Dallas because it was moved there a year after it was created in Boston only adds to the administrative confusion.

- Do not use characters incompatible with messaging and directory standards, such as X.400, X.500, DNS, and Internet standards, unless absolutely necessary. Make sure domain names conform to DNS character standards, which will lessen system substitutions and confusion for users and administrators. (NETBIOS names are notorious for underscores, which should be eliminated if possible.)

- Do not use names that might be culturally offensive, especially in international environments. Windows 2000 ADS is designed as a global system. Parts of the directory may be replicated to the four corners of the world. Cultural sensitivity is not only good world citizenship, it is an axiom of all international projects.

- Choose organizational names that encompass the entire enterprise.
When loading Windows 2000, select server names that are appropriate as Exchange messaging server names.

For mailbox names, try to establish a standard and get everyone in the organization to adhere to it.

Since mailbox names are so important, we want to go into them in some detail. Common conventions for mailbox names are shown in Table 23-3.

<table>
<thead>
<tr>
<th>User Name</th>
<th>Account/Alias name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virginia L. Guaraldi</td>
<td>VGuaraldi</td>
<td>First character of first name + entire last name</td>
</tr>
<tr>
<td>Virginia L. Guaraldi</td>
<td>Guaraldi-V</td>
<td>Last name “-” first initial of first name (a dash is preferred over an underscore for standards compatibility)</td>
</tr>
<tr>
<td>Virginia L. Guaraldi</td>
<td>VLGuaraldi</td>
<td>First initial of first name + first initial of second name + entire last name</td>
</tr>
<tr>
<td>Virginia L. Guaraldi</td>
<td>virginiaguaraldi</td>
<td>Entire first name + entire last name</td>
</tr>
<tr>
<td>Virginia L. Guaraldi</td>
<td>virginia.guaraldi</td>
<td>Entire first name “.” entire last name</td>
</tr>
</tbody>
</table>

In Table 23-3, we show capitalization and non-capitalization. The names that form a messaging server perspective are not case-sensitive. However, we believe it makes them more readable in the Global Catalog when naming conventions utilize common capitalization like VGuaraldi.

While planning mailbox names, make sure you investigate legacy messaging systems. Some will truncate your names. In some cases, you will want to adjust your naming conventions to ensure they work as well as possible after truncation. Old host-based systems and some PC messaging systems had a limit (often 10 characters) and would truncate anything longer.

Your Existing Messaging and WAN Environments

Your existing messaging systems and WAN environment may impose special constraints on your design. If you are migrating away from the existing system, the imposition may be temporary. The most common scenario for most migrations will
be from an existing Exchange system to Exchange 2000. As we have already stated, migration from Exchange 5.5 will affect your design considerations in a number of ways.

If you have a legacy Exchange environment older than the most recent service pack release for 5.5, you should make upgrading to the 5.5 most recent service pack a prerequisite for your Exchange 2000 deployment. Without that, it may be impossible for you to build an effective and stable environment.

The same is true of other legacy messaging systems. We cover coexistence and migration with and from many of them in Chapter 18 by looking at specific features of Exchange 2000 and how to use them. Here, we are more interested in the way those systems might affect your design efforts. Whatever your existing system, there are a few common critical areas to focus your attention on because they may affect topology and design considerations during the migration and after. We will now take a look at the more common topology considerations affected by legacy system migration. While there may be others, this should give you an idea of the type of scrutiny required and the ways it might affect your design.

### Physical server placement

Some companies want to consolidate servers as they migrate from whatever they had to Exchange 2000. They want fewer, larger servers, and they want them in the glass box environment where they can control power, redundancy, administration, physical security, climate, and so on. Other companies build the more traditional PC WAN environment, where resources like servers are distributed closer to users across the corporate WAN environment with higher speed WAN links between them. Exchange 2000 works well with either design, or even with a hybrid. The important issue is whether the servers have sufficient bandwidth between them and the users to successfully perform their roles and to ensure adequate response. For instance, an Exchange server requires access to a global catalog server to resolve user requests and certain system attendant requirements. If the only global catalog available is on the other side of a slow or unreliable WAN link, your service levels will almost certainly diminish, and the server may be unable to perform its tasks at all.

Many contracts with ISP and other WAN providers allow for increase and (sometimes) decrease in bandwidth with some notice. You want to discuss your possible requirements up front, as this will often affect what they recommend (for example, fractional T1 rather than 56K leased line—the former makes it easier to go to 128, 256, 384, or full T1 than the latter).

Our recommendation is that you approach your design in one of the following two ways and make your approach explicit in your written design plan.

- **Purposely overestimate bandwidth usage and server usage through modeling, piloting, and projecting.** After the system goes into production, you wean back the bandwidth as appropriate and sometimes even consolidate the servers. Once the system is up, you have the production measurements to
help you do this responsibly. This can be especially useful if you are migrating from an existing messaging system, such as NT4/Exchange 5.5. Often, it is easier to replace almost one-for-one on physical servers and then consolidate.

✦ **Estimate, pilot, and minimize the costs by designing less than you think may be needed.** Once in production, you increase as users complain or as you notice bottlenecks. This approach often results in lower initial cost and lower end-user satisfaction, but sometimes the end users do not notice. A case where this might be workable is where the rollout will take some time, and the system will not get full use during that time.

A hybrid of these two approaches is also common, where you assess the ramp of usage and utility and build your physical location map and bandwidth to stay just ahead of the requirements.

There are special cases like hybrid environments of Exchange 5.5 and Exchange 2000. In that type of environment, you can only move servers between sites before you install your first Exchange 2000 server. After that, you are not allowed to logically move the servers from site to site, though you could still move them physically. This prohibition could impact your plan for dealing with consolidation.

Whatever approach you use, certain services and servers have roles that require they be placed near other servers, users, or both. See the section on server roles in this chapter for more discussion on how to place servers based on role. A general rule of thumb is to ensure a minimum of 56KB from user to server, and 10Mbit from server to server where the servers need to talk to each other. We prefer to see 100Mbit server to server backbones on LANs (and sometime 1 gig) and full T1 between servers on the WAN backbone. A second rule of thumb is to keep unnecessary services off your Exchange boxes so they will provide the best possible messaging response to users. If you do add other services, carefully monitor their effect on memory, disk IO and LAN IO, and CPU usage. Then, adapt as appropriate.

Remember, the speed of the line only means the theoretical maximum that one might pump through it. Often, companies have other transmissions pumping across those lines. The real issue is available bandwidth for your messaging use. Even there, do not assume that the available bandwidth is static. Find out what other applications and uses will be coming online that might affect your usage.

**Connectors**

If the existing system is an Exchange system, many of the connectors in use by Exchange 5.5 could be mapped directly to Exchange 2000 connectors. Alternatively, you could go ahead and assess whether there should be changes to the connectors now that routing groups and administrative groups have changed. In most cases, you will be altering your topology to combine what used to be sites into new boundaries for routing groups and administrative groups. While doing that, you will want to revisit your connectors, your address space (where you control routing), and the resulting link state routing table. For example, you may choose X.400 connectors for
your 5.5 environment instead of site connectors, primarily because the site connector is not schedulable. When moving to Exchange 2000, you’ll want to consider using routing group connectors, which are schedulable and perform better than X.400 connectors.

Pay special attention to placement of bridgehead servers—especially if connecting Exchange 5.5 environments to Exchange 2000.

If your existing environment was not Exchange 5.5, then you still want to focus on connectors and establish them so that your messages and directory route speedily, deterministically, with redundant paths, and with sufficient security.

You will want to make some effort to understand the inherent routing of your ISPs, especially when using Internet connectors. This can be extremely important in international environments. It is possible that messages going from one location to another—only 200 miles apart as the crow flies—could be routed thousands of miles to get there. We heard of this happening in Greece, because the customer had chosen two different ISPs. The messages went all the way back to the US before they got to the other ISP. While you do not usually care what path the message takes within the cloud, knowing the routing can help you to determine whether one ISP is a better choice than another. It can also help when you are experiencing difficulties and troubleshooting.

### Authentication and management

If your existing environment was Exchange 5.5, you were limited in the administrative groups you were allowed to create. This changed in Exchange 2000. With Exchange 2000, you should review the permissions granted and most likely limit them far more than you did in Exchange 5.5.

With regard to authentication, two issues exist. First, authentication in Exchange 2000 is performed by Windows 2000 ADS. You want to have a global catalog server within high-speed reach of the users and the Exchange 2000 server to facilitate authentication. Second, different e-mail clients authenticate with different frequencies. You will want to study the chapters on different e-mail clients and give some thought to how frequently they authenticate in order to build your topology.

### Other areas to investigate

Legacy systems can make life interesting. Look carefully at special character requirements, naming restrictions, attachment handling and encoding, X.400 connector requirements, porting, and recovering old messages. All of them can affect your design—sometimes they require you to retain old servers somewhere on your map for a restoral.
Chapter 23 ✦ Topology and Design Considerations

Server Roles

With the addition of Windows 2000 and Exchange 2000, many new server roles are possible. In general, be cautious about adding additional services to Exchange servers expected to provide a robust messaging response. Specifically, here’s a quick look at the various server types from a topological perspective.

Exchange servers

The server roles you got accustomed to in Exchange 4 and 5 have been expanded in Exchange 2000. Now, we have Conferencing Server and the possibility that companies will build their Web servers under Exchange to take advantage of the caching. Table 23-4 lists common Exchange 2000 Server roles.

Table 23-4 Exchange 2000 Server Roles

<table>
<thead>
<tr>
<th>Server Type</th>
<th>Locate with Highest Bandwidth Possible to End Users</th>
<th>Create Multiples for Redundancy or Performance</th>
<th>Locate in Each Domain If Possible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mailbox (Private) Server</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Public Folder Server</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Web Server</td>
<td>Not necessary, but sometimes prudent</td>
<td>Yes</td>
<td>Depends on design goals</td>
</tr>
<tr>
<td>Conferencing Server</td>
<td>Yes</td>
<td>Yes</td>
<td>Depends on design goals</td>
</tr>
<tr>
<td>Connector</td>
<td>Not necessary, but sometimes prudent</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Windows 2000 servers

Table 23-5 shows the Windows 2000 server roles. Because Exchange 2000 environments install on and rely on Windows 2000, these server roles are required, as well as the ones specific to Exchange 2000 and shown in Table 23-4.
Table 23-5
Windows 2000 Server Roles

<table>
<thead>
<tr>
<th><strong>Server Type</strong></th>
<th><strong>Locate with Highest Bandwidth Possible to End Users</strong></th>
<th><strong>Create Multiples for Redundancy or Performance</strong></th>
<th><strong>Locate in Each Domain if Possible</strong></th>
<th><strong>Add Service to Machine Running Exchange</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>File/Print</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Undesirable to seriously use this capability on your Exchange box in busy environments</td>
</tr>
<tr>
<td>Domain Controller</td>
<td>Yes, preferably in the same Windows 2000 site</td>
<td>Yes</td>
<td>Yes</td>
<td>Acceptable with sufficient power on the server</td>
</tr>
<tr>
<td>Global Catalog</td>
<td>Yes, especially if Windows 2000 clients are being serviced</td>
<td>Yes</td>
<td>Locate in same site and domain as Exchange server</td>
<td>Acceptable with sufficient power on the server</td>
</tr>
<tr>
<td>DNS</td>
<td>Yes, for redundancy</td>
<td>Yes</td>
<td>Yes</td>
<td>Acceptable with sufficient power on the server</td>
</tr>
</tbody>
</table>

Multiple domain controllers within a domain are a very good design idea. They will decrease Exchange look-up traffic and replication. They also provide redundancy.

While it is possible to run multiple BackOffice services on the same machine, it should be done with careful consideration to resource usage. In most messaging environments that we work in, customers do not know how to predict usage patterns very well. Even when we have historical information, new features and capabilities drive radical changes in usage (imaging and graphics drive up message sizes, Internet connectivity drives up the number of e-mails, and so on). So our recommendation is to combine Exchange servers with other BackOffice and Windows 2000 services and products on the same box with caution.

**Other BackOffice servers**

In many organizations, Exchange 2000 lives in a design that includes other Microsoft BackOffice applications. To provide a more complete picture of server role design considerations, we thought it would be helpful to include some of the other server roles that might affect your design, as shown in Table 23-6.
### Administrative and Routing Groups

In your Exchange 2000 design and topology, rethink and rework your administrative groups and routing groups. Often, Exchange 5.5 sites will be combined into one Exchange 2000 routing group, or broken out into multiple Exchange 2000 administrative groups.

When planning, you will want to take the minimum four steps:

- Define your administrative and routing organizational boundaries.
- Add another administrator group for your Exchange technical support staff.
- Move your routing groups into that administrative group.
- Delegate appropriate permissions.

If your environment is one with a single routing group of servers in one location with full time and dependable connectivity, you probably do not need special routing groups. But it still may be desirable to create routing groups if your design objective is to manage the flow of messages across administrative boundaries. In this case, you still want to enable and create routing groups.

Some people feel that the best way to think about routing groups is as an analog to “Sites” in earlier versions of Exchange.

---

Table 23-6

<table>
<thead>
<tr>
<th>BackOffice 2000 Server Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server Type</strong></td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>SQL Server</td>
</tr>
<tr>
<td>IIS</td>
</tr>
<tr>
<td>SNA Server</td>
</tr>
<tr>
<td>Proxy Server</td>
</tr>
<tr>
<td>Index Server</td>
</tr>
</tbody>
</table>

---

**Administrative and Routing Groups**

In your Exchange 2000 design and topology, rethink and rework your administrative groups and routing groups. Often, Exchange 5.5 sites will be combined into one Exchange 2000 routing group, or broken out into multiple Exchange 2000 administrative groups.

When planning, you will want to take the minimum four steps:

- Define your administrative and routing organizational boundaries.
- Add another administrator group for your Exchange technical support staff.
- Move your routing groups into that administrative group.
- Delegate appropriate permissions.

If your environment is one with a single routing group of servers in one location with full time and dependable connectivity, you probably do not need special routing groups. But it still may be desirable to create routing groups if your design objective is to manage the flow of messages across administrative boundaries. In this case, you still want to enable and create routing groups.

Some people feel that the best way to think about routing groups is as an analog to “Sites” in earlier versions of Exchange.
You should build your routing groups based on the patterns of usage of your users. Who communicates to whom most often? The answer to that question will often illuminate the most efficient routing design. For example, if the pattern of use you expect or observe shows volumes of public folder replication between two servers, then it is best if they are in the same routing group.

Routing group effectiveness is dependent on reliable, full-time connections. When connections are intermittent or slow, routing groups technology may be ineffective. Also, remember that in a mixed mode environment, routing groups must correspond to administrative groups.

**Administrative Models**

One of the big design change considerations in Windows/Exchange 2000 is how you will construct you administrative model. Many of the constraints imposed on you previously in Windows NT4 and Exchange 5.5 are removed. There are a couple of basic models for administration from which others (including some hybrids) are derived. They are distributed and centralized.

**Distributed management model**

This model is the most similar to the old Exchange site model. It allows complete control of the Exchange organization from company regions or divisions—hence the name “distributed.” You will see it most often in companies trying to change little as they migrate from Exchange 5.5, or where there are branch offices that operate independently. When it works well, a central group sets and maintains standards that are adhered to religiously by field personnel. While that group owns the standards, they do not own administration itself. In this model, administration is owned by personnel in the field. Administrators in the field are limited in their authority to administrations of their own group. Figure 23-1 shows an example of a distributed management model with administrative groups for each location.

**Centralized management model**

In an alternative model called centralized management, there is only one main administrative group at Corporate HQ and multiple routing groups (London, Delhi, Nairobi, and so on). In locations like Brisbane, there are no local administrators. In this model, all administration (except physical tape changes or reboots, if necessary) occurs at Corporate HQ. This model is illustrated in Figure 23-2.

Hybrids of the two models are common. In a mixed environment, you may choose to centralize just routing management or policy management. Windows 2000 allows extensive customization of administrative delegation so the options are endless. The important point here is that these decisions affect your design in tangible ways.
Figure 23-1: Distributed management model

Figure 23-2: Centralized management model
Permission and Policies

With regard to the effect of permissions and policies on topology and design, there are two major factors to consider. First, examine who is responsible for user management, administration and routing, data management, and real-time collaboration. Second, consider what accessibility is appropriate for different end-users and to what degree policies will be used to enforce access control.

User management

In most designs, the people involved in user management will also be assigned the responsibility for Exchange recipients. In Exchange 2000, the two are closely related, as both are Active Directory Service objects. You may wish to build into your design multiple administrative groups to provide for administration of different groups of users. The grouping you use could be anything but is often geographic or business unit based.

Administration and routing management

We already talked about routing groups and centralized versus decentralized administrative models. Depending on how you reconcile your needs to your plan, different permissions will be granted to accommodate different models. Most often, this will be accomplished by creation of administrative groups. In certain scenarios (such as a mixed environment), creation of administrative groups can affect your design for routing groups.

At the time of Exchange installation, unless you are joining an organization, your Exchange server will automatically disable administrative group creation. You have to enable that if you want to create groups.

Data management

Management of mailbox stores, public folder stores, and the Web Store can affect your design for physical placement of servers. For instance, it is hard to change tapes or do a hard reboot if you cannot touch the server. This could be an issue unless backup and restore responsibility is assigned to a different administrative group that happens to be local. Other issues that can affect your design for physical placement of servers are models for administrative groups, security, storage groups, and even mapping of software to the hardware. If you decide to have different people responsible for different data, you may have to build different storage groups and place them on different machines.

Real-time collaboration management

If your organization will use some of the new features in Exchange 2000 extensively (such as chat, conferencing server, instant messaging, or unified messaging) it is
likely that your decision will drive decisions about network topology, bandwidth, administrative groups, server hardware requirements, and mapping software to hardware.

**Other permission design considerations**

In Exchange 2000, you can specify user and group access by object class rather than by container, as in earlier Exchange versions. This means that you can be far more granular in your permission assignments if you choose. For instance, you might want to allow viewing status of an object, such as a mailbox or public folder, while not allowing size alteration.

**Policies**

Policies are used to implement a particular system behavior based on the needs of your designs and topologies more often than driving a particular design or topology. For instance, you might use recipient policies to create multiple e-mail addresses for a set of recipients by filtering for the recipients based on attributes, creating additional e-mail proxies, and applying the recipient policy. Similarly, in the Information Store, you might use system policies to manage a store setting for the sales department.

**Site and Administration Boundaries**

Exchange 5.5 sites and Windows 2000/Exchange 2000 sites are not the same. Nor does the Exchange 5.5 site correlate to a Windows 2000/Exchange 2000 domain. In Exchange 5.5, a site was a boundary for administrative, message routing, and network topology. In Windows 2000/Exchange 2000, a site is simply a network boundary usually determined by one or more IP subnets and available bandwidth. It is divorced from the notion of administration, because Windows 2000 and Exchange 2000 provide higher granularity and more flexibility in assigning administrative responsibilities. Windows 2000 and Exchange 2000 use routing groups and administrative groups to allow separation of these two administrative responsibilities. Given this, how should you assign site and administrative boundaries?

Only one Exchange organization is allowed per Windows 2000 forest, and Exchange 2000 organizations cannot span Windows 2000 domains. Since limited functionality exists at the moment for merging and splitting forests, it is imperative that your plan structures your forest properly so that your entire Exchange organization fits comfortably in one forest. When it does, the global catalog servers will contain all the information for the forest they reside in, but no information for other forests — even in the case where a trust relationship exists between the domains. Further, since directory replication does not automatically occur between forests, you need to utilize a third-party LDAP-based application or add Microsoft Meta Directory Service to handle the replication.
Because ADS uses site design to determine the best use of available network resources, the topology of your site, domain, and placement of your global catalog servers must be considered carefully.

If you are not clear on how ADS works, you should do a quick review of Chapter 6.

When a client makes a directory service request from a global catalog server, ADS will direct that request to a server in the same site and domain as the Exchange server that the client is connected to. Of course, that presumes that one is available. If not, the request goes out over your LAN or WAN to another server, delaying the response for that client and increasing traffic on your LAN or WAN. Both delayed response and increased traffic are undesirable.

At least one Global Catalog server (GC) should exist in each Windows 2000 site. Add more as load or redundancy require.

Taking this issue a bit further, the clients depend on the Exchange server they access. The Exchange server is prohibited from writing to any global catalog that is not in the same domain as the Exchange server. This is a significant departure from the way the Directory worked in Exchange 5.5. The best designs usually ensure they are in the same domain and have high-speed LAN connectivity.

Exchange servers that clients access should be hosted in the same domain as the users who will access them.

Active Directory Service and Replication

If you already have a team working on Windows 2000, they probably have an Active Directory design and topology. You or someone knowledgeable (and experienced, if possible) with Exchange 2000 will want to review and adapt it for Exchange 2000. While they may have done a great job for a Windows 2000 design, it is likely that they did not build a design and topology that is appropriate for Exchange 2000, unless deliberately trying to do that. Significant items, such as placement of global catalog servers, were probably fine for a Windows 2000–only design but wholly inadequate for most Exchange 2000 designs.

If you have legacy messaging systems, you will have to design a way to cause updates and changes to the legacy system directory to interchange with ADS. To accomplish that, you may need Microsoft Meta Directory Service, which is a third-party directory exchange product, or you may need to create your own using the LDAP programmatic access built into Exchange and Active Directory Service.
In cases where your legacy system is in NT and Exchange 5, you will have to do some housecleaning, which may affect your design during the migration period. Exchange 2000 requires a one-to-one relationship between NT user accounts and Exchange mailboxes. Your design must include this one-to-one relationship and a method to migrate any legacy system to Exchange 2000. For instance, if you do not have that one-to-one relationship already in place in your NT 4 and Exchange 5.5 system, you need to establish it before you proceed. See Chapter 18 for various methods to do that.

If you are migrating from Exchange 5.5, you will probably be running in mixed mode rather than Windows 2000 native mode. In this case, certain restrictions will be imposed on your design temporarily with regard to routing and administrative groups (covered elsewhere in this chapter) and account/mailbox mapping.

In a mixed mode environment, both a Windows 2000 and a Windows NT 4 account can access multiple Exchange 5.5 mailboxes, but not multiple Exchange 2000 mailboxes.

In a mixed mode environment, you will want to establish connection agreements in your design to give you the means to control ADC replication between an ADS domain controller and an Exchange 5.5 site recipient container. Each ADC can host multiple connection agreements. Refer to Chapter 18 for more information on how to use the ADC and connection agreements.

Public Folders

In your design, you should establish standards for naming and organizing public folders. Creating a well-defined folder hierarchy and corresponding levels of permissions and administrative delegation will really pay off over time. It is especially important to define and secure the top levels of the public folder hierarchies. We believe you should specifically limit permissions for creation of route level folders to the people who understand your plan for public folders and who will implement your rules.

You will also want to adapt your design and topology in consideration of volume and frequency of public folder replication. Patterns of usage will drive replication, location of servers, WAN bandwidth, hardware server sizing, backup and restore plans.

Other BackOffice Components

As BackOffice continues to grow, there is more and more integration of the various products. An example is using SQL Server to automatically generate an e-mail every day to some group with data content from a SQL database tied to a Web page used as a survey on a Web site. Some of the high integration relies on certain permissions.
Others work better with high or constant availability. You should determine what functionality other teams in your organization want from your messaging system for their BackOffice products or projects. Once you know what they are looking for, research the requirements and build the solutions into your design and topology.

**Message Routing**

The importance of the new link state routing model over the old GWART and RID model is simple — you can now sleep at night, knowing your messages are not bouncing back and forth between Singapore and Cairo. Not having to worry about messages caught in loops when two links go down means that you can build a cleaner connector topology. Taking advantage of routing groups and address space allows you to fine-tune your model.

*Note*

RAS goes away in Exchange 2000 and is replaced with RRAS, so adjust your design accordingly.

Little errors can have big consequences. Thinking that a “space” and an “*” are interchangeable can cause messages to be routed where one does not want them to go. Be cautious about your understanding, and more importantly, track some of your messages after making changes to make sure they are going to the appropriate place.

**Bandwidth Management**

You will find that the effort that went into Exchange 2000 to help you diminish bandwidth requirements and help manage them delivered tangible results. Both changes in the routing model and the ability of the directory to replicate at an attribute level rather than at an object level make a real difference. If you have a legacy messaging system running across your WAN, your design should take into consideration the increase in efficiency and balance that off against the new features that will add to the load. Running a pilot, where you test usage patterns, will help you get the initial design right. Using link monitors, performance monitor (queue monitors, and so on), and end-user surveys will help you assess the impact.

**Money, Security, Service Levels, and Policies**

Once you have accumulated enough organizational and technical information to work your topology and design, you need to reconcile your plan to the budget, your organization’s requirements for security, and the planned or committed service
levels for users and business units. Any of these can radically alter both topology and design. Hand-in-hand with service levels are the policies for usage and administration that your company both sets and enforces. While policies and the enforcement of them do not usually alter topology, they can alter design or the success of service and administration.

We like to see policies governing creation, usage, and administration. Examples include

- Appropriate uses of your messaging system
- E-mail etiquette (reply all, use of capitals, flame mail, retention of e-mail)
- Creation and use of various recipient types
- Creation of public folders, replication of public folders, archiving
- Mailbox storage
- Forwarding of e-mail to and from other accounts
- Virus protection configuration on clients
- Ownership of e-mail (does it belong to the company or the individual privacy policies?)

While there are many other appropriate areas for policies, you get the idea. Administrators cannot enforce policies that do not exist—or policies that are obscure, illogical, or confused. Nor can they enforce policies that are not widely understood and subscribed to by management, users, and other administrators.

Do not misconstrue our view of the administrator’s role. We are not advocating that administrators become policemen or policewomen. Administrators do not wear badges and generally do not want to. Management and internal security teams are responsible for most enforcement in most organizations.

Requirements for redundancy, physical security, protection from hackers, and budgets may require you to get creative and design differently. While it is somewhat true that you only get what you pay for, it is also true that employing a deft understanding of the product, the technical issues, and the needs of the organization often allows us to work within budgets and deliver the service levels and security the organization needs. If you cannot deliver all of them, you must address the issue head-on. One technique is to create one design with what you think is the best balance of all the constraints and requirements, then create one or two more showing what can be done with the proposed budget and how it affects security, service level requirements, business utility, and deployment schedule. Business people make decisions every day about money based on facts they can understand and the counsel of their trusted advisors. You need to present to them in a way they can understand. The decision is ultimately theirs. That’s what they get paid the big bucks for.
Summary

In a sense, this entire book is about items that affect how you design and what your topology should be. Yet some issues and foci stand out. You start with an idea of how your company is organized and build your administrative model from that. You find out what patterns of messaging and collaboration usage exist now and what is expected, and you build your design around that. You take into consideration the coexistence with and migration from legacy systems. You think about how other applications, like other parts of the BackOffice suite, will fit in. You carefully choose names, assign server roles, decide where your data will reside, specify replication and frequency. Then you adjust all to fit a budget, security model, and service level. Once you have your messaging system designed, all you have to do is build it.
Exchange on the Internet

The Internet is now the nucleus of almost all messaging and collaboration systems. No other area in Exchange 2000 received as many enhancements as the integration of Exchange 2000 into the Internet. In this chapter, we look at what it means for the Internet to be at the heart of most Exchange 2000 capabilities and how you can take advantage of those capabilities. We begin by examining the use of the Internet as an Exchange backbone, followed by a discussion of how Exchange servers can communicate through the Internet to each other or to other companies, mailbox recipients, contacts, and individuals. Then we look at utilizing the Internet for Exchange client/server access. Finally, we review some specific considerations for administering and using Exchange on the Internet.

The Internet as an Exchange Backbone

Using the Internet as an Exchange server backbone is not a new concept. The original electronic messaging protocol, Simple Mail Transfer Protocol (SMTP), was built to use TCP/IP, and—as it evolved—the Internet, for the transport of messages. In NT 3.x, 4.x, and especially Windows 2000, Microsoft migrated the default transport protocol for NT from NetBEUI to TCP/IP and added Routing and Remote Access Service (RRAS), Virtual Private Networks (VPNs), and security capabilities to expand the scope of its use. In Exchange 4, 5, and 2000, Microsoft added support for Internet protocols and took more and more advantage of Window’s underlying migration toward TCP/IP and Internet-readiness. Now you can readily use Exchange to tie your organization into others on the Internet, communicate between parts of your organization over the Internet, or use the Internet as a backbone for additional protocols, such as IMAP4, POP3, and NNTP for additional messaging and newsgroup capabilities. Two
electronic messaging standards commonly utilize the Internet as a backbone transport for messages: SMTP and X.400. The SMTP protocol is based on the TCP/IP model, and its evolution as a standard is under the jurisdiction of the Internet Engineering Task Force (IETF), which is accessible online at www.ietf.cnri.reston.va.us/. For more information, see Chapter 3.

The working of TCP/IP and SMTP backbones is best understood in the context and structure of the ISO/OSI model. The model for communication between a client and a server on a network is known as the Open System Interconnect (OSI) model, and the International Standards Association (ISO) has adopted it as a standard. This model shows the interconnection between devices as made up of seven layers, as presented in Figure 24-1. Each layer provides services to the layer above it. The seven layers of the OSI model are as follows:

- **Application layer**: Provides network access to applications.
- **Presentation layer**: Ensures that both senders and receivers of messages use the same data format, such as UNICODE, and also includes any encryption or compression that may be required.
- **Session layer**: Responsible for establishing a communication session. All messages between senders and receivers occur within a session.
- **Transport layer**: Responsible for the quality of service of the communication system. It ensures that messages are not lost or duplicated.
- **Network layer**: Decodes addresses and routes messages appropriately.
- **Datalink layer**: Packages message data into frames, controls the flow of frames from source to destination, acknowledges the receipt of transmissions, and retransmits if necessary.
- **Physical layer**: Defines the hardware, including the terminal devices the male and female connectors used, the pin configuration, and the transfer of the data as a serial bit stream.

The TCP/IP model for the Internet, which can be viewed as a subset of the ISO/OSI model, is composed of a stack of protocols sitting on top of one another and defining access to the Internet. Figure 24-2 shows the protocol stack for TCP/IP for the Internet, which consists of four layers:

- **Application layer**: Provides Internet clients and servers with network access and ensures that there exists mutually understandable standards for representing data.
- **Transport layer**: Establishes connections between clients and servers or virtual circuits through which clients can connect to servers. Responsible for end-to-end delivery of messages.
- **Internetwork layer**: Responsible for routing messages.
- **Network Access layer**: Responsible for transmission and receipt of data frames. Utilizes the services of the physical network.
**Figure 24-1:** The OSI layer model

**Figure 24-2:** The TCP/IP protocol stack for the Internet
The TCP/IP stack for the Internet includes several protocols that belong to three of the layers:

- **Application layer**: Includes e-mail protocols SMTP, IMAP4, and POP3, as well as FTP for file transfer and HTTP for allowing browsers to retrieve data from Web servers. This layer also includes the Network News Transfer Protocol (NNTP).

- **Transport layer**: Consists of two protocols: TCP and User Datagram Protocol (UDP). These protocols define the type of message context. TCP is a connection-based protocol, while UDP is a connectionless protocol. UDP is less reliable because it doesn’t guarantee delivery of messages after a connection is established.

- **Internetwork layer**: Consists of the IP protocol. IP, like UDP, is connectionless and responsible for data routing. IP uses routing tables to determine where to forward messages so that they can use the shortest route to reach a destination.

To use the Internet as a backbone, you use your applications and Application layer protocols on top of a Transport layer on clients and servers under your control. Application and transport protocols sit on top of the Internetwork layer, which lies predominantly with your ISP and others that have transmission lines, servers, and routers that are beyond your control and make up the Internet. Take a closer look at some of the Application layer protocols utilized by Exchange 2000 and the Internet Information Server (IIS).

- **SMTP**: The native messaging protocol of the Internet, as defined in the RFC822 standard. SMTP consists of a small set of commands that facilitate the transfer of mail between SMTP-compliant systems. The commands, in alphabetical order, include DATA, EXPN, HELO, HELP, MAIL, NOOP, RCPT, RSET, SAML, SEND, SOML, TURN, and VRFY. The SMTP protocol is typically used by messaging clients applications (especially POP3 and IMAP4) to send mail.

- **POP3**: A popular client protocol for receiving e-mail comes in three versions: the original POP, POP2 that requires SMTP for electronic messaging, and the latest, POP3, which doesn’t require SMTP (but applications of POP3 often couple POP3 with SMTP where POP3 receives and SMTP sends). Clients use POP to receive messages from a remote server. POP3 consists of ten core commands that allow a POP3 client to download mail for offline use. POP3 is defined in RFC 1088.

- **IMAP4**: Similar to POP3 in that it allows a messaging client to connect to a messaging store remotely. It is different in that it allows a messaging client to access their mail without downloading it to their computer. To enhance the capabilities of remote access to e-mail, IMAP4 provides additional features, such as search by keyword, access to Public Folders, access to more than one
e-mail folder, view message headers before downloading, download specific messages, download an attachment separately from a message, and flag messages as unread or read. IMAP4 is covered in RFC 1730.

✦ **NNTP:** Facilitates posting, distributing, and viewing news items and group discussions between NNTP clients and news servers. The standard for NNTP is RFC 977. NNTP includes commands that allow access to newsgroups posts that are managed by a USENET provider such as UUNET or MCI.

In addition to the protocols specified above, Exchange 2000 and IIS use the Domain Name Service (DNS) of Windows 2000. The DNS server provides resolution from a fully qualified domain name to a specific IP address. DNS is based on a standard, RFC 1035. You can think of DNS as a virtual telephone directory that can resolve a domain name to a specific IP address. If the local DNS server cannot provide resolution, it will contact other DNS servers for help.

Exchange 2000 server running on a Windows 2000 server has expanded the use of its backbone capabilities over the Internet beyond clients accessing e-mail servers using traditional protocols for reading, downloading, or sending e-mail messages. Enhanced capabilities include browser-based access to e-mail, such as Outlook Web Access. It also includes server-to-server message routing, and server-to-server and client-to-server newsgroups, WebDAV, and XML. In this section, we focus on the specific capabilities of the server.

**Message routing**

Exchange 2000 servers can route messages in one of four ways: on the same server, in the same routing group, between routing groups, and to a server outside the Exchange 2000 organization.

Routing messages on the same server is pretty straightforward. The message is delivered from the outbox of a mail sender to the inbox of a mail recipient. Very little routing occurs here.

In Exchange 2000, the concept of a routing group was introduced. A routing group, like a site in Exchange 5.5, is a collection of well-connected servers that can communicate over fast, permanent, and reliable connections. The second mechanism for routing messages is to deliver in the same routing group. Delivering a message from one Exchange 2000 server to a recipient on another server in the same routing group can utilize either an SMTP protocol or a Remote Procedure Call (RPC) communication for delivery. If both sender and recipient are on an Exchange 2000 server, SMTP is used; otherwise, RPC is used.

If you have two servers connected by the Internet and you decide to add them to a routing group, you are using the Internet as a backbone for your routing group. This is only a good idea if the Internet connection is fast, permanent, and reliable.
Your Internet connection may not be fast, permanent, or reliable enough for a routing group backbone. For example, if you are using ISDN, the channels could be “nailed” up or connected on demand. The latter is less like what a routing group would expect than the former. With slow or unreliable Internet connections, you should expect routing group connector problems and use an alternative type of connector like SMTP or X.400.

Message routing becomes more complicated when delivery is between different routing groups. Delivery, in this case, may require multiple hops through additional intermediate routing groups. When an Exchange 2000 server has to send a message to another routing group, it identifies a route to deliver the message. Then, the server uses one of three routing connectors—Routing Group Connector, SMTP Connector, or X.400 Connector—to forward the message to its destination.

The Routing Group Connector is used when both message groups are Exchange 2000 routing groups. The Exchange 2000 SMTP Connector is used between an Exchange 2000 and an older version of Exchange server site. The X.400 Connector is used to connect between an Exchange 2000 Routing Group and other servers. Either can use the Internet as a backbone. Both are less susceptible to slow speed or unreliable links, often making the Internet an economical and appropriate choice.

Make sure you understand how to build the connectors correctly to maintain appropriate security. See Chapters 10, 11, and 22 for more information.

Finally, the most complicated routing is to messaging servers outside the organization. In this case, one of two connectors can be used: the Exchange 2000 SMTP Connector or the X.400 Connector.

**Exchange 2000 SMTP Connector**

The Exchange 2000 SMTP Connector is a new version of the Internet Mail Service that was available in prior versions of Exchange. This connector allows SMTP connectivity between servers and/or uses an SMTP server as an intermediate hop between servers.

Since all messages are transferred in ASCII, we discourage the use of SMTP Connectors over the Internet for security reasons. X.400 is a better choice.

**X.400 Connector**

The X.400 Connector can be used both between two routing groups as well as to an outside organization. You can use an X.400 Connector to transmit messages to other X.400-compliant messaging servers across the Internet. X.400 is preferable when there is a low bandwidth between the two organizations. It also provides more security.
In Chapters 25 and 26, we go into more detail on both SMTP and X.400 Connectors and how to use and administrate them.

**Newsgroups**

Exchange 2000 supports the NNTP protocol in order to allow Exchange 2000 server administrators to set up newsgroups. A newsgroup is a collection of messages posted by individuals as part of an online asynchronous discussion. The messages are posted using the Internet and distributed by USENET. USENET is the assembly and distribution point for all newsgroup messages. Unlike Exchange 5.5, the NNTP service is built in to Windows 2000. When Exchange 2000 is installed, the basic Windows 2000 NNTP service is enhanced with two additional capabilities: support for newsfeeds and support for a master/slave arrangement. Both are commonly used across the Internet.

A newsfeed is basically a collection of newsgroup articles that get replicated from one NNTP server to another. This replication can either be a push from one server delivering to another or a pull when one server draws articles from another. Exchange 2000 can participate in both.

To promote the reliability of NNTP servers, Exchange 2000 makes it possible to create a master/slave configuration for NNTP that would allow for the slave server to take over when the master server is unavailable. See Chapter 11 for more detail on how to administer NNTP.

**WebDAV**

Exchange 2000 introduces WebDAV as an extension to the Hypertext Transfer Protocol (HTTP) version 1.1. WebDAV facilitates file operations, such as copy and move, as well as document tracking over the Internet.

As we mentioned earlier, WebDAV is an extension of HTTP. So what exactly is HTTP? HTTP is the universal Web standard that allows information to be formatted and communicated over the Web. It is the mechanism by which Web servers and clients respond to commands, such as GET and POST.

WebDAV extends HTTP with a set of commands that allow clients to access documents and collaborate over an HTTP connection. As a result, WebDAV resources can be published on an IIS virtual directory and accessed by browsers that support the WebDAV protocol. For example, an Internet Explorer 5.0 client can access a WebDAV directory and its content as if it were accessing a directory on their own computer.
Some of the important WebDAV features include file-locking, access to resource properties, and file manipulation. WebDAV is a standard based on RFC2518. Exchange 2000 extends the features of WebDAV to include additional capabilities, such as extended search in WebDAV resources, notifications when new messages arrive to a WebDAV folder or changes are made to messages, and access to a WebDAV calendar for creating new entries or reviewing entries.

**XML**

Extensible Markup Language (XML) is an extensible version of HTML and a descendant of SGML. XML is being touted as the next “coming” of data exchange. XML allows users to define their own tags and thus create a self-defining medium of exchange for data.

In Exchange 2000, XML is deeply integrated into the use of WebDAV, in the way it supports the extensions to HTTP and in the content of the resources that are being shared over the Web.

**Client/Server Access over the Internet**

You can access e-mail through a client application or through a Web browser across the Internet. Using a regular client, such as Window’s Outlook 2000, you can configure the servers that you want to connect to in order to download or read your e-mail messages. Similarly, you configure the servers that will be used to send your messages toward their destinations. If you are using Outlook 2000 configured for Internet access as your Exchange 2000 client, go to the Tools menu, select Accounts, and a dialog box will appear, as shown in Figure 24-3.
To add an e-mail account, select the Mail tab, click the Add/Mail button, and the Internet Connection Wizard appears, as shown in Figure 24-4.

![Figure 24-4: The Internet Connection Wizard](image)

In the next screen of the Internet Connection Wizard, enter your name and click Next. The screen shown in Figure 24-5 appears.

![Figure 24-5: Internet E-mail Address screen](image)
Enter your e-mail address and click Next. The screen shown in Figure 24-6 appears.

![Figure 24-6: E-mail Server Names screen](image)

In this screen, enter your server names for incoming mail and the SMTP server that will be used for outgoing mail. Click Next and the screen shown in Figure 24-7 appears. These servers can be on your local LAN, or they can be anywhere on the Internet. If your DNS and your ISP’s DNS cannot resolve the names, or they require authentication or security credentials you do not possess, or even if there is a firewall obstruction with critical ports closed, you may not be able to connect. On the other hand, if everything is accessible and you authenticate, these servers can be anywhere on the Internet backbone and you will be able to access them.

In the screen shown in Figure 24-7, enter the account name that you will use to access your mailbox account and the password you will be using. Also specify whether you will be using secure password authentication by use of the check box. Choosing this feature depends on whether your e-mail server has been set up to support secure password authentication. Clicking Next brings you to the screen shown in Figure 24-8.
Chapter 24 ✦ Exchange on the Internet

This screen allows you to choose your Internet connection method. Clicking Next brings you to the last screen of the Wizard, shown in Figure 24-9.
Once your e-mail account is set up in Outlook 2000, you can use the Tools menu Send/Receive item to send and receive your e-mail from the account you specified.

If at any time after you set up your account you need to change its properties, you can always bring up the account (shown in Figure 24-3) and click the Properties button. A Properties dialog box with four tabs will appear, as shown in Figure 24-10.
You can use the four tabs—General, Servers, Connection, and Advanced—to change the setup of the e-mail account that you are accessing.

Another way you can access your e-mail from the Internet is by using a Web browser, which was introduced with Outlook Web Access (OWA) in Exchange 5.0. Exchange 2000 introduces a redesigned OWA. OWA consists of a server component that publishes Exchange 2000 mailbox content in a format that can be accessed from a Web browser across the Internet. The new OWA provides Web browsers with the same capabilities as previous versions, including accessing e-mail, Public Folders, calendar, creation and management of password changes, name checking, and conversion of Outlook forms to HTML.

In addition, OWA 2000

✦ Offloads processing from the server to the client
✦ Enhances Public Folder capabilities to include contact and calendar information
✦ Enables messages to contain multimedia content and other embedded items, such as appointments and ActiveX objects
✦ Uses named URLs to refer to other Web-based resources

You can get the most advantage out of OWA if you use Internet Explorer (IE) Version 5.0 or later. Using IE 5.0 or later plus OWA 2000 implements use of Dynamic HTML (DHTML) to create such features as an expandable tree-view for public and private folders, a drag-and-drop capability, right-click context menu options, and the ability to process many Outlook requests on the client browser rather than send all the requests to an IIS server.

On the client-side, the OWA is displayed as two frames. The left frame is used for navigation, and the right frame for rendering pages. When a user connects to Exchange using OWA, a Logon screen with an address, an account, and a password is displayed. IIS authenticates the user, and OWA on the server returns the page containing the two frames to the browser.

The user can either access OWA through an open server using HTTP as a protocol, or through a secure Web server using HTTPS. The HTTPS protocol utilizes the Secure Socket Layer (SSL) protocol for security and authentication.

**Special Considerations**

Many of the enhancements in Exchange 2000 have opened information to access through the Internet backbone. Providing these capabilities presents a conundrum. On the one hand, the expanded availability of information means that the Exchange 2000 servers and the IIS servers will see more use than ever before, and
in many cases, more and more ease of use. On the other hand, the increased use creates administrative issues of scale and security that are new to many organizations used to somewhat isolated messaging systems. To address issues of security and scalability, Microsoft provides two additional servers in the Windows .NET server family:

✦ **Internet Security and Acceleration (ISA) Server 2000**: ISA 2000 provides secure, fast, and manageable Internet connectivity through the integration of an extensible, multilayer enterprise firewall with a scalable high-performance Web cache.

For more information, read *Microsoft ISA Configuration and Administration*, published by Hungry Minds, Inc.

✦ **Application Center (AppCenter) 2000**: AppCenter 2000 provides an easy tool to manage and balance the workload of multiple Windows 2000 servers to maximize throughput and minimize turnaround time for client requests.

For more information about these servers, visit the Web site at www.microsoft.com/servers/net/default.htm.

**Summary**

This chapter provided an overview of how Exchange 2000 integrates into the Internet. With stronger protocol support for POP3, IMAP4, TCP/IP, SMTP, and NNTP, greater and easier integration is achieved. With NNTP, enhancements using newsgroups becomes easier than ever. As each new generation of Exchange is delivered, electronic messaging becomes more and more a part of knowledge management, since a key feature of knowledge management is to share knowledge with others. Today, that translates into “share with others over the Internet” and use the Internet as your corporate backbone. For many, Exchange 2000 is “The” messaging and collaboration platform product for sharing knowledge over the Internet.
Administration and Support

W hen Exchange 2000 Administrators and consultants are not installing and deploying, they are often performing administrative tasks to preempt failure and to ensure service levels for the business communities they serve. When they are not doing that, they are often troubleshooting problems that arise. In this part, we discuss how to ease the administrative burden by focusing on how Administrators can preempt failures. We examine the more important Exchange and Windows 2000 utilities. We provide and explain checklists for basic administration. We introduce some very important procedures, such as backup and restore, as well as areas of special attention, like wireless, in Chapter 28. Finally, you get a glimpse of real-life problem-solving processes in Chapter 29.
Easing the Administration and Support Burden

Administering a robust, collaboration and messaging, enterprise-wide environment can be challenging. In the process of building Exchange 2000 and Windows 2000, Microsoft addressed many issues with the Administrator in mind to keep the task of administration simple. In this chapter, we discuss topics that fall outside the control of Microsoft but in your area of responsibility. Following these techniques and ideas can help make order out of chaos and lower both the total cost of ownership and the anxiety level of the Administrators in your organization.

Documentation

Documentation is the process of memorializing ideas or communication, usually in some written form. Good documentation includes the ideas of presentation for clarity, organization, and accessibility. Many Administrators are reluctant to take the time to document because they get little support for documentation and many kudos for fixing things. However, many of the best Administrators and consultants will not engage in a project without committing themselves to thorough documentation. How do you get from one reality to the other? By having a greater understanding of how documentation helps you and what types of documentation are worth creating. Documentation is all about lowering the administrative burden. For example, with good documentation, team members can retrieve information about events they were not
Planning and design

Lowering the total cost of ownership and the administrative burden starts with planning and design. There are two aspects to this effort: what you plan and design (covered in Chapter 23) and the documentation of your plan and design (including its iterations). In fact, documentation is an ongoing process, which will be carried out not only during the design and implementation stage, but even during the day-to-day running of your system.

It goes without saying that your plan must be a written, living document that changes as the design and plans iterate and continues as your messaging system evolves. It is a document that should be maintained by the organization for the life of your Exchange implementation.

It never ceases to astonish us how few organizations build their own written plans. Yet that is the first thing that most outside consultants will insist on. It also amazes us that some organizations pass random documents off as their “plan.” If the document does not have starting and ending points, steps to get from one point to the other, a schedule, budget, and contingencies, it is not much of a plan — maybe not even a plan at all.

Don’t be fooled by those that say, “We’re too busy to document” or “We’re doers, not writers” or “Nobody reads this stuff anyway.” Without a real, iterative, current plan — and stakeholders to support it — it will be more than difficult for the support staff to deploy and support the environment with quality. It may, in fact, be impossible.

Here are the components we like to see in any plan. (Your plan may have others as well.)

✦ Executive summary
✦ Mission statement
✦ State of existing environment, including existing standards, physical and logical topology
✦ Business utility required
✦ Ground-level view of the end-point environment, including proposed new standards, physical and logical topology
✦ Roles and responsibilities for internal and external resources (see Chapter 17 for specifics on the roles we suggest you consider)
✦ Skills assessment and knowledge acquisition (this is the training plan)
Chapter 25  ✦ Easing the Administration and Support Burden

✦ Development plan and schedule
✦ System integration plan and schedule
✦ Deployment plan and schedule
✦ Support before, during, and after deployment plan (more on that below)
✦ Budget

All sections need to be written at such a level of granularity that a responsible person could understand the detail and execute successfully off the document. This does not mean that you should spend all your time writing. Skilled internal or external consultants committed to documentation can keep the document up-to-date as long as the input streams of information are in place. Often, technical personnel who don’t have consulting skills hate to write and therefore do not write. But all consultants worth their salt know that consulting delivers a work product that almost always includes a written document, and often, an oral presentation to accompany it. Because of that, consultants tend to be more comfortable writing.

This is not an aspersion about the relative value of consultants versus Administrators. We value both—as do all great organizations. Just as many Administrators are not comfortable writing, many consultants are not comfortable with their hands on the keyboard. And some folks are a blend of consulting and administrating skills.

If you can, combine your written document with frequent oral presentations and meetings to ensure that it’s understood and accepted by others in the organization; this will also help ensure that the document remains a “living document.” Implement a system that will help maintain revision levels of the document. And save some back versions to be able to explore the history of changes.

If internal or external consultants are not available to you, use a technical writer—preferably one who is skilled with plan documentation.

Tip

Note

Ongoing support

It’s not uncommon for an organization to build masterful plans for their design and deployment but have little or no documentation for the day-to-day support of the organization. It is also common for whatever documentation there is to be discontinuous from the original design and deployment documents. In almost all situations, you want your planning documentation to address some of the issues that your organization will face when it gets to “steady state.” You also want the organization that’s responsible for support to iterate those documents and make them their own. This handoff can be challenging, since the group that does the design is often not the same as the group that does the support. If you already read Chapter 17, you know that we strongly recommend that you include support staff in your deployment field teams and in your entire planning and deployment process. That makes the handoff easier. Another way to ensure an easier handoff is to have the
team that will own day-to-day support deeply involved in the documentation effort, either writing or editing and signing off.

Your ongoing support plan should include sections that deal with these topics:

✦ **Needs-based training plan:** What skills do the various constituencies (Administrators, users, troubleshooters, and so on) need, and how do they acquire them?

✦ **Service levels:** What are the targets for service levels? (These should be set from a business perspective and should not be guarantees, but rather targets that are used for metrics.) What communication is provided to the business units?

If you miss your targets, either the targets are wrong, something extraordinary occurred, or your team is not giving their best effort to deliver at the targeted level. Although the first two reasons are more common than the latter, you will want to remember that if it is a case of your team’s lack of effort, assessing blame is not the same as assessing the situation and working together to effect a solution. For your documentation to iterate accurate, and for you to have a healthy organization, you must allow people to make mistakes.

✦ **Escalation:** Who is told what when and how are they told? Forms of communication include phone, beeper, e-mail, presentation, video conference, report, memo, and so on. Careful statements about frequency of communication and what to do if you cannot get through are crucial.

✦ **Trouble ticket database:** Having a collection point for issues and resolutions is important. Ensuring that the information in that database is reviewed and corrected is equally important.

It is much more difficult than you’d expect to make sure that the information in your database is correct. Encouraging your team to close calls often causes the information to be more inaccurate than normal. Calls should not be closed until the business unit signs off, and documents should be reviewed regularly by a knowledgeable, independent (for example, not the originators) staff.

**Standards and conventions document**

When an organization is committed to the standards process, change is iterative, controlled, based on as much knowledge as is available, deliberate, and usually better understood by both the implementers and by the people affected. A good implementation of the standards process makes some room for deviation and for grandfathering without loosing its grounding in sameness. While events may occasionally cause surprises, what you and the team do about unexpected events is pretty well established and protects the organization’s short-term and long-term goals.
All organizations should have a standards and conventions document (hereafter referred to as the standards document) that is also a living document. However, its change cycle should be less frequent than the support documentation, and possibly even the design and deployment plan. Often, a standards document has minor changes made quarterly and major changes once a year. Both the document and the process that creates it are important enough to examine.

In producing a standards document, you may find some items—like selections on property pages—challenging to document, which may cause us to ignore them. It seems to us that human nature tends toward the fast and easy. That’s where screen capture comes into play, since it allows for fast, easy documentation. On the CD that accompanies this book, we have included an evaluation of a screen capture program we use to capture most of the screenshots in this book, called HyperSnap.

**Standards and Conventions Process**

Good standards and conventions documents result from a sophisticated process with representation from many parts of the organization—yet, the process is rarely democratic. If the process is democratic, people usually vote based on what they already know, do not understand the technology or business needs of the organization well enough to have an informed opinion, or cause the process to proceed so slowly that technology is always way ahead of the standards committee. Pure democracy will not deliver the best business choices for your company. Nor will a fascist approach, where large numbers of stakeholders are frozen out of any decision making, education, and understanding. The best process is one that:

✦ Establishes a number of principles based on the business and technology needs of the organization
✦ Grandfathers in non-compliant technology
✦ Provides a path to get from the old to the new
✦ Insists that all money spent is spent in the direction of the new standard (except where explicit exception is allowed)
✦ Educates the stakeholders driving agreement

A good process prevents over-indulgence in technology issues devoid of business value, randomness and difference without compelling business reason, and excess disruption of business. It also forces out individuals who do not set the needs of the company ahead of their personal interests.
The standards document should be detailed enough that two able Administrators separated in two different rooms, given the same components and your standards document, would build your messaging environment the same way. The standards process, illustrated in Figure 25-1, helps your organization in a number of ways.

- It invariably lowers your total cost of ownership.
- It limits the number of products, revisions, and implementations you have to test.
- It ensures that the team is focused on the same solution.
- It allows you to limit the scope of your support documentation and still cover the appropriate topics.
- It gets issues in front of the decision-makers and planners that would otherwise be buried in the field.

**Figure 25-1:** A process for standards and conventions
General policies

We recommend that you establish business partnerships with a small number of key vendors. Buy the products these vendors sell unless there is a compelling business reason not to. Note, we did not say “technical reason.” There will be technical reasons, but in order to be valid in this process, they must translate to compelling business reasons.

If the model for administration is to support whatever anyone decides to buy without any limitation or control, then the administrator’s job is impossible. In that type of situation, management needs to redefine the job.

If existing policies increase the administrative burden, they need to be considered in that stark light directly. Sometimes the policies are popular. One that comes to mind immediately is the “best of breed” model, which has proponents everywhere. Can anyone argue against “best of breed”? Sometimes we do advocate for a product that doesn’t have the praise of the pundits for all the latest bells and whistles. We reason that, except in rare instances, people do not use all the features of a product. Smart companies tend to add features to their products within a year of seeing them in competitor products. In addition, there are other considerations, such as buying from a few vendor partners. “Best of breed” can fragment your purchasing power and your vendor relations. The policies you will consider fall into three main categories: physical, logical, and process. They are illustrated in Figure 25-2.

![Figure 25-2: Standards and conventions components](image)

Physical

Most companies have little standardization at a physical level. They already have Local Area Networks (LANs) and Wide Area Networks (WANs). Many are not controlled by a standards committee, or if they are, at least not by the same standards groups that own the responsibility for other parts of the messaging environment. We have seen many large companies that seem to approach standards like the guy in the old joke: “Of course we have standards. We have plenty of standards. I think we get a new one every time we buy something.” We have also seen single
customers with LANs that include 10baseT, 10base2, token ring, 100baseT, and Gigabit Ethernet. It gets even more complicated when you get to the WANs. Their desktop boxes cover a spectrum of utility, but often go much farther and look like the results of the sale du jour. Servers often also come in all stripes and colors.

While it is sometimes impossible to have only one brand, model, or transmission standard because of the legitimate needs of the business (international availability and vendor model changes are two examples), the standards process and document should strive for fewer brands, models, and standards rather than more. It should provide the best upgrade path for the procured equipment (where “best” means “most economical” relative to short- and long-term). An example of the difference between a good standard and a mediocre standard comes from specifying memory for PCs. Often we hear customers say they specified 128K, but that’s not enough. You want to specify the size of the DIMMs or SIMMs so you know what your options are for adding memory later. If you don’t specify, the vendor will sell you whatever it has or whatever is cheaper, and you may not have any open slots for expansion.

We did one engagement where getting an answer to the question of how much memory was in each desktop meant opening thousands of boxes one by one. This predicament forced the customer to stay with Windows 95 even after they wanted to go to Windows NT. It was a budget-buster.

You will make administration easier if you limit the physical profile of your messaging organization, because there are fewer different types of equipment to deal with. How can this be done? We suggest you commit yourself to supporting as few physical differences as required to support the messaging enterprise. We further suggest that you

✦ Engage the individuals who set policy for depreciation
✦ Depreciate hardware over shorter terms for fewer legacy models to deal with
✦ Discuss trade-up policies with your vendors
✦ Look at leasing with a refresh
✦ Plan for lifecycle, including retirement, at the time of purchase
✦ Purchase software maintenance and upgrades so you can stay current

None of these suggestions are panaceas. Nor is this an exhaustive list. Our suggestions merely provide a sense of direction and some tangible ideas for your standards. None of these topics are easy for an organization to address, but the only way to drive down the administrative burden is to get some control as an organization over the number of differences that face your Administrators.
Logical

Standards and conventions for the logical aspect of your messaging environment include all settings and configuration variables, naming conventions, routing, service role assignments, and overall logical topology. Each of these requires definition so that they can be replicated when successful and iterated when problems arise. With standards in place, an Administrator (even a new one) can use standards documentation to eliminate differences between what should have been built and what actually was.

The best way to get your configuration standards documented is to insist on documentation while installing and configuring, especially during the pilot.

Procedural

Standard operating procedures are common in big iron environments. They are also common in standards-based environments that are trying to ease the workload of the Administrators. To make this tradition work in your environment, ensure that all common practices are based on established, written procedures that answer the following questions:

✦ How is the procedure done?
✦ How does one propose changes? How are they reviewed, accepted, ratified, and adopted?
✦ If the procedure can’t be completed, how do you fall back and restore business utility that was there before you started?
✦ How are problems escalated?
✦ Who do you communicate to? How frequently? Using what form?

When dealing with uncommon practices, similar techniques are used and the same basic principles apply. Make sure you engage your best people during uncommon situations, since they’re the ones who know how to apply the drills.

Application development

In a messaging environment, application development tends to go one of two ways. Either it is ad hoc-driven, with anyone who has access and desire doing development, or it is controlled and deliberate. Though a combination of both methods is common in organizations, controlled development is easier to administer. Establishing standards for either scenario is possible and desirable. Enforcing them in a controlled environment is more likely to succeed. Even when people ignore the standards,
having them in place (documented and well-understood) is important because you can tie service levels to adherence. If other players violate the standards, their service levels may be adversely affected, since the service levels were established for the standard environment.

Pay special attention to issues like managing revision levels of forms, including distribution.

**Administrative Assignments**

Assigning the right roles to the right people and ensuring they have the skills to do the job is a basic task, but still difficult to achieve with fast-changing technology, shortages in the workforce, and limited time for training. In the world of Exchange 2000, you have far more flexibility to assign roles broadly and still manage your enterprise securely. Assigning jobs to individuals that have sufficient, but not excessive, skills will help relieve the workload of Administrators with advanced skills. It is not prudent to have senior engineers or Administrators doing simple user adds and changes, yet that is what we see in many organizations. Your ability to manage this issue will go a long way toward helping you ease your workload.

A creative way to help your team gain knowledge and experience, while controlling costs, is to assign them to participate in pilots and deployments. This can provide experience with the products and your standards required for later support. If you are using consultants and system integrators to help, some should be doing back fill for your team and others should be assigned to help with pilots and deployments. If they are involved in pilots and deployments, they should be transferring knowledge and talent to your team in a somewhat formal way as fast as your team can absorb it.

**Choosing a candidate**

Finding the right candidates for the various jobs in your messaging system requires foresight. Think of it in terms of trying to build a sports team. You can hire some free agents, but not an entire team of them. There will be some veterans and some rookies. You have a salary cap. And you want to win day in and day out. To succeed, you need to develop the talent you have and also attract new talent. You need to choose the right candidates in the draft. You need to know what roles you want them to fill and be able to assess their talent. Once you have that figured out, you need to get them signed. In a standards-based environment, you will find it easier to define the roles you want candidates to play and to match their talents to the job than in a non-standards environment. From the standard itself, you should be able to construct assessment tests that measure the individual’s skill level. From your procedures, you should be able to ascertain how well this individual fits into your organization.
Your human resource department will help you make sure that, in an effort to assess skills and adhere to standards, you do not discriminate or act in any unfair, inappropriate, or illegal way. This is especially important in international settings, where you may not be familiar with cultural norms or local laws.

Assigning a backup

Your assigned employee may go on vacation. She may be out sick, out for training, or decide to leave your employment. In such cases, who does the work? Be sure to establish a backup and ensure that they get enough hands-on experience to do the job. They also need keys, passwords, and other tools to get through security.

Backup and Restore

There is probably no more important topic than backup and restore. The stress on Administrators when information is unavailable to end-users is more than burdensome — it can be absolutely devastating. To diminish this possibility, you will want rigorous standards and conventions for backup and restore. Then you want to ensure that you run regular drills demonstrating your prowess and proving that the policies, procedures, team knowledge, and equipment work. It is hard to believe how often we run into cases where backups have not been successful, circular logging is on and customers do not understand its implications, no one has ever done a restoral, there is insufficient hardware to restore offline, and so on. These situations are real, terribly time-consuming, and embarrassing to recover from.

Quality

The purpose of all written standards and conventions is to memorialize the team’s knowledge and produce a quality environment for everyone. Our experience is that when companies commit to a responsible standards process, the business utility delivered has the tonality of quality. The personnel involved in support are acknowledged as professionals committed to quality. They demonstrate confidence and clarity of commitment. The result is that total cost of ownership is lower than in organizations where chaos reigns and business utility is higher. An offshoot is environmental health for the people involved, even though the mission they are involved in is high profile, critical to the business, complex, and at times stressful for all constituencies involved.
Troubleshooting in a Standards-based Environment

Standards-based environments possess many positive attributes. For example, troubleshooting becomes a faster scientific process, because it is much easier to assemble facts. Simple techniques, such as looking for differences between the systems built to the standards and the ones that are not working, become quite effective. Iterating the standard becomes an exercise in controlled change, evaluation, and go forward or back in decision making. In a standards-based environment we use variations of the following process for trouble-shooting. In Chapter 29 we talk more about ideas and techniques.

✦ Collect detailed information about problems. (How often have you been told that your Exchange server is down, when the problem is really the user’s client or the bandwidth between user and server?)

✦ Assess the scope of the problem (one user, all users in one building, a particular routing group, everyone on the planet?).

✦ Use scope assessment and standards documents to deduce components to investigate.

✦ Compare differences between potentially failing components and standards-based components that are functioning properly. (If none are functioning properly, you have a standards problem.)

✦ Either fix the standard or make deviant components the same as the standard.

✦ Test.

✦ Document.

✦ Communicate.

Summary

In this chapter, we provided our perspective on how to ease the administrative burden. Variations of this model are used in every successful company we work with. (Remember, supporting everything that any business unit decides to buy is a prescription for disaster.) We have emphasized the need for process and suggested processes that have a history of success. We have stressed the need for documentation and advised on the types we rely on. In general, we have shared our commitment and belief in standards-based organizations.
Daily, Weekly, Monthly and Yearly Administrative Tasks

In this chapter, we delineate various administrative tasks that should be scheduled in order to keep your Exchange 2000 Server environment running smoothly with minimum down time or degradation of service levels. The actual list that is appropriate for your organization will depend on your organization’s needs. Nevertheless, you should be able to get a good feel for many of the items you need to consider when you build your own checklist.

We categorize tasks into three different groups: real-time, on-demand, and scheduled. Scheduled tasks can be performed on a daily, weekly, monthly, or annual basis—or any mix of the aforementioned. The combination of Exchange 2000 Server and Windows 2000 Server provides the opportunity for many previously scheduled tasks to easily be automated with real-time monitoring and close to real-time notification. Tasks that are automated in this way can be set to generate alerts in real-time (for example, with a pager), or near real-time (e-mail). When an alert is received, you can take appropriate action. In some cases, an early alert allows you to be proactive rather than reactive in performance of your administrative duties. In many cases, the alerts will allow you to address the issues before end-users are affected.
Focusing on Real-Time Tasks

Many administrative or maintenance tasks are ideally monitored or performed in real-time with a close to real-time notification via e-mail of a problem. This is particularly useful if the Administrator is carrying a wireless e-mail device and is able to receive alerts and status messages whenever in range and almost continuously. With real-time monitoring of various Exchange 2000 and Windows 2000 parameters, it is possible to move your team to a proactive posture rather than a reactive one. There are several monitors built into Windows 2000 Server and Exchange 2000 Server that are useful in this way. Additionally, there are many third-party software products, which are MMC snap-ins that provide additional monitoring features. In this section, we will discuss the real-time monitors that are native to Exchange 2000 and Windows 2000 Server.

Exchange Monitoring and Status Tool – What to monitor

In Exchange 5.5, the Administrator utilized the Link and Server Monitors to monitor the health of the Exchange site. Reviewing the information contained in the Monitor was often a daily administrative function. In Exchange 2000 Server, the Exchange Monitoring and Status Tool is much more sophisticated.

Cross-Reference
See Chapter 9 for more detail on Link and Server Monitors.

It is configured to monitor the following services by default:

- Microsoft Exchange Information Store Service
- Microsoft Exchange MTA stacks
- Microsoft Exchange Routing Engine
- Microsoft Exchange System Attendant
- Simple Mail Transport Protocol (SMTP)
- World Wide Web Publishing Service

Period of Volatility

We define a “period of volatility” as the 90-day period after any significant change, the time interval when you will often run into issues and problems. By giving this period a name and altering behavior during it, you can be better prepared to staff for and deal with likely problems. During this period, we usually increase monitoring, diagnostic logging, and communication to team members, business unit leaders, and end-users. We also try to minimize our time off and nightly sleep.
In addition to the default services, it is recommended that the following services be monitored (usually by adding a new monitor rather than adding onto the default monitor). In each of these services, the Administrator can control whether the alert occurs on a status change to “warning” or “critical.” The Administrator can also set trigger levels for the various parameters:

✦ Available Virtual Memory
✦ CPU Utilization
✦ Free Disk Space
✦ SMTP Queue Growth
✦ X.400 Queue Growth
✦ Network Interface Card Traffic (NIC)
✦ Available Physical Memory
✦ I/O Channel Utilization

The key issue in setting the alert level and parameters is to set a level that is sufficient so that you do receive false alarms but low enough to allow you to respond in a proactive manner before an event occurs that diminishes the functionality to the end-users.

The ideal situation is to achieve transparency to the end-user. Although problems may occur, they do not inhibit users from doing real work. In a sense, the problems are transparent to the end-user.

All of the items in the bullet list can be monitored either from the Exchange Server Monitoring Tool or the Windows 2000 Server Performance Monitor. We like to see all of them monitored, especially during any period of volatility.

Once you set up monitoring, do not forget notification. Exchange 2000 Server has a Notification object container that you should use to easily configure the method of notification, script, e-mail, and/or page when an event triggers an alert. In the Notification container, it is possible to select all servers, or just individual ones, to monitor.

Create a Public Folder that only the Administrators have access to, and send a copy of all e-mail alerts to that folder. This will create a historical record. It will also allow access to any alerts from the entire site to other Administrators.

**Task Manager**

Task Manager only provides real-time information. However, that information is one of the first places we look when we sit down at a keyboard for quick insights into the
health of the server. If memory usage and CPU usage are high, the system may be under too much stress. In Figure 26-1, they are extremely low after a new install in a pilot setting, but probably won’t stay that way as the server goes into production. Selecting the process page allows you to look at which processes are running and sort them by size (for example, which are taking up the most memory). The Applications property page provides you with similar information on applications that are running on the server.

![Windows Task Manager](image)

**Figure 26-1:** Real-time monitoring with Task Manager

### Dealing with On-Demand Tasks

Many important administrative tasks are not scheduled, but must be performed on-demand. Who performs them should be reflected by your administrative group assignments and permissions. A typical list of on-demand tasks includes:

- Addition, deletion, moves, and changes to user accounts

Note

Since Exchange 2000 Server is now a component of the Active Directory Schema, the Administrator who performs addition, deletion, moves, and changes to user accounts is generally the person who also will add the user to the Windows 2000 Server domain. This is different from an Exchange 5.5 system, where there were two separate and distinct security models. Exchange 2000 Server and Windows 2000 Server share a common model. Merging these tasks should help reduce administrative costs.
✦ Creation, deletion, and management of Public Folders

Make sure there are policies in place to deal with Public Folder usage, permissions, storage life, hierarchy, and so on. Without these policies, administration feels like a controlled 50 MPH skid.

✦ Permissions administration for users, Public Folders, and so on
✦ Create, modify, and remove e-mail addresses
✦ Create, modify, move, and remove mailboxes
✦ Run Performance Optimizer

Cross-Reference

Tip

When to Run Performance Optimizer

✦ Run to see if it has ideas on optimizing performance
✦ Run after adding any software item such as a connector
✦ Run after implementing any software patch (SP), Service Pack, Hot Fix, or upgrading of Exchange 2000 Server software
✦ Run after any hardware changes, especially the addition of any hard drives
✦ Run after any changes to users, site, or organization
✦ Run after the addition of new e-mail-based applications to the server that may affect traffic and/or storage
✦ Run after installing or modifying connectors
✦ Run after dealing with hardware repairs, upgrades, and so on
✦ Run after managing forms and development
✦ Run after managing and implementing software upgrades on client and server machines
✦ Run after maintaining virus protection and security

See Chapter 22 for more information on virus protection and security.

Scheduling Tasks

We tend to think about administrative tasks based on the frequency at which they’re performed: daily, weekly, and monthly/annually. In your organization, the frequency or the precise list of tasks may differ from the ones we suggest below, but all organizations need to generate a list and ensure no critical task is missing.
Daily tasks

We like to see daily tasks done first thing in the morning before most of the business users arrive. Therefore, if the tasks reveal problems that need attention, the administrative staff has time to act before usage gets heavy.

Check the Windows 2000 Server Event Viewer

Every morning, the Administrators should examine the Windows 2000 Server Event Viewer logs for each server for which they are responsible. These logs, called application, security, and system, record events that occurred previously on that Windows 2000 Server. Be sure to enable diagnostics logging for the Exchange 2000 Server so that it will record events to the application log. In cases where either an “Error” or a “Warning” event occur and are noted in the logs, immediate action should be taken to deal with the problem. In the case where there is an “Audit Failure,” it means that an audited security access attempt failed and may not need immediate action or may be an indication of an attempted security breech.

If the detail on an event refers to a log, as shown in Figure 26-2, be sure to look at the log. No alert should be ignored. Ever get the inclination to act like the people who turn off the alarm bells at the nuclear power plant because they are annoying? Don’t do the same thing with your Exchange system. Track down and deal with every significant alert. Learning to distinguish significant from spurious or unmemorable requires research (look them up) and experience.

Figure 26-2: Drilling down on an Event Viewer entry
It is a good idea to clear the event logs from time to time. Since event logs grow rapidly, this is especially true when you are having problems. An alternative is to set them to overwrite. If you do, ensure that they are large enough to give a clear indication of what is happening before they get overwritten.

If you delete the event logs, it is a good idea to save a copy offline for reference. They can be saved in three formats: .EVT that is able to be read only by the event viewer, .TXT that is an ASCII text format, and .CSV that is the comma-delimited text format. The .TXT and .CSV formats can be read by any ASCII file editor. The .CSV format can easily be imported to Excel.

**Check Exchange 2000 Server and Windows 2000 Server Monitors**

Assuming you have the right monitors set up on your machine and the Exchange Server Monitor and Windows 2000 Server Monitors are not fully configured to deliver real-time alerts, then the monitors must be reviewed several times a day. If they are configured for automated alerts, a daily check should suffice—except when there are known issues or problems.

**Check and change the backup tape(s)**

Depending upon the methodology you use in your backup plan, you may need to change the backup tapes. When this is done, care needs to be taken to store the removed backup tape in a secure place, either in a fireproof media safe or cabinet. In many cases, the backup is stored off site if it can be easily retrieved. Check and verify the backup software to ensure that there were no errors during the backup process. Make certain that all data that was supposed to be backed up was successfully recorded to the tape.

For more information on backup and restore, see Chapter 28.

**Weekly tasks**

Weekly tasks can be accomplished any time during the week. We suggest you do not choose Friday night, as it tends to spoil your evening and sometimes the whole weekend.

**Check the Microsoft Web site**


**Check for virus protection updates**

This process can be automated, but in most organizations it is not. Often, audits will highlight the fact that virus protection software data files, or the core software itself, is hopelessly out-of-date.
Spot check to ensure the virus protection is configured as per your standard. We find it is common for users and others to decrease protection by turning heuristics off or by altering the setting for what is scanned when.

**Verify replication**
Ensure that the right Public Folders are replicating and with the right frequency.

Some of these tasks may or may not be an Administrator’s function in your organization. Make sure that you do not violate IT policies by overstepping your authority. On the other hand, find out who has each of these responsibilities and ensure someone is doing the job.

**Monthly and annual tasks**
Some organizations do a great job on daily and even weekly tasks, but ignore the monthly or annual tasks because they never get scheduled. We believe they are just as important to a healthy messaging system.

**Verify the last backup**
This task requires sufficient scratch disk space to restore the entire object being verified. It can be large (such as a large Information Store). If there are multiple Information Stores, they can be restored individually. If so, scratch disk space must be equal in size to the largest. This restore will not normally be saved, but the hard drive space is required for the verification process.

**Storage and load allocation**
Examine the distribution of mailboxes and Public Folders on the server that you administer and determine whether or not to move them in order to balance processing load or storage capacity.

**Clean up mailboxes**
Check the size of individual mailboxes. This task should be performed monthly. Regularity will get end-users used to the policy and give you a better chance of enforcing it with minimum disruption. Infrequent or random enforcement of standards will generally cause significant turmoil with end-users. In some cases, companies have hard limits for individual storage of e-mail, and exceeding those limits is not discretionary.

If your company has no hard limits, you might try publishing a list of who uses what storage and use peer pressure to shame them into altering their pack rat ways.
Make certain that company policies concerning the amount of e-mail that can be stored online is detailed and accurate. Make certain that it is clearly explained to the users, and then work with them to achieve its objectives.

**Test UPS**
Test any Uninterruptible Power Supplies (UPS) and/or backup generators.

---

**Caution**
A bad UPS may stop a production server if you test it while running production. Test during off-peak times and use a careful method so you do not damage your servers. An alternative is to cycle a UPS out and test when not on the server at all.

---

**Clean the tape drive**
The tape drive should have a cleaning kit supplied that is specific to that hardware. Cleaning the drive will remove magnetic particles that can wear or scratch a tape. This task should not be forgotten.

---

**Tip**
When cleaning the tape drive, make sure you wait at least 15 minutes after cleaning before reinserting a data tape into the drive. This will give the cleaning solvent time to evaporate. The solvents that clean the tape drive heads will readily dissolve the binder for the magnetic coating on the tape, thus ruining it.

---

**Replace hardware**
Tapes wear out and should be replaced annually in order to ensure that the information recorded on them is restorable.

Run hard disk system utilities to defragment and scan your drives. Replace hard drives, as their incidence of errors increases before catastrophic failure.

**Update software**
It is important that you keep updating your software versions, especially virus protection.

**Run through a disaster simulation**
Run through a disaster and recovery simulation on a regular basis. Actually restore a server or an Information Store or a storage group to ensure that you have all the components you need and that you and your peers have the necessary experience.

**Review Public Folder hierarchy**
Is the hierarchy working as planned? Should some of the folders be renamed or moved? Can age limits or size limits be altered?
Review NNTP newsfeeds
Are the NNTP newsfeeds being used? Are they the right feeds for your organization?

Record keeping
As any of the tasks (real-time, on-demand, or scheduled) are performed, a record of who performed the task and the date and time completed should be kept. This can provide useful diagnostic information that may be helpful in future problem-solving.

Summary
With the advent of Exchange 2000 Server and Windows 2000 Server, many of the tasks performed by Administrators have been combined (as in adding new users) or automated (as in the new alert and notification capabilities). These tools allow the Administrators to perform a much more proactive role in maintaining the Exchange systems in their care. However, there are still many manual tasks that need to be performed, either as on-demand or scheduled tasks. Each environment and organization is different and may have special needs or structure. Modify the contents of this chapter and produce your own custom Administrator’s handbook in order to best meet those needs.

✦    ✦    ✦
Exchange Administrators have a number of utilities available to them for addressing common problems and assessing system status. Some of those utilities come with Exchange 2000 and some come with Windows 2000, but even the ones that come from Windows 2000 are used by Exchange services and Administrators. This chapter covers the major utilities available to Exchange Administrators and clarifies when and how to use them.

**Repairing Inboxes with ScanPST**

Some users store all their messages on the corporate messaging servers. Others have Offline Storage (.OST) files to keep their local storage synchronized with the message store on the corporate servers, or Personal Folder File (.PST) to store messages outside the Exchange Information Store or Web Store. A .PST is usually placed on a local drive, although it could also be placed on a network drive.

Administrators can easily be confused about the use of the inbox repair tool. ScanPST does not fix problems on the Exchange Server Information Store or Web Store; that is the job of utilities such as ISInteg and ESEUtil, described later in this chapter. Instead it repairs a damaged PST or OST file.

ScanPST.exe is loaded when you load Outlook on the client. In a representative Outlook 2000 installation, it was found in Program Files\Common Files\System\Mapi\1033\95.
Using ScanPST involves a few caveats:

- Depending on the severity and nature of the problem, ScanPST may not be able to fix it, or could cause catastrophic failure and preempt any subsequent effort to save data. Therefore, back up even the damaged PST before attempting repair. If the utility causes catastrophic problems, the backup at least allows you to restore the less catastrophic problem situation.

- As with many database utilities, you must have enough disk space to make a complete copy of the database being repaired, and additional space for temp files.

- To speed up the process, use machines with fast I/O, disk drives, and plenty of memory.

Caution

It is vital to back up before you attempt to run any utilities that work on your data, including ScanPST.

To run ScanPST, you should

1. Make a backup copy.
2. Ensure you have plenty of disk space available.
3. Run ScanPST.exe from the command line.
4. Select the PST you wish to repair.
5. Click OK.

6. Wait until the utility completes (that is, do not interrupt it while it is running).

7. If errors are found, you will be asked if you wish to repair them, as shown in Figure 27-1. If you do not select to do the repairs, the PST will remain damaged. On the page that reports errors, you can also specify where you want the repaired file to be written. Select a drive with high-speed disk access and sufficient space wherever possible.

Turning off the machine, or otherwise interrupting drastically, could cause irreparable damage to the database. To ensure no one interrupts the machine while it is running, turn off any screen saver and any sleep mode settings and put a sign on the machine.

![Figure 27-1: Running ScanPST.exe](image)

### Making Database Repairs with ESEUtil

ESEUtil is a server-side utility used to repair, defragment, and compact low-level damage to Exchange private and public folder information store databases. *Defragmentation* is the process of moving data within the database to ensure that records are contiguous. *Compaction* compresses the database file and removes the blank space.

*Note* 
Defragmentation addresses the order and organization of the data; compression addresses the size of the database file that holds the data.

ESEUtil repairs, and provides information about, violations of the basic database structure. It comes with Exchange 2000 Server and is also provided as part of Windows 2000 Server (for repairs to the directory).
ESEUtil is only one utility provided by Windows 2000 to use and repair the directory. It is a subset of NTDSUtil. Anyone using these utilities should be fluent with Windows 2000 and own responsibility for the Active Directory. See *Windows 2000 Server Bible*, published by Hungry Minds, Inc., for more information on the directory utilities.

ESEUtil is found in \program files\exchsrvr\bin.

In the normal course of operations, the Exchange Information Store Service runs regularly scheduled maintenance.

For information on setting the schedule, see Chapter 8.

The scheduled maintenance helps defragment the database. You should need to defragment manually only when you have made a very large number of database changes all at once and want to clean up the database quickly—for example, if you have deleted user mailboxes for a whole division. The database runs faster and with less overhead when it is defragmented.

The temporary database should be local and on a high-speed disk. Make sure you have free space equal to 10 to 25 percent more than the size of the database to make room for the temp file. If the system ran out of space in the middle of running this utility, your effort would likely fail and might permanently damage that copy of your database.

Even with regularly scheduled defragmentation, the database is not compressed; the bits are put in order, but blank space remains. To defragment and compress the database, run ESEUtil offline on the database in question.

ESEUtil is run from the command line after you stop the Information Store Service to close the file.

Either work on the original offline file or make a copy of it to work on. If you work on a copy, do not have new transactions written to the original, unless you are just testing for likely success and the amount of time a procedure would take.

ESEUtil allows a number of switches:

- **/d**: Use this switch for defragmentation. ESEUtil moves pages with data into a temporary database (tempdfrg.edb) and discards unused pages.

When defragmentation is complete, the system writes a new database signature to your database. Thus, before you start, do a full backup to clear all transaction logs. Immediately after the backup, perform a fresh, full online backup; none of your old transaction logs can be written to that database, because the signature is different.
✦ /r: This switch tries to recover broken links in the database by examining the structure and acting on the inferences that result. If the utility finds -1018, -1019, or -1022 errors, it removes the page. Since it removes data (albeit corrupt data), this option is only used as a last resort. The outcome is somewhat unpredictable, and loss of data is a real possibility. When the option is completed, you are instructed to delete all the transaction logs, as they are no longer useful—the page numbers have changed, and a new signature to the database has been written.

✦ /g: This switch is used mainly for feedback to the Microsoft development team. It highlights database integrity, is read-only, and does not alter the database in any way. However, the option needs space to run. It looks at the database index and builds a second index called integ.edb, and then compares them.

✦ /m: This switch provides a file. A useful option is /mh with /v to dump the message headers in verbose mode. If the message headers are being corrupted or are in an inconsistent state, it will tell you. Microsoft recommends that you run the /mh switch as a first check of information store integrity. Since the /mh switch checks headers only, it runs faster than some of the other modes.

✦ /p: This switch attempts to repair the database.

/p does not help with log file playback, and precludes you from using existing log files, causing potential loss of data. Like the /r option, use it only after you have exhausted other options. Many would use /p before /r as the last resort. We suggest you contact Microsoft support for recommendations if you are in serious enough trouble to need /p or /r.

✦ /c: This switch restores the database.

These switches are shown in Figure 27-2.

![Figure 27-2: Options for running ESEUtil.exe](image-url)
Because of the high probability of failure or loss of data, use ESEUtil to attempt a fix or recovery of your data only after all other options, such as restoration from tape, have failed.

**Using ISInteg to Regain Data Consistency**

ISInteg, another server-side utility, is commonly used after running ESEUtil. While ESEUtil ensures the data structure integrity, ISInteg ensures consistency of the data entries from an application perspective.

If your Information Store will not start, run both ESEUtil and ISInteg after ensuring you have a good backup.

ISInteg is run from the command line. After you install Exchange 2000, it is found in the exchsrvr\bin subdirectory. ISInteg has two modes: *test* and *patch*. Test mode provides a glimpse of what is ailing your information store database. ISInteg looks for table inconsistencies, such as out-of-synch or non-referenced objects and incorrect counts. Patch mode is used when you have restored a non-Exchange generated backup (that is, one done by a Windows 2000 Server offline without Exchange). The utility attempts to repair the private or public database and synchronizes the GUIDs. Available switches are shown in Figure 27-3.

- `-s <servername>`: Specifies the server name
- `-fix`: Checks and fixes the databases (default is check only)
- `-verbose`: Causes more thorough logging
- `-l <filename>`: Specifies the log filename (default is \isinteg.pri /pub)
- `-t refdblocklocation`: Default is the location of the store
- `-test <testname>`: Specifies certain tests from the number of tests available:
  - **Folder message**: aclitem
  - **Mailbox (pri only)**: delfld, acclist, rcvfld (pri only), timedev, rowcounts, attach, morefld, ooflist (pri only), global searchq, dlvrto per user artidx (pub only), search newsfeed (pub only), dumpsterprops
  - **Ref Count tests**: mssgref, mssgssoftref, attachref, aclistref, aclitemre, newsfeedref (pub only), fldrcv (pri only), fidsub, dumpsterref
  - **Group tests**: allfoldertests, allacltests (allfoldertests is a common one to run)
  - **Special tests**: deleteextracolumns
- `-dump [/l logfilename]`: For a verbose dump of the store data
To run ISInteg

1. Ensure you have a good full backup of the database.

When working on databases with the Exchange utilities, use a copy that was made on hard disk rather than tape; it is much faster to create and restore a copy on local hard drives than from tape.

2. Go to exchsrvr\bin.

3. Type ISInteg from the command line and use whatever options you need.

4. After altering the database in any way, ensure you get a good online full backup immediately.

Figure 27-4 shows an actual repair of a database on an earlier version of Exchange.
Testing RPC Communication with RPCPing

Remote Procedure Calls (RPCs) remain an important communication method in Exchange 2000. They are an alternative to static mappings for dynamic-link libraries. RPCs enable an application to run partly on the server and partly on the client. In Exchange 2000, RPCs are used for communication within a routing group, some client-to-server connectivity (that is, Outlook 2000 in some configurations), and the Exchange Microsoft Management Console snap-in.

When you use RPCs, first ensure that you have enough LAN bandwidth to avoid RPC timeouts. Exchange includes an RPCPing utility to test RPC connectivity across physical, network, and transport layers of your network.

RPCPing consists of two parts: the server and the client. These utilities are found in the \Support\RPCping subdirectory on the Exchange CD-ROM.

RPings

RPings is the server-side utility, which must already be running before you start the client-side utility. Type RPings.exe at the command prompt to begin the program. The program can respond to a ping for all available working transport protocols (TCP/IP, IPX, and so on). If you wish to focus on only one protocol and restrict the RPC ping to that protocol, use one of the available switches. The switches are invoked by typing -p and the protocol name:
-p ipx/spx
- p namedpipes
- p netbios
- p tcpip
- p vines

When you are done and you want to exit RPCPing server, type @q at the command prompt.

**RPingc**

RPingc is the client half of RPCPing.

In this context, client and server are relative terms. In the lexicon of client-server communication, a client can be a workstation—or even a second Exchange server, if you want to determine whether two servers in a routing group could maintain RPC communication.

There are four versions of the RPCPing utility found in the \support\RPCPing subdirectory, each in its own child directory. Select the one appropriate for the operating system on the client you are testing. Execute Rpingc.exe from the command prompt after you have started the server component. There are several options:

- **Protocol Sequence:** Select the protocol sequence: Any (use this for all protocols), named Pipes, IPX/SPX, TCP/IP, NetBIOS, or Vines.
- **Exchange Server:** Specifies the server to connect to by specifying either a NetBIOS name or a TCP address.
- **Endpoint:** Dictates protocol-specific ports that RPC will use to communicate from the client to the server—for example, you can choose the store to test communication between the client and the port on the server used by the Information Store Service.
- **Number of Pings:** Controls whether RPCPing runs continuously or only for a certain number of pings.
- **Mode:** Controls whether the ping is returned directly from the RPC Server or from an endpoint.
- **Run with Security:** Verifies authenticated RPCs.

RPingc establishes connectivity using the parameters you specify if possible. If no connectivity is possible the way you specified it, ensure that the server was up and running before you started the client, and try different protocols and endpoints. If
you find that some protocols work and others do not, move the ones that work to
the top of the binding order. If no protocols are working, ensure you have basic con-
nectivity (that is, that you can open a file on the server, or at least ping the server).
If you do have basic connectivity, look for a corrupt DLL; a likely candidate is
RPC.dll, which may need to be replaced. If you replace the DLL and seem to have
basic connectivity, begin network transport and protocol analysis.

Review and Repair Queues Using MTACheck

The role of the Message Transfer Agent (MTA) has changed in Exchange 2000. In
earlier versions of Exchange, MTA was responsible for almost all message move-
ment; therefore, its queues were of high interest. In Exchange 2000, the MTA serves
the X.400 connector only. Thus, its queues have only those messages bound for, or
coming from, the X.400 Connector.

MTACheck allows you to check and repair those queues. The MTA stores the mes-
sages on the Exchange Server with files that have a .dat extension in a subdirec-
tory called \Exchsrvr\Mtadata. Those files can be corrupted, often by improper
shutdown of the MTA (power failure, pulling the plug, or a bolt of lightening on
your server). Corrupt files can cause message-movement problems and may inhibit
the MTA from starting or stopping.

If, when it starts up, the MTA senses that it was not shut down properly, it runs its
own MTACheck. In this case, it logs the results to the Windows 2000 Application
Event Log and creates an MTACheck.log file, which it places in the
Exchsrvr\MTAdata\MTAcheck.out directory.

Tip

If you are experiencing problems with your MTA, stop and start the service and
look at the Application Event Log.

When the files are corrupt, they are often repairable. If stopping and starting the
MTA service does not resolve the problem, you can use MTACheck, a command-line
utility, to attempt a repair with more explicit control. MTACheck (found in the
Exchsrvr\Bin subdirectory on the Exchange server) attempts to make repairs to all
the MTA queues and deal with all damaged messages in those queues by exiling
them to the Exchsrvr\MTAdata\MTAcheck.out directory. Like the automatic ver-

To run MTACheck, first stop the MTA service by opening the control panel services
applet, finding the MTA Service, and clicking the stop button, as shown in Figure 27-6.

If you forget to stop the MTA service, you will be reminded when you run the utility.
If the queues are large, the check can take a while. When it is complete, you will
have some response at the command line, as well as the Event Viewer (shown in
Figure 27-7) and log file entries mentioned earlier.
The following command switches are allowed in MTACheck:

✦ /f <filename>: Enables you to specify the name of the file to log status in.
✦ /v: Causes verbose logging to be turned on; requires the /f switch.
✦ /rd: Removes all queued directory replication messages.
Removing queued directory replication messages helps make the queue smaller. The directory is self-healing. The messages are retransmitted later, so the effect of removing them is to allow MTACheck to run faster, lower the likelihood that you have corrupt messages, and introduce some latency to your directory replication.

- /rp: Removes all queued public folder messages (like directory replication messages; these messages are retransmitted when the public folders synchronize and realize they are missing)
- /rl: Removes all queued link monitor messages

Alert other Administrators who are responsible for systems on both sides of the Link Monitors if you decide to remove these queues. Otherwise, they may come to spurious conclusions about the meaning of the monitor when the alarms start to trigger because an expected return message has not arrived as a direct result of your deletions.

**Examining the Event Log**

The Event Viewer is used to view the Windows 2000 Event Log. Many subsystems and applications write to the Event Log, including Exchange, many of its components, and some of its utilities. In this context, an event is any occurrence that requires information to be logged or a notification to be generated. Being very familiar with the Event Viewer and log is critical to administrating an Exchange environment.

System, Application, and Security logs can be viewed through the Event Viewer. Exchange and its subsystems can cause an entry to any of the three, but the majority of entries are made to the application log. The three types of events logged are Errors, Warnings, and Information. An example of an event viewer log is provided in Figure 27-8.

Diagnostic logging is controlled on property pages for subsystems or services that support it—for example, the MTA and the X.400 Connector. Diagnostic levels can be set to None (default; preferred), Minimum, Medium, and Maximum.

Though diagnostic logging can be set to provide a lot of information about various services or subsystems in the Event Logs, it also takes bandwidth and overhead for your server. If it is set too high—or set and forgotten—when no further utility is derived, it could seriously affect your messaging system adversely by bogging down your server for no appreciable purpose.

Log files can be viewed directly through the Event Viewer, or exported to a text file and imported into another application for sorting or other manipulations. The files themselves are located in the \system32\config subdirectory and have an .evt extension.
You will often find yourself using the Event Viewer to look at the local server and other servers across the network. The ability to look at other servers across the network (when you have sufficient permissions) is another way to administer large, geographically dispersed Exchange messaging environments centrally.

**Exploring Other Utilities**

This section examines other minor utilities commonly used when troubleshooting Exchange; some come with Windows 2000.

For more information on those utilities, see the *Windows 2000 Server Bible* and other books on Windows 2000 published by Hungry Minds, Inc.

**Error**

This utility, found in the `\Support\Utils\i386` subdirectory, is used to convert error messages from the store, MAPI, and database message strings intelligible to human beings.
**FileVer**

This utility, found in the `\Support\Utils\i386` subdirectory, is used to identify the version numbers of DLLs and `.exe` programs.

**Mdbvue32**

This utility, found in the `\Support\Utils\i386` subdirectory, provides information on PST and OST files. It can also be used to log into system mailboxes, such as the System Attendant (SA); sometimes messages get stuck in the SA mailbox and need to be manually removed using this utility.

**Task Manager**

To evoke the Task Manager, right-click on the lower right of the Windows task bar. Task Manager enables you to see which services and applications are running and stop them. On the performance monitor screen, you can see how much load is on your processors and memory, as shown in Figure 27-9.

![Performance monitor](image)

**Figure 27-9:** Performance monitor

**Replmon and RepAdmin**

The replication monitor and replication administration tools are available on the Windows 2000 resource kit. These tools allow you to view, monitor, and troubleshoot the state of replication connections throughout the Windows 2000 Active Directory. The replication monitor tool shows a graphical representation of replication links and enables you to test individual connections and run the Knowledge
Consistency Checker. The replication admin tool has similar functionality as the replication monitor and also enables you to look at replication metadata and replication partner relationship consistency.

**Schema Admin**

The schema admin tool allows you to read, export, and change the schema from an MMC snap-in, rather than having to create LDAP scripts. This snap-in is available to add to a custom MMC console when you install the admin.msi administrative tools package that is located on the Windows 2000 installation media.

**Windows Management Instrumentation**

Windows Management Instrumentation (WMI) is the Microsoft implementation of Web-Based Enterprise Management (WBEM), which provides uniform access to management information. Exchange 2000 supports WMI by including WMI providers that can be used to access status and other information about the Exchange system.

**Exchange 2000 Resource Kit**

One of the most useful toolkits to have for administering Exchange is the *Microsoft Exchange 2000 Resource Kit* (Microsoft Press). This kit includes an online book with content-search capability, containing all the information in the printed version of the Resource Kit. The Resource Kit includes many utilities, tools, and programs for distribution list management, public folders, security, simulation, administering, planning, and development on Exchange. Items of interest in the Resource Kit include the following and more:

- **Exchange 2000 Server Deployment Planning templates**
- **Exchange 2000 Server Tools and Help**
- **Archive Sink**: Archives all messages to the temp folder \WINNT\ (Archive Sink is a DLL activated after it is registered in the Windows 2000 registry)
  
  Using Archive Sink can consume a significant amount of disk space; housekeeping should be done regularly, including purging messages (by deleting them or retaining them offline).

- **AWP Tool**: Clears all passwords for servers specified in connection agreements, after which each connection can be reestablished by resetting its user name and password

- **Compare Tool**

- **DSCFlush Tool**: Clears the DSCAccess cache for the computer on which the command is executed (local machine)

- **Duplicate Proxy Tool**: Enables you to search the directory entries based upon parameters input in the command (using wild cards allows flexible searches)
DXAFIX Tool: Repairs the Microsoft Exchange Directory Synchronization Service database xdir.edb, whose default location is \Exchsrvr\Dxadata

Exchange Mailbox Merge Tool: Merges tool mailboxes

LoadSim: Simulates certain types of loads on an Exchange server

For more information about LoadSim, see Chapter 19.

LPADC Tool: Enables you to add, delete, or test Active Directory Connector passwords (must be run on the server on which the ADC you wish to test is located)

MailBox Store Finder Tool: Creates a list of Fully Qualified Domain Names for each mailbox store located in the Active Directory

MailBox Statistics Tool: Generates statistics (mailbox by mailbox) for better insight into the use, volume, and patterns of e-mail usage

MailQ Tool: Enables you to view the mail queues for SMTP, X.400, and the store for a selected server

MAPISend Tool: Enables you to send MAPI messages or perform MAPI commands from a command line (can be useful in scripting procedures such as startup or maintenance)

MTAView Tool: Enables you to view the structure of an Exchange 2000 Server MTA database

NET2STRM Tool: Directs the output from the network monitor to a .txt file or the screen

Profile Generation Tool: Creates Outlook 2000 profiles

Topology Diagrammer Tool: Diagrams the Exchange 2000 topology

Web Monitor Tool: Monitors queues, disks, memory, CPUs, and services links via the Web

Winroute Tool: Allows you to examine the routing, link state, and connector state of a specified server

Summary

Microsoft provides powerful utilities in both Exchange and Windows 2000 to help with your administrative tasks. This chapter examines each utility and shows you how to use them. To succeed with Exchange, you must rely on these utilities regularly. Gaining familiarity with many of these utilities before you need them is crucial to your success as an Administrator. And learning to use them in your daily work with Exchange can make your job a lot easier, providing the tools you need to examine, assess, and repair many of the problems you encounter.
Areas of Special Attention for Support

This chapter examines some areas deserving special attention. Performance/scalability and backup/restore are two of the more important and challenging topics in a book about administering Exchange — important because protecting the corporate record and expanding messaging systems are common to almost all installations, and challenging because recommendations depend on the actual topology and implementation of your organization. To address the challenge, this chapter provides universal recommendations and then evokes common scenarios and explains how to address them.

Performance and Scalability

Exchange 2000 makes real strides in performance and scalability that you should take advantage of anywhere performance is critical, especially in large organizations. In any size organization, you can use the utilities and monitors to discover where your performance bottlenecks are.

For information about utilities and monitors, see Chapter 27.

Use the planning techniques discussed in Part IV to predict your needs to scale your system. This section discusses which features apply to different-size organizations and scenarios and how you might take advantage of them.
The smallest organization

If you are in an organization with one or two servers or a small-business server, focus on getting the most performance out of your hardware and mapping your software to your hardware. Specifically

✦ Focus on having multiple spindles and separating data from log files.
✦ Use all the channels on your motherboard and your drive controller cards.
✦ Buy as much memory as you can afford.
✦ Ensure that your clients also have fast enough machines.
✦ Ensure that the LAN or WAN links are solid and fast enough to deliver the expected response times.

A fast server box connected to a slow client machine over a slow-speed LAN or Internet connection will be perceived by the user as slow. Rather than speed the server up, focus on the slower components to gain end-user performance.

In the smallest organizations, scalability means adding more drives, memory, or additional servers. If you add additional servers, make sure you deal properly with licensing.

The intermediate-size organization

Combinations and permutations become more interesting in a medium-sized organization with multiple servers and potential for multiple routing groups, administrative groups, connectors, and almost any combination of services. Even if you have few employees, you may have many vendors or customers, and so you may even need multiple forests and Exchange organizations; you may also need high availability and throughput for access from the Internet. You will probably also use more of the development features.

To begin scaling and performance in a medium-sized organization, determine whether your current situation is balanced (that is, free of bottlenecks) and whether it provides for the committed service levels. If not, focus on the most serious bottlenecks first. Think about other problems as you propose and implement solutions. You should review the following:

✦ **Hardware**: Is there enough memory, CPU MIPS, hard-drive space and speed, I/O, backup devices?

✦ **Hardware/software mapping**: Are the right services allocated to the right servers? Are the right users on the right machines? Can you reduce the effect of slow-speed WAN links by moving information closer to users (by public folder replication, placement of physical servers, and so on)?

✦ **WAN bandwidth**: Are you using all the bandwidth? Is there too much or too little on any link (after accounting for future growth)? If you need more
throughput or want to control the usage of asynchronous links, close the windows on the connectors — causing mail to aggregate, passing less frequently but in greater quantity. You can also restrict certain types of mail, such as public folder replication of certain folders, to off-hours. Regarding scalability, when ordering WAN links, always ask what the cost is to go to the next level of throughput. Buying a fractional T1 is usually a better deal than a 56KB dedicated circuit. The path for more throughput is easy on the former and quite expensive on the latter.

✦ **Connectors:** Do you have too many? Too few? Does your design have redundant links (routes to get the messages through even if parts of your WAN or certain bridgehead servers are down)? Are your cost values set right? To scale your system, assess whether you would be better off with more connectors, or with faster links to the existing connectors. Assess whether the server with the connector can keep up with the traffic, or whether moving the connector to another server (or a new server) would improve performance.

✦ **Monitoring and diagnostic settings:** Are too many settings turned on, slowing down your system? Are too few on, or are they set too low, to help you troubleshoot?

✦ **Topology:** Should you go to a front-end/back-end server model for Internet scalability and performance? Do you need additional high-speed network cards for a backup LAN? Are your servers protected from Internet hackers?

### Large organizations

All of the considerations of small and medium-sized organizations also apply to large organizations. The considerations addressed here are more likely to apply to large organizations but may also apply to some medium-sized organizations:

✦ **Directory access, topology, and usage:** How many Global Catalog servers do you have? Where are they? Which LDAP calls do you allow? How much information is returned in each LDAP call? How many names are in your offline address book? How do you manage changes to your ADS schema, which all cause a complete re-replication of the entire directory?

✦ **Language:** Many large organizations are multilingual. The language used to administer the server, the languages used in e-mails by the users, and the language characters supported in connectors can all become issues.

✦ **Model for information store backup and recovery:** Do you have the right number of storage groups and tape devices? Do you have the right users assigned to the right storage group to ensure your system is not making too many instances of a message?

✦ **Public folder models:** In very large organizations, use of public folders can move rapidly toward chaos. Revisit your policies and adapt them as you scale. Ensure that you balance the cost (in time or money) to replicate against the utility of having the information closer to users.
✦ **Retention models:** Most large organizations give serious thought (with the help of their legal team) to retaining messages. But you can often scale larger, get better performance, and recover faster with existing hardware if you store less.

✦ **Cluster components:** Many large companies use clusters to maintain high availability. Make sure at least monthly that all the drives and components of your cluster are sound.

### Managing memory

Organizations of any size can benefit from paying attention to memory management. The Extensible Storage Engine (ESE) manages the transaction logging system used when writing to the information store. One of the criteria for ESE was that it be somewhat self-tuning. Other criteria included recoverability from a crash, and reducing the number of I/O operations. All of these require deft memory management.

When ESE makes a change such as deleting an e-mail message, what it really does is write the change to a memory page that corresponds to a page in the Information Store. Then, ESE applies the transaction to the database and writes it to a transaction log. The transaction often includes a number of operations; all of the operations either complete and clear the transaction or roll back if any of the associated operations did not complete.

In order to be able to write pages from the database into memory, ESE must reserve memory space. As you can imagine, the reservation system should be dynamic. ESE uses Dynamic Buffer Allocation (DBA), which increases the size of the buffer cache for ESE dynamically.

In Exchange 4.x, the buffer cache was set by performance optimizer and not set dynamically, so that if you added memory and forgot to rerun performance optimizer, Exchange might derive little benefit from the added memory. That is not the case with Exchange 5.5 or Exchange 2000; full benefit from the memory is immediate, with no additional effort on your part.

The more memory you add, the more Exchange has available for caching transactions, and the faster it will usually run. The only exception to the dynamic allocation is when you specifically assign memory to exclusive use by an application such as SQL server; but in practice, not many production Exchange servers run on the same box as SQL server. Buy the biggest Dual In-Line Memory Modules (DIMMs) you can to leave room for more, and buy as much as you can afford; then add as you see usage increase. ESE uses all available memory (physically in the machine, not dedicated to another application and unused by other processes). It frees memory up dynamically as other processes call for it. When they release it, it will reclaim it for its own use again. You should not be alarmed when you go to Task Manager and see Store.exe using most of your memory. When everything is working right on an Exchange 2000 server and the Exchange server is working hard, that physical memory number is often low. In the example shown in Figure 28-1, available physical memory is 316,600; as the server becomes busy, that number will dynamically change.
Slow-Speed Links

Some of us remember horribly slow speed links, such as 100- and 300-baud. You won’t see links that slow today, but you will be trying to push more bits through your slow links than you ever did in the old days. The bottom line is that slow-speed links still need a lot of attention, and a bit of scientific artistry, to gain maximum performance.

In your Exchange environment, you often need to tie in some users, or two servers, across slow-speed links. If the issue involves end-users, focus on their patterns of use. You may be able to improve performance for them by downloading all their messages to their local machine, even though that may not be technically faster in any measurable way. If the issue involves server-to-server or user-to-server slow-speed links, try the following solutions:

✦ **Configuring connectors:** Look at the use of wildcards in your connectors, to be sure that you are not pumping too much traffic through them. If the link has usage charges like many asynchronous telco lines, be picky about what goes through them. Ensure that the sliding windows are set for the proper type of line you have (good lines require fewer checkpoints than bad lines). Make sure your connector costs are set right so that you get some protection when other circuits are down, but do not push any excess messaging through the slow-speed link. Many companies set connectors for slow-speed links to reject messages over a certain size.
✦ Using Address Space property pages: You can route messages to, or away from, slow-speed connectors. Pay special attention to the scope of the connector; scoping connectors can be an extremely effective way of controlling message flow.

✦ Adding I/O boards to your server: Add boards for multiple RAS lines with better throughput and less impact on server CPU utilization.

✦ Implementing OST and PST usage for client machines: Both OST and PSTs allow users to work locally, with less frequent transmission across slow-speed links. This option requires no attention to download; users can be doing something else. For them, the perceived improvement in performance can be great, even though it may not be an improvement in the speed of transmission or by any other technical measurement.

✦ Using OWA and clients that leave mail on the server: OWA and IMAP4 clients can leave e-mail on the server. Instead of downloading the whole mailbox, all that is being transmitted are headers and then only the messages of interest.

## Mixed Exchange Environments

There are three profiles for what we call a mixed Exchange environment, all challenging to administer in different ways. The first profile is an environment with differing versions of Exchange itself.

For a more detailed explanation of this profile, see Chapter 18.

In applying the first profile, be sure to do to the following:

✦ Get all of your bridgehead Exchange servers to version 5.5 SP3 first
✦ Deal with your underlying NT domain structure
✦ Review and rework the naming conventions from your previous configuration

Get not only your bridgehead servers, but all of your Exchange servers, to version 5.5 SP3 before you coexist or migrate to Exchange 2000—unless doing so will slow down your project.

The second profile consists of a legacy environment (other than Exchange) that you wish to connect to and coexist with. In this case, try to make Exchange your backbone as quickly as possible.

With all the other features in Exchange, few people realize that Microsoft has built a very sophisticated messaging switch.
Once Exchange is your backbone, you can take advantage of many of the administrative features for connectors, including monitors, notifications, and advanced messaging features, such as automatic rerouting. Doing so allows you to make the backbone rock-solid and tie legacy systems to it, as shown in Figure 28-2. The only servers communicating directly to the Internet are the Exchange servers. The MS Mail servers communicate to each other by communicating through the Exchange servers that communicate through the Internet. Even in earlier versions of Exchange, this was a very effective approach.

![Diagram](https://example.com/diagram.png)

**Figure 28-2:** Exchange as a backbone with legacy MS Mail systems connected

The third profile consists of a legacy environment (other than Exchange) that you wish to migrate from as quickly as possible. Start by using the same approach of migrating to an all Exchange 2000 backbone as quickly as possible. In the process of migrating, protect additional backups and any programs that decrypt, as they often store messages as a flat file that can be read by anyone unless you keep it secure. Ensure that end-users can reply to e-mail and don’t lose access to signed and sealed messages, and that you can recover mail from backup tapes.
Roving, Wireless, and Remote Users

With wireless e-mail, you can fill the cracks in your day responding to e-mail. Not only does this increase your productivity and response time, but when you go home at night after a day on the road you don’t have hundreds of e-mails to deal with. Here is one of the many different scenarios for wireless e-mail.

Research in Motion (RIM), a Canadian company, makes a number of wireless handheld devices that connect to Outlook and Exchange. One of the devices is called a BlackBerry. It comes in two formats; one is the size of a pager, the other the size of a PalmPC or Palm Pilot. On the pager-sized unit, the keyboard works surprisingly well with your thumbs. Similarly the pager-sized unit has a small screen, but a wheel scrolling the screen solves this problem for the body of most e-mails. Although you can see the text of an e-mail on the BlackBerry and can forward a message, you cannot open attachments.

You can connect a BlackBerry to your Outlook client machine connected to the Internet while running a redirector to redirect BlackBerry messages. Alternatively, you can purchase and install a service on your Exchange server. With the Exchange service e-mail is delivered to your mailbox and a copy sent over the Internet to the RIM servers, which dispatch it over the wireless carrier to your unit. When the service is installed on your Exchange server, a BlackBerry property page is added, as shown in Figure 28-3. This property page enables administrators to manage many aspects of the end-user BlackBerry experience.

![Figure 28-3: BlackBerry Exchange Service property page](image)
If your BlackBerry originates or responds to a message, it transmits an encrypted message to a local radio carrier, which then transmits it across the Internet to your Exchange Internet connector. From there, your Exchange messaging system delivers it to your mailbox. If the Exchange servers running the RIMM service or local PC redirector are down, your BlackBerry can communicate with other BlackBerries, using the PIN for each unit.

If you wish to synchronize your contacts and other information, place your BlackBerry in its cradle and attach the cradle to your serial port. Then use the software provided by Intellesynch.

Some users combine inbox rules with wireless; when a message is sent to a user, an inbox rule triggers that says something like “If this message is really important, send it with the word Urgent in the header. Then it will be transmitted to my wireless device.” Other users get all their e-mail forwarded to their wireless.

Tip

Backups and Restores

No topic is more important to users and management than being able to recover the corporate record quickly and at will. You can recover at will only if you have prepared for it with regular and successful backups; recovering quickly, especially in large-server environments, requires the correct application of money and design to improve speed.

Lost information

As users store more and more information in Exchange, you must protect that corporate record both from catastrophic loss (for example, calls to retrieve one or many messages because a user hit the delete key too fast) and from failure on the client machine. In the latter case, you may be able to restore users’ messages only if they are using your Exchange server to store their mail. If they are using a local PST, you will be able to restore their mail only if you or they were backing up their PST and you have access to it. You may also have problems if you are trying to restore mail that was created before an upgrade of Exchange or a change in topology that affects encrypted e-mails. Certain versions of the Information Store will not restore and start on differing versions of Exchange, meaning a longer, more arduous process to recreate the lost environment in order to do the restore and access the e-mail. Think of yourself as a cross between an archeologist and a librarian. Be prepared to uncover (that is, recover) the old information, and to protect the information in your care and prepare it to be uncovered in the future.
Exchange-aware backups/restores

Unlike earlier versions of Exchange and NT, Windows 2000 and Exchange 2000 both contain the same version of Backup, and it is Exchange-aware. This means it can do a backup while Exchange is running and has files open. If you are running any of the earlier versions, they will not be able to back up Exchange 2000 properly, as shown in Table 28-1.

### Table 28-1
**Versions of NTBackup.exe**

<table>
<thead>
<tr>
<th>File Name, Version, and Size</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTBACKUP.exe, 1,129,000 12-07-99 7:00a</td>
<td>Ships with Exchange2000 and Windows 2000; is Exchange-aware</td>
</tr>
<tr>
<td>NTBACKUP.exe, 716,560, 3-8-96 4:00a</td>
<td>Ships with Exchange 4.x/5.x; is Exchange-aware</td>
</tr>
<tr>
<td>NTBACKUP.exe, 716,560, 7-15-96 3:30a</td>
<td>Ships with Exchange 4.0a; is Exchange-aware</td>
</tr>
<tr>
<td>NTBACKUP.exe, 675-488, 3-9-96 4:00a</td>
<td>Ships with NT4 SP 2; is not Exchange-aware</td>
</tr>
<tr>
<td>NTBACKUP.exe, 329,777 8-02-96 11:00p</td>
<td>Ships with NT4; is not Exchange-aware.</td>
</tr>
<tr>
<td>NTBACKUP.exe, 675,504, 9-23-95 10:57a</td>
<td>Ships with NT Server 3.51; is not Exchange-aware.</td>
</tr>
<tr>
<td>NTBackup.exe included with Windows 2000 Version 5.0.2172.1</td>
<td>Do not use to back up Exchange 2000</td>
</tr>
<tr>
<td>Windows 2000 Version 5.0.2195.1117</td>
<td>Has the right backup version for backing up Exchange</td>
</tr>
</tbody>
</table>

**Caution**

Always use a current version of backup. Older versions, even if they are Exchange-aware, do not work reliably on your newer information stores and might cause damage.

*Exchange-aware* means that you can see the Exchange Information Store as an object to back up and be able to back it up while it is online, as shown in Figure 28-4. It also means that you can do a restore that writes the incrementals and transaction logs back on top of a full backup.

**Note**

Depending on whether you are running older connectors, certain other files like the DX or PCMTA files may still require an offline backup.
Where is the data?

In earlier versions of Exchange, you had to back up the messages, the directory, and the transaction logs. In Exchange 2000, the directory information resides in Windows 2000 Active Directory Service and not in Exchange. To back up the directory, use the System State folder in Windows 2000 backup utility, which allows you to specify what to backup. Choices include

✦ Active Directory
✦ Boot files
✦ COM+ class registrants
✦ Registry
✦ Sys volume

If the objective is to back up sufficiently to be able to restore your Exchange environment to the same state it was in before a catastrophe, then there are files other than the obvious ones of the directory and the Information Store. Some of those files are in the Exchsrvr folder, which includes certain logs and items such as message tracking data (if you have message tracking turned on). If you do not ensure they are backed up and recovered, you will find it impossible to bring your system back into the same state as before a crash. Other necessary files, such as user-side files (including PSTs and OSTs), can be located in many different places, depending on how you have organized your system. Table 28-2 lists some of the files to be sure to back up.
Table 28-2
Data Structures

<table>
<thead>
<tr>
<th>File Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PST</td>
<td>Local personal storage folder file to download messages and contact information into for Outlook clients</td>
</tr>
<tr>
<td>OST</td>
<td>Local offline storage file for synchronizing local messages with server messages</td>
</tr>
<tr>
<td>PAB</td>
<td>Personal address book file (Outlook 2000 will try to convince you to import your PAB to Outlook’s contacts and stop using your PAB)</td>
</tr>
<tr>
<td>Log files</td>
<td>Come in various types including transaction logs used in backup and restore</td>
</tr>
<tr>
<td>Transaction log files</td>
<td>Can reside in MDBData and DSADATA or other directories and drives if performance optimizer is run</td>
</tr>
<tr>
<td>SCD</td>
<td>Schedule + 7.0 files (now replaced in Outlook 2000 with calendar information stored on the server and/or in a PST or OST)</td>
</tr>
<tr>
<td>Message tracking files</td>
<td>Stored in Exchsrvr folder</td>
</tr>
<tr>
<td>STM and EDB</td>
<td>Body and header of Internet messages as stored by Exchange 2000</td>
</tr>
<tr>
<td>SLV</td>
<td>Audio and video files</td>
</tr>
<tr>
<td>Messages in transit through MTA</td>
<td>MTADATA subdirectory</td>
</tr>
<tr>
<td>Connector files</td>
<td>Files that connectors (such as the Microsoft Mail for PC Networks connector) store in various subdirectories</td>
</tr>
<tr>
<td>Key Management Service Databases</td>
<td>Store information about key pairs used for advanced security</td>
</tr>
<tr>
<td>(SRS) database</td>
<td>Used for interoperability between Exchange 5.5 and Exchange 2000</td>
</tr>
<tr>
<td>Cluster quorum</td>
<td>Stores cluster configuration information (if you use clustering) as well as shared data</td>
</tr>
<tr>
<td>Scripts</td>
<td>Used for system or data management</td>
</tr>
</tbody>
</table>

How Backup and Restore work

In order to understand how to do a restore, it helps to understand how backup and restore actually work. Because the ESE uses a transaction log as well as a database, and because we want to be able to back up while online, there are a couple of file types with special roles in restoring that need to be handled in backup. The
Windows 2000, Exchange-aware backup handles them automatically, but you should still understand how they work, in case manual intervention is required.

**Log files**

ESE writes changes to data to log files. These changes are processed into the database later; how much later depends on system load and other algorithms for maximum performance. This design has many advantages, and one important implication: When you back up the database alone, the uncommitted transactions are in the log files, not the database. So you need to back up and restore both the database and the log files.

Ensure that the log files are on a separate spindle from the database if at all possible. When the system writes a piece of data to a log, it is appended to sequentially. When the system then applies this change to the database, the disk is accessed randomly. To maximize performance, drive heads should move as little as possible. This state is accomplished by assigning logs to one spindle and data to another, so that one head is writing log entries and parked near the end of the log file, ready to write the next sequential entry, while the spindle’s head is writing data somewhere else.

Each Exchange database log is exactly 5,242,880 bytes (5MB) unless it is corrupted. Logs include transaction logs, reserved logs, and previous logs.

**Transaction logs**

Transaction logs hold database changes before the changes are written to the database as sequential data. You have one active transaction log for each storage group. The location of your server’s log file path is controlled by the storage group property page; an example for First Storage Group is shown in Figure 28-5.
If you are running the default server installation, your storage groups are set for circular logging off. Turning it on (default in earlier versions of Exchange) means that there will be only one log file. In rare cases (for example, circular logging was set off, a successful full backup was not completed to remove the transaction logs, or circular logging was set back on), there could be more than one log file even when circular logging is set.

Other items on that same property page include the location of the transaction log, log prefix setting and whether you want deleted database pages to be zeroed out. Zeroing out deleted pages increases security (the data after that cannot be recovered) but decreases performance. The system specifies log prefix.

**Reserved logs**
There are two special log files, res1.log and res2.log. These logs are 5,242,880 byte (5MB)-sized files which the system creates for emergencies. They remain empty until the system runs out of space for previous logs. If your disk runs out of space and the system cannot create a new 5MB-sized file, an error is sent to the service that tried to write to the database, shutting it down. If that service had transactions in memory, it needs to flush them all, so res2.log becomes active and assumes the role of a normal transaction log to clear those transactions. All transactions in memory are written to res2.log. Once the service is shut down, an event is generated noting the shutdown in the Event Viewer. When res2.log fills, then the system uses res1.log again. You cannot restart the services until you have sufficient available disk space to create transaction logs.

**Previous logs**
If the circular logging check box is not checked, each storage group will have more than one log file. These logs are called the previous logs. Exchange always writes to the active logs. When the active logs are full, Exchange renames the files, takes them out of use, and starts new logs. On a busy server with many transactions, logs are created quickly. Not all storage groups have new logs created at the same time. Previous logs remain on the server until you perform a successful full or differential backup of the database they have a signature for, using the Windows 2000 Exchange-aware backup or a third-party product that is Exchange-aware. If no one is paying attention to the backups and your backups are not successful, these logs continue to accumulate and could run your system out of space (another good reason to segregate the data and log files to separate spindles). Once you complete a backup successfully, the previous logs are deleted.

Never delete log files manually, unless you are absolutely certain that they are anachronistic and not part of the current backup set. Without the previous log files, not all the data required to bring a storage group or database back to current is available. Logs contain a signature that correlates to their database file. They cannot be restored to any other database.
Checkpoint file
In order to keep track of how much of the database has been backed up during an online backup, the system creates a file edb.chk, otherwise known as the checkpoint file. This file contains a pointer indicating the location in the edb.log file before which all transactions have been written and after which they have not been written. After each successful backup, edb.chk is reset.

Patch files
While Windows 2000 is backing up a running Exchange server’s log files, other services may still need to write changes to the associated database. For this situation, special patch files are produced. Patch files usually exist for a very short time (only during backup) and are usually visible on the hard drive only during that time. Because they are created during backup and erased after a successful backup, patch files could still be left on the drive if no successful backup was completed. In these files are transactions that wanted to be written to the area of the database that was already been backed up. There can be only one patch file for each database. The process works as follows:

1. Backup begins. The database is backed up and then log files begin backing up.

2. If services need to write to the database in an area that was already backed up, a PAT file is created.

3. Transactions coming in during backup are applied either to the logs and EDB database only (as normal), or to logs, database, and the patch file. The determining factor is whether they need to be written to the committed or to the uncommitted area of the database. After the backup is complete, the PAT file is written to the backup copy of the database.

4. The PAT file is deleted.

5. The next database is backed up, using the same process.

6. After backup, previous logs are deleted.

Backup strategies
Circular logging off is the default on the storage group property pages (as you see when you create a storage group). Leave circular logging off, but ensure that you have a good, written backup plan that you follow religiously. If you turn on circular logging off, you receive the warning shown in Figure 28-6; if you answer Yes, you must stop and start the information store to complete it. In our opinion, Microsoft should also have required permission from your mother so it could be really difficult.

Caution: With circular logging on, transaction .log files are filled and overwritten. That means they lose the information for the overwritten transactions. Unless you are doing a full backup more frequently than the logs are overwritten, you will lose data in the event of a failure that requires a restore.
Your plan should include a full backup to start and then incrementals until the next full backup. How often should you do full backups? The answer is the same one that mountain climbers give when novices ask how often to bang in a piton. “You will fall twice the distance to your nearest piton, if it holds. How far do you want to fall?” The same goes for full backups: If tragedy strikes and you need to restore, you will have to restore the most recent full backup and all of the incrementals, assuming they are all good backups. If not, you have to go back further, and you may not get all the information anyway. The more tapes you need to restore from, the more likely that you will have a failure and the longer it will take to restore. In small to medium-sized organizations, a weekly full backup with daily incrementals is common.

In your backup plan, include rotation of tapes and regular archive (put one of the full backups away for a year or two). Make sure your plan has provisions for the right number of tape drives and tape changers (if necessary). Also, ensure that you have offsite storage and a local fireproof media safe.

You may want to take backup a step further, with third-party applications (from CommVault, Cheyenne, or Norton) that do far more than simply back up and restore. For instance, Galaxy from CommVault is considered a storage-management product; CommVault products address issues such as reducing duplicate tapes simultaneously, minimizing the stress on backup tapes, providing context-rich search capability for individual messages to reduce the effort of small-scale restores, and managing the tape pool.

Include in your plan a criticality analysis, including how soon users expect data to be recovered. Think about whether the data is casual, important, urgent, emergency, or critical. If it is casual, then it may not need to be recovered fast, and sometimes not at all (though you should still have the ability to bring it back). If it is critical, then it probably needs to be able to be recovered very quickly for your organization to do its normal work.

If your system includes a cluster, you need to know how to back up Exchange virtual servers on cluster nodes, as explained in the Backing up data on a cluster server node section later in this chapter.
Backing up

In order to invoke a backup from the Exchange-aware Windows 2000 backup utility, you must do the backup from a machine that has Exchange 2000 or at least Windows 2000 installed.

First, decide what you are going to back up. Windows 2000 Backup can be configured to back up any Windows 2000 files, as well as all Exchange databases on any Exchange server it has connectivity to and permissions to access.

If you do not wish to select all the Exchange databases on a specific Exchange server, you can back up only specific databases or storage groups. You have similar options when it comes time to restore. You can restore all of the messaging information on an Exchange server, or you can restore individual databases, storage groups, or the Web store (drive M). Of course you can also use Windows 2000 backup to back up and restore non-messaging data. Figure 28-7 shows a backup of a number of individual storage groups: Engineering, First Storage Group, and Sales Storage Groups. The Quality Assurance storage group is not included, but could have been.

![Figure 28-7: Backing up selective databases on an Exchange Server](image)

Tip

Instead of backing up databases individually (although this is allowed), back up an entire storage group at one time. The reason is that backup and restore both involve backing up the log files, but the log files are maintained for a storage group, so backing up individual databases means backing up numerous log files more than once.
After you have decided what to back up, decide what type of backup you want to do. Alternatives include normal (full), copy, incremental, and differential, as shown in Figure 28-8. Your choice of backup dictates the type of restore you are allowed.

Figure 28-8: Selecting the type of backup

After you have chosen a type of backup, set schedule, description, append, or replace options and label for media if overwritten. If you click on the advanced button, you can specify Back up data in remote storage and Verify data after backup. Seriously consider the verify option, even though it takes more time and generates work for your server and wear on your tapes.

For more information on Windows 2000 backup and all the other sophisticated Windows 2000 backup utility settings, see the *Windows 2000 Server Bible* (Hungry Minds, Inc.)

### Restoring

At a presentation, a hand went up in the audience. “Excuse me, sir, you have been talking about backup for over a half an hour. Would you spend equal time on restore, as that is all we are really interested in?” The point to the question is often missed by us techies. It does not do any good to spend money or time on backup if you cannot restore. That observation leads us to strongly encourage you to verify your backups regularly and simulate situations where you have to do a challenging restore. Ensure that the people and the equipment are adequate to the task. Do not wait for catastrophe before fine-tuning your practices.

If you are trying to recover a storage group to the same Exchange server that it was created on, that server does not have to be offline while you are restoring; only the database being restored does. You will receive a warning from Exchange that services will be stopped, but the warning does not mean all services.
If you have multiple tape devices, you will be able to restore to multiple storage groups at the same time, but only if you have reserve databases (the system reserves one for the restore process). If you want to restore more than one storage group, ensure that you do not use all the available databases. Keep extras, if you intend to do parallel restores. Remember that you cannot do parallel restores to the same storage group simultaneously.

To restore a database or storage group follow these steps:

1. Decide exactly what it is you wish to restore. If there are multiple databases, they must be from different storage groups for simultaneous restore. If you want to restore multiple databases serially in the same storage group, specify that. If you start one restoral and then set another to start, the Restore.env file is overwritten by the second, causing a problem with your first database.

2. Ensure that you have enough disk space to handle both the data and the temporary files for the storage groups being recovered.

3. Invoke Windows/Exchange 2000–aware backup and select either the restore wizard or the Restore property page. Once you select what to restore, you will be asked where you want it restored to and a path for temporary log and patch files, as shown in Figure 28-9. Then you can start your restore.

![Figure 28-9: Selecting data for a restore](image)

4. Verify that the backup was completed, and look at any exceptions.

5. Verify that the data was restored correctly.
Verifying/validating backups

We cannot stress enough the importance of verifying your backups and restores. In both cases, be sure that you do more than launch the tape. Look at the logs after completion; verify that the activity completed and note the exceptions. Then verify the data (you should specify Verify data when you invoke a backup).

To be certain the backup occurred without errors, or to find out what errors may have been posted, look at the Windows 2000 backup log and the events in the Windows 2000 event log. There you will find entries that tell you if your backup completed as scheduled and whether there were errors. If you find errors or exceptions, research and resolve them instead of ignoring them; they could mean that the backups you have are not solid, or that future backups are likely to fail.

In large environments, you may not be able to verify all backups and restores all the time. If not, focus on the most important databases or storage groups, backing them up in a rotation so some are done each night.

Figure 28-10 shows the first screen in a restore. Here you specify the server that the data will be restored to, where to locate the temporary patch files and log, and whether this is the last backup set.

![Figure 28-10: Options for a restore](image)

Back up data on a cluster server node

You can use Windows 2000 backup when your Exchange server is loaded on a Windows 2000 cluster, as long as certain conditions are met:

- Exchange server must be running on a cluster node.
- The cluster service must be functioning properly.
As with other backups, you have options. You can set up the backup manually, or use the Wizard. If you select manual, perform your backup from the node that owns the disks (if what you wish to do is back up all the disks in the cluster controlled by that node). If you use the Wizard, on the screen that asks what you wish to back up, select Everything on my computer. If you do not wish to back up everything, select only the databases or storage groups of interest, using the Wizard or the manual process in much the same way as you would with a regular backup.

Cluster backup presents several caveats that are not a concern in a regular backup:

- Ensure that the node you want to back up can actually access the drive with the data, by verifying that it has ownership of the cluster quorum disk. Often this is documented when the cluster is set up, but if not, you may need to wait for a quiet period when you can stop the cluster service elsewhere and then back up.

- If you select Back up everything on my computer, you will back up all of the clustering software itself, as well as the node system state information. If you back up the System State data only, you will also get the quorum, but not all the other cluster software.

- You might see errors during backup such as Completed with Skipped Files, and find by examining the Windows 2000 Backup log that both clusdb and clusdb.log failed to be backed up. You can safely ignore these error reports and still have successfully backed up the quorum logs from the cluster quorum drive.

- You need back up only one node cluster quorum disk. If you have that, you can safely ignore backing up the other quorums on the other cluster nodes. You should still back up the clustering software on the remaining cluster nodes, to ensure that you have everything you need for a restore.

Permissions

In some larger organizations, the group that owns Exchange does not also own administration of Windows 2000. If you do not own Windows 2000, you need to either procure it or work closely with the Windows 2000 administrators, as you cannot accomplish most restores of Exchange without access permissions to alter Exchange entries in Windows 2000 Active Directory.

Speeding up recovery with a dedicated system

Many organizations want to be prepared for the possibility of a catastrophic failure and the probability of a speedy recovery. Many see the need for service level guarantees that include quick recovery of a mailbox. In order to achieve that, they often invest in a Dedicated Recovery System (DRS) and training. A DRS is a recovery LAN involving N servers with ADS, sufficient hard-disk space, and tape drives to recover
an entire server or any portion of it. Usually the servers already have the latest version of Windows 2000 running on them. This model has less utility in a Windows 2000 and Exchange 2000 environment than it did in previous versions of NT and Exchange, but can still be useful. In some cases, you are going to be forced to do restores right on the active LAN because of the tight integration of Windows 2000 ADS and Exchange. However, even then you are better off if you have all the hardware readily available and known to work.

Disaster recovery servers with removable hard drives are useful (you can buy carriers and chassis from Kingston [DataExpress] and others). In some cases, you could have the servers on a separate LAN run older versions of Exchange and Windows loaded on one set of hard drives, as well as a more current version providing quick changeover for legacy restores. You could also have multiple versions on the cold-swappable drives for multiple serial recoveries.

The obvious question of how long a recovery will take is an exercise in arithmetic. To get a rough idea, find out how fast your tape drive will read and write from a tape and divide that into the number of gigabytes of storage you need to recover. The good news about Exchange is you can store almost infinite amounts of information (16TB is the actual number); the bad news is that recovering infinite amounts of data takes infinite amounts of time, breaking your data down into appropriate storage groups and databases, and leaving adequate space for database repair.

For more information on this topic, see Chapter 8.

The half-full drive model

Many feel that it is a best practice to ensure that your disk drives with your databases never get more than half-full, allowing you to defragment them with space to spare. When a restore starts, it will not have to overwrite your existing database, as there is room for a second copy. You gain both time and security.

If your restore is unsuccessful, you must try to repair the original database or go to another, older backup. Because you may wish to repair the original, damaged database, back it up before you start a restore, unless you have enough space for two copies.

More differences between Exchange 4.x/5.x and Exchange 2000

In addition to the changes between Exchange 4.x/5.x and Exchange 2000 covered in Chapter 8, there are two more important differences.
First, in Exchange 4.x/5.x you were allowed one private and one public database. In Exchange 2000, you are allowed up to five storage groups; each storage group is allowed up to five databases.

Second, you can be running all the other databases while doing a recovery on one. If you have multiple tape devices and available temporary databases, you could be doing multiple restores while continuing to run all the other databases.

When backing up and restoring Exchange 2000 databases, you are prohibited from running multiple backup or restore processes in a single storage group.

**Item recovery option**

On the Private Information Store property page, you can set Item recovery on and specify how long you want the items to be retained, as shown in Figure 28-11. Setting Item recovery on has two immediate effects. First, when items are deleted by users, they do not get deleted in the Information Store until the period specified by this setting is satisfied. The store retains them, but hides them (much like your recycle bin). When the parameter set on the property page is satisfied, they are deleted from the database. Second, your Information Store grows in size immediately by the amount of additional storage required for the retention period. If the item that needs to be recovered is still in the Information Store on the server at time of recovery, the user can invoke the recovery. If it is not, you can restore the information store that would still have it to your recovery server, and then invoke the end-user recovery.

---

**Restoring with ADS as part of Windows 2000**

Some of those differences between Exchange 5.5 and Exchange 2000 are readily seen in restores. Now that the directory is not part of Exchange, restores can be more difficult. For example, you can no longer restore a single user by restoring just the Information Store and then running the consistency checker. The server must be on a dedicated recovery LAN for this operation, since you’ll need the AD restored as well.

Although you can have standby servers, you cannot install Exchange until you want to perform the server recovery. At that point, install Exchange in disaster recovery mode. During a disaster-recovery installation, information necessary to rebuild the Exchange server is retrieved from the AD. If you deleted the server object in the AD, you can’t do a restore.

If your Exchange server was the only one with AD, recover that first, and then recover Exchange.
Recovering an Exchange Server

Recovering an Exchange 2000 Server means that you need to recover Windows 2000, the Exchange 2000 software, configuration information, databases, logs, and all the other files that were on the machine before its demise. If you have taken advantage of the ability in Exchange 2000 to have multiple storage groups and databases, you need to recover all of them and their associated transaction logs.

For a full server recovery, follow these steps:

1. Reinstall Windows 2000, using disaster-recovery mode to get critical pieces of ADS back. If the server is part of a cluster, this reinstallation will require multiple steps.
2. Reinstall Exchange 2000 Server, selecting the options appropriate for disaster-recovery mode. Do not try to recover Exchange data yet.
3. Recover any other role-specific files, such as KMS or SRS. This recovery requires steps different from or in addition to a restore of servers not running these services.
4. Once the Exchange server is running properly with all the roles assigned to it, recover Exchange data.

Before you attempt your restore, attend to the following items:

✦ Repair, replace, or enhance any failed hardware.

✦ Procure all software and license information, including all applicable service packs, license keys, or software updates as specified in the Windows 2000 and Exchange 2000 Release Notes and on the Microsoft Web site.

✦ Assemble and verify backups of the system drive, and of any drives where you have placed required applications or data. Ensure that they are both good backups, and that they have the correct time and date, by checking backup log files.

✦ It may be possible to recover information by running database utilities on your damaged database(s). Make a backup of damaged databases (if possible) and ensure the backup is “good” by checking the backup log file. You might have to do this offline, if you cannot get a backup with Exchange running. To ensure that you have everything, include backups of Web Storage System and any special sever role databases (for example, SRS and KMS server databases).

✦ If your server is a Windows 2000 member server, verify that ADS has an object for the Exchange 2000 server you intend to restore before you start the restore.

Caution

If there is no object in ADS for the server you intend to restore, you cannot continue with your restore. Do not attempt to recover the server until you remedy this deficiency. The Exchange 2000 Server you intend to restore must appear as an ADS object before you start your Exchange reinstall.

✦ You must have a functioning domain controller. If the only domain controller in the ADS domain is the Exchange server you want to restore, perform an ADS restore to bring up the domain controller and all its information before you attempt to restore Exchange.

✦ You need to have Windows 2000 running with all the system-state information existing before the Exchange server failure. You should use a Windows 2000 backup that backs up system-state information such as configuration information, Windows registry, and IIS metabase.

Tip

Because of the complexity of restoring an Exchange server when it is also a domain controller, build Exchange servers as member servers.

Here is a generic version of how to recover a complete server. These steps must be taken in order:

2. Install the same version of Windows 2000 (Server/Advanced server/Data Center).
3. Make sure software is installed in the same path and logical drive.
4. Use the same server name as the original.
5. Install the same Windows 2000 components.
6. Make the server part of a workgroup. Do not add it to the domain.
7. Install the same service pack level, and any appropriate patches or hotfixes.
9. Restore static application files and data (system software, any other static data).
10. Restore the system state information including Registry and IIS metabase.
11. Restart the machine, ignoring any services that do not start (SMTP will not work until Exchange is installed).
12. Install Exchange in disaster recovery mode by using the /disaster recovery switch.
13. Restore Exchange databases.
14. Restore SRS.

Getting by Firewalls

Firewalls exist to protect us, but they create special problems for Exchange organizations. For example, suppose you are a consultant, and your customers have rigorous firewalls and have closed down most ports. They may be unwilling to open ports for your Outlook client, or even for IMAP4 or POP3 clients. In that case, your users may have to use OWA to connect to your server.

Or perhaps your users want to VPN from one of your sites to another to connect to their Exchange server. Microsoft VPN requires RRAS if there are two proxy servers in between the VPN client and the server.

There are many more possibilities, but the important message is to ensure that your firewall and topology are in synchronization so that the ports you need open are available. If they are not and you are not allowed to open them, alter your topology (use relay hosts, front/back architecture, and so on) so that the utility is not lost in an effort to protect the company.

Summary

Many areas in Exchange need your attention, but some were not big enough for whole chapters, and so they are combined here: performance profiles for small, medium-sized, and large organizations; memory management (to ensure you understand how it works in Exchange 2000); slow-speed links and ways to work around them; mixed Exchange environments and roving, wireless and remote users; backup and restore (a crucial part of the job of every administrator); and the challenges that firewalls pose and ways around them.
Difficult technical problems can cast a shadow over your life that appears immense. Big problems affect many users and vital business activities, creating stress for all and pressure everywhere. In a sense, that is not surprising, because Exchange may be the most important group knowledge tool of the company (certainly messaging and groupware are right up there with telephony, FAX, and pagers).

But no matter how big a messaging system problem you think you are facing, there are two reasons to remain calm. The first is perspective: big, overwhelming problems are war, world hunger, and irreversible damage to the planet, not Exchange 2000 glitches. The problems you will probably encounter in working with Exchange and Windows 2000 may be serious or complex, but they are not overwhelming, and they can be solved. Breaking down the problems into constituent pieces (as shown in this chapter) makes them addressable, and therefore fixable.

The second reason to remain calm in the midst of an Exchange or Windows 2000 crisis is that there has never been a messaging system that could not be fixed with sufficient budget, effort, resources, and time. This chapter shares the basics of troubleshooting.

The Troubleshooting Process

To address complex problems in an Exchange messaging environment, break them down into smaller, less complex units. As you solve the smaller problems, you will usually find that the overall problem becomes less and less complex. This chapter cannot tell you how to solve all the problems you will encounter, but it should help you assess where to look and what types of tools to use to address the problem; and it can give some real insights into what to do.
Getting a handle on the problem

In order to understand the problem you are facing, you need to examine the environment and system. Therefore, the very first thing you should do when a problem arises is run through a checklist:

✦ Collect all written information available, including specifics of error messages and system documentation (including topologies).

✦ Procure either passwords and access, or the people with passwords and access.

✦ Look at the Windows 2000 Event Viewer for all servers that might be involved. If necessary (that is, if they are overwriting previous data), alter them so they are at least 2.5MB. Otherwise, you might not retain the information you need to solve the problem; it could be overwritten before you see it. (Now that disk space is inexpensive, you can set the event log overwrite to off.)

✦ Look at the real-time monitors on each server, as shown in Figure 29-1. Look at the performance monitor and other real-time instruments to assess the health of the server and services.

Figure 29-1: Real-time monitors

✦ Set diagnostics to minimum on all the Exchange processes that may be related to the problem (unless you are certain where the problem lies). This step reduces clutter and information overload. It also helps to keep your event log from filling up or overwriting with information unimportant to the problem. Once you are more certain of what you want to examine, increase or decrease the settings selectively. If the problem is intermittent, you may need to leave certain logs set higher in order to capture the traces of the problem. Some events are hard to reproduce.
If the problem has to do with flow of messages and address spaces or connectors, enable message tracking on all servers.

It takes time before messages can be tracked. If message tracking is not turned on already, turn it on as a good long-term strategy. It will not help short-term problem solving.

Use the Active Directory and Exchange System Manager to verify what the topology looks like. Determine what level of access you have to remote Exchange servers (you may need to alter property pages at remote servers). You may need to have someone in the local site available to make changes for you, or you may need additional IDs and WAN access.

Make sure performance monitors are running for MTA, connector queue management, and server health. If the server has more than one processor, make sure you look at all of them. If the problem is with performance, don’t run them on the local machine.

Use Help About to verify which versions of Windows 2000, Exchange 2000, and Exchange Client software are running.

Make sure you have access to the Microsoft Web site Knowledge base and Web site, as shown in Figure 29-2.

Collect symptoms.

Figure 29-2: Microsoft Exchange Web site
With this information, you should be able to understand enough of the system and the symptoms to start determining the problem. From here, the diagnostic path tree branches out quickly, depending on what you find and depending on the nature of the problem.

Using the information available

Exchange and Windows will provide you with a significant amount of information, but like the runes in a fantasy novel or on an artifact, the information may not make much sense to you unless you have studied the lexicon and understand how to interpret it. The areas that provide the most information for use in troubleshooting are covered below, but this is not the exhaustive list.

✦ **System log:** The Windows 2000 Event Viewer has multiple logs. The system log helps you discern Windows 2000 Server Service problems. Typically you will see alerts for running out of space, server shutdowns and startups, hardware problems, etc. After finding the item that piques your interest, double-click to open it up and focus on the detail.

✦ **Application log, security log:** The application log and the security log are also available in the Windows 2000 Event Viewer. Of the two, the application log, shown in Figure 29-3, is the more important to your troubleshooting effort, as all of the Exchange Services run as applications. When you turn diagnostics up on the many Exchange Server property pages, the application log is where they are most likely to write their information. As with the system logs, find the entries that are compelling and double-click to examine the information.

![Figure 29-3: Event Viewer application log](image)
✦ **Directory log:** Because Exchange 2000 relies on Active Directory Service and does not have its own directory, the log for the Active Directory Service is very important in most Troubleshooting.

✦ **DNS Server log:** Insights into many mail-flow and connector problems come from this log. If DNS entries are wrong or inadequate, many of your Internet and connector underlying protocols suffer and may fail.

✦ **File Replication Service log:** The File Replication Service makes entries into this log. It is not as useful as the other logs in most Exchange troubleshooting, except where replication is the issue.

✦ **Setup log:** Exchange Server generates a setup log when installation takes place. It is found at the root of the destination drive in which the Exchange Server was installed, and called Exchange Server Setup.log. This log can be read in Notepad text.

✦ **Dr Watson log:** Dr. Watson is installed with Windows 2000 Server. It monitors applications and dumps vital information when an application crashes. It does not dump information when you have a low-level catastrophe such as the infamous Blue Screen of Death (though they are rare in Windows 2000), but it does provide vital information for many Exchange effecting software conflicts, bugs, or problems.

The most important information in the Dr. Watson log is usually the time, date, and header information, as well as the last or second-to-last DLL listed. Often one of those DLLs is the culprit, or an indicator of which system caused the failure. The last or second-to-last DLL or application listed in the Dr. Watson dump is the place to start looking for the root cause, or clues to the root cause. The rest of the dump information is often helpful to Microsoft, so do not discard it, but do not dwell on it either; it is not helpful unless you want to go very deep into the analysis from a programmer’s perspective. Remember, Dr. Watson appends the latest dump onto the end of the log; start from the bottom and work your way up.

If you don’t want to try to remember where the logs are found, do a search on *.log from the root of the drives that have Exchange or Windows 2000 loaded on them. Also, take advantage of the filtering options in the Event Viewer to limit the scope of your search.

Debugging Windows 2000 drivers and applications using utilities such as Dr. Watson often requires the Debug Symbol files. These are special debugging files specific to the version of the operating system or application code and normally need to be updated with each service pack. Symbol files should be installed for the operating system as well as applications like Exchange Server in order to facilitate debugging sessions with Microsoft. Without these symbol files, the information in dump files is often incorrect or incomplete. Seriously consider installing the symbol files for Windows 2000 and Exchange 2000 as a part of your server-build standards.
Non Delivery Reports (NDRs): Do not ignore the NDRs that your users receive. They often indicate which machine had problems with the message, and what type of problem it had. Usually you can discern whether the originator of the NDR thought the cause was a system-level problem or a problem with one particular e-mail only. Look closely for the MTA/IMS that returned the message. If it came from a local MTA, it may be a connector or routing problem. If it was bounced from a remote MTA, it is more likely to be a problem in the address.

Make sure that you get original copies of NDRs if they are available. If the originator of the message is using Outlook, have him or her send you the actual NDR message (not forwarded or replied to). A user can send an original copy of a message by using the Insert Item menu option from the Outlook application. Getting the original NDR is particularly important if the message has been routed through the Internet, because it will contain all of the SMTP routing and header information for the message. To access this information, open the message, and from the View menu choose Options and look in the Internet Headers field.

Connector notices: The connectors send administrative details if configured to do so. This is especially useful with the Dirsynch connectors. Figure 29-4 shows an example of an inbound Non-Delivery Report (NDR) or notice.

Link monitors and server monitors: If configured properly with notifications, these provide a good indicator of the overall status of your servers. The notifications can be configured to trigger an e-mail, page, or other action when problems arise.

Replication monitor utility: You can use the Replmon.exe utility to check your logical replication topology and the status of individual replication links. If events in the Active Directory log are indicating replication problems, consider using this utility.

Administrative notices: The System Administrator (SA) is your friend and often sends you e-mail, if configured to do so, advising you on issues with the system. Figure 29-4 shows a message generated by the System Administrator indicating that MTA congestion prohibited a message from going through. Based on this message you would investigate the MTA queues, stop the MTA, run MTACheck.exe, and review the event log.
Most common causes of problems

Robust Exchange implementations can be quite sophisticated, and problems can crop up in many areas. Here is a list of some areas where problems may appear:

✦ **Server runs out of disk space**: If you run out of space, your information store and/or routing engine will stop. You will also probably fragment the drive severely. With the low cost of disks today, you can probably afford to buy ample disk space and should work to prevent this. However, if you do not have the right controls on your system and some of your administrators do not understand, they could cause this problem in many different ways (for example, by forcing a replication of a large and unexpected public folder onto your system). You may be able to control space by managing administrative group permissions so that only people attentive to this issue have permissions to manipulate data administratively.

✦ **Permissions**: If you manage permissions (Client permissions, Public folder permissions, All permissions) astutely, you will preempt many types of failures. Insufficient permissions, however, can cause problems and can prohibit you from fixing them.

✦ **Name resolution, DNS or AD unavailable**: Many services and protocols rely on name resolution. Without it, they cannot work properly.
Incorrectly addressed messages: These can result from the use of .pab files or contacts. Another cause of incorrectly addressed messages often occurs in migrations or when old data is restored. Users responding to e-mails generated when the old address was valid, or finding stale information in their contacts and address books, may create e-mail with currently undeliverable addresses. Similarly, many users add incorrect information and have multiple address books or contact lists with stale information. Users should go to the server-based Global Catalog first, and then their local contact lists.

Hardware: Hardware failures or even severe slowdowns (especially involving disk controllers and memory modules) can be caused by physical problems or low-layer disk issues. Checking for errors, monitoring the rate of increase in errors, using disk defragmentation (as shown in Figure 29-5), and other Windows 2000 basic utilities help your Exchange server to run at its level of ability.

![Figure 29-5: Recommendation of disk defragmentation]

Other information tools
Most larger organizations subscribe to Microsoft Premier Support. Many medium-sized organizations either have Premier Support, enlist the services of consultants who use Premier Support or have other access to the deeper resources at Microsoft. From its earliest days, Premier Support has had specialists focused on messaging systems. They can bridge the gap between the product development and bug fix teams and the customer and consultant support teams. They can even be instrumental in contacting engineers at other vendors to tackle problems with vendor products affecting a customer’s Exchange system. When troubleshooting, consult Microsoft Premier Support and triage on the basis of:
✦ Severity of the problem
✦ Whether it looks like a bug in a Microsoft product
✦ Urgency
✦ Requirement for on-site consultants
✦ Cost

In some cases, the fastest and most cost-effective solution is to hire on-site consultants who have access to Premier Support.

TechNet subscription
When working on Exchange or any of the Microsoft Office or BackOffice products, you may want to have the recent TechNet CD-ROMs nearby. TechNet, which contains quite a bit of useful information, comes in subscription form and provides regular updates to both the knowledge base and the Patches and Drivers CD.

Online Knowledge Base
While TechNet is useful for many types of queries, the Internet-based Online Knowledge Base is better for late-breaking news. It is easy to use with browsers such as Microsoft Internet Explorer 5 or other leading browsers. Figure 29-6 shows the Wizard that enables you to easily navigate through to the topic support, and then to Exchange IMC.

Figure 29-6: Online Knowledge Base
Use the Internet Microsoft or TechNet Knowledge Base before you call your Solution Provider or Microsoft. It is much cheaper, and both of them will start there if you haven’t already. Take time to learn how to use sophisticated queries. There are often articles on your subject of interest if you know how to generate a well-formed query. Use favorites and history lists to help you get back and navigate. Print only what you need. Using the screen is often easier than using paper, and it saves trees.

**Exchange list servers and Web sites**

A number of list servers and newsgroups contain information about Exchange. Inquiries to list servers and newsgroups tend to work best for non-immediate requirements. An example is shown in Figure 29-7. However, if you have stored articles, list servers and newsgroups may also be good for research on immediate issues.

![Figure 29-7: List server or newsgroup response](image)

In list servers and newsgroups, sometimes you get an accurate response to a well-formed question, sometimes not—and occasionally you get a seriously wrong answer. Learning from others is a good educational technique, as long as you do further research to be sure the answers are sound. Active participation in newsgroups and list servers helps lower costs and increase knowledge. For efficiency, rather than having all your people subscribe, consider subscribing with the e-mail ID of a Public Folder and propagating the information using Public Folder Replication. See the CD-ROM for more information on useful URLs and Newsgroups.
**Protocol analyzers**

Protocols underlie all transport on your messaging system. When protocols are not working, you may need to use a protocol analyzer to capture packets and study the packet decode. There are many different protocol analyzers on the market; in a Microsoft environment you will often see SMS’s protocol tool or McAfee’s NetWork General products.

A protocol analyzer provides three important pieces of information:

- Bandwidth use
- Protocol decoding
- Packet statistics (number of dropped packets; percentage of broadcast or multicast packets)

When setting up Exchange systems, you need to know that the underlying network is solid. If the underlying protocols are broken or their flow inhibited, you need to know before you implement an Exchange solution. It is very common for protocol and LAN or WAN problems to be blamed on the messaging team, even if the opposite is true.

To ensure that the underlying protocols and LAN/WAN are solid, run a protocol analysis baseline during which you capture protocol traces and information over a 24- to 48-hour weekday period. By analyzing the results, you will learn about the common protocols, average transmission, and regularly occurring spikes.

A protocol analyzer can capture communication between two X.400-connector RAS machines. The RAS port looks like another network card and can be viewed at a protocol level with a protocol analyzer to resolve certain types of problems.

Some protocol analyzers also handle WAN protocols, so they can be used to verify bandwidth and protocol support across lines used for connectors.

If the underlying protocol connectivity is not solid, there is no hope for RPCs. RPCs bind to an underlying LAN or WAN protocol that must be sound for the RPCs to work. If network bandwidth is insufficient, then there may still be problems with RPC communication, as RPC expects reliable high-speed lines.

**Applying logic**

When dealing with complex systems, especially when changes are being made, problems are bound to occur. The troubleshooting tools described through this book and summarized in this chapter can prove invaluable in problem determination and correction. But applying random tools haphazardly is rarely efficient.

The best method is to determine the problem deductively. In so doing, do not confuse the actual problem with the symptoms of a problem. For instance, the failure of the SMTP connector to pass outgoing messages may not be a problem with the SMTP connector; the real problem might be related to the DNS or some other prerequisite service, protocol, or physical device (router, hub, and so on).
Basic troubleshooting method

The problem-solving method presented here may appear a bit rigid, but it is meant (for the uninitiated more than the advanced sleuth) to provide insights into types of problems and the logic used to solve them. It reveals the type of processes you might use when troubleshooting problems in an Exchange environment and the tools that might be helpful in assessing the problem and resolving it.

1. Collect information and assemble tools.

2. Build a hypothesis: What do you think is going on? Even if you cannot get that far, build a hypothesis about what components seem to affect one part of the problem. Try to create a hypothesis leading to a solution that moves the problem away from the end-users quickly. Suggest a test of your hypothesis.

Getting the utility back to the end-users (that is, making sure the users can do real work) is the way you move the problem from HotSite to problem. In fact, our very definition of HotSite is any problem that prohibits a number of end-users from doing their work.

3. Collaborate on your hypothesis with your peers and other external resources such consultants and Microsoft personnel. They may know more than they have communicated to you about the symptoms, and they may be able to comment readily on the likely success or failure of a test of your hypothesis.

4. Test your hypothesis in a controlled way, one variable at a time. Try to work toward being able to toggle results (setting a parameter one way yields a certain result, and another way yields another result). If a particular change does not produce the intended result, try to have a backout plan to return you to the original condition. Further randomizing your systems will not help your troubleshooting process.

5. Document your results. Memory is not good enough when working in a team, or on complex problems over long periods of time. Sometimes problems reoccur, so it helps to have documentation on problem fixes. Having documentation will also help you do a post-mortem write-up for your management team.

6. Once you have one piece of the puzzle, build a hypothesis and test for the next; repeat the process.

7. Keep going until you can address the root cause of the problem. If you stop too early you may have a workaround, but not a solution to the root cause. Minor issues tend to compound over time if not addressed.

8. Test the results from the end-user and business perspective. Too often, an exemplary effort ends in defeat because the technical team solved all the technical problems, but forgot one little thing (such as restarting a service), so end-users still could not do their work in the morning. Do some testing as if you were a normal end-user (that is, do not log in as an administrator and do testing; your results may not be the same as those of an end-user).
Examples of problem-solving

Here are a couple of examples of the application of basic scientific method and simple logic-tree techniques for solving Exchange problems using the available tools.

Not all problems originally categorized as Exchange problems turn out to be Exchange problems.

Example A: Inconsistent mail delivery

Problem
Messages are being delivered inconsistently to the Internet. Symptoms include

✦ Some users in the organization get NDRs when sending to specific domains on the Internet. Others can send to these domains with no trouble.
✦ Internal messages are working flawlessly.

Solution
1. Turn up diagnostic logging on SMTP services.
2. Check event logs: nothing significant.
3. Request that senders provide NDRs. Comparison reveals that a similar error — “Sender domain must exist” or “Sender domain must resolve” — is generated by different domains on the Internet.
4. Search the Microsoft Knowledge Base. Hits on “Sender domain” point to security setup on the inbound SMTP server that does a reverse lookup on the user name to ensure it is valid.
5. Look in Active Directory to find that all users that get rejected messages have the same e-mail domain: Southwest.Company.com.
6. Contact the ISP to find out if a reverse lookup zone exists for Southwest.company.com; it does not. Once the zone is created, messages are delivered to all.

Example B: Mailbox access dilemma

Problem
An Outlook 2000 client cannot access its mailbox.

Solution
1. Start the client.
2. Select Tools ➪ Services ➪ Exchange Server.
3. Verify that the server and correct alias are listed.

4. Re-enter the alias name and click Check name. If it becomes bold and underlined, then you know connectivity as well as RPCs between the client and server are working.

The next step is to look at permissions. If the server fails to verify the server is an alias, investigate connectivity problems using tools such as RPCPing.

**Example C: Mail delivery quandary**

**Problem**

Outbound Internet SMTP e-mail is passing, but no inbound Internet mail is being delivered.

**Solution**

Recently the SMTP connector had been moved from one Exchange server to another. The team didn’t have time to plan; instead, they employed the motto *carpe diem*, even though it was the middle of the night. Predictably, the move of SMTP service was not reflected in the firewall or DNS.

After corrections were made, Internet mail still was not being received. Next the team checked the Cisco router and found it was still filtering to the old Exchange SMTP server.

Once all settings were fixed, inbound Internet mail passed. No mail was lost, and no NDRs had been sent to the originators as the ISP had been storing the mail it could not deliver during the ten-hour period in which it took to find and solve the problem.

**Example D: Message flow in an Exchange enterprise layout**

The Valimar LLC company has three Exchange sites on a WAN—Massachusetts, South Dakota, and California—with the following features:

- Single Forest and domain
- One Exchange server in each site
- Sites interconnected via a Routing Group Connector

The Massachusetts site has an SMTP Connector. The company has registered its SMTP domain name as Valimar.com, so incoming Internet mail is addressed to username@Valimar.com

**Problem #1**

While the users at the Massachusetts site are able to send and receive Internet mail, users at California and South Dakota cannot.
Solution to problem #1
The inbound Internet mail is rejected by the California and South Dakota sites because addressing at these sites has been configured as Valimar.California.com and Valimar.SouthDakota.com without adding a second address space entry to reflect the company’s registered SMTP domain name Valimar.com. At the California and South Dakota sites, valid addressing looks like @enterprisename.sitename.com. Because of this, the Massachusetts MTAs had knowledge of South Dakota and California users as username@enterprisename.sitename.com instead of the desired username@Valimar.com. E-mail for username@Valimar.com at these sites was rejected; the MTAs in Massachusetts had no notion of where to forward incoming Internet mail that was addressed to username@Valimar.com.

To correct this mail forwarding issue, the entry for *.Valimar.com to the address space property page in Active Directory for users is added to the others in California and South Dakota, so their users have multiple SMTP addresses in the namespace. This change makes username@Valimar.com valid again for all users and set as the return address for all users. After the directory replication, the Global Catalog in each location is updated with these changes. Now when Internet mail arrives for users in California and South Dakota, the Massachusetts site knows that it is valid for the recipients and knows how to forward it to them.

Problem #2
The outbound SMTP mail from South Dakota and California is not being relayed to Massachusetts and then to the Internet Mail Connector at Massachusetts, because South Dakota and California Routing Group connectors to the Massachusetts site are not configured with SMTP address space. The connectors do not know that they are supposed to be able to pass SMTP mail to Massachusetts.

Solution to problem #2
To address this problem, a cost of 1 is added to the Routing Group connectors in California and South Dakota while leaving an * e-mail domain. This change ensures that these connectors are responsible for passing all outgoing SMTP messages from South Dakota and California to Massachusetts.

As a fault-tolerant option the administrator is asked to configure the California-to-South Dakota and South-Dakota-to-California Routing Group Connectors with SMTP Address Space with cost of 3 and * e-mail domain field. This way, if a direct link from South Dakota to Massachusetts or from California to Massachusetts is down, SMTP messages can be transmitted to Massachusetts via the other site.

Example E: Setting up event sinks and event registration
You have created a purchase requisition public folder. For this folder, you have created a synchronous event sink called PurchaseReqs.mySyncEventSync that you set up to be invoked for OnSyncSave and OnSyncDelete events. Additionally, you
set up the event sink to be invoked only if the content class of the item being saved or deleted is http://content-classes/contoso.msft/purchasereqs. You also set up the event sink to be executed for all operations in http://servername/public/PurchaseReqs/folder, and any of its subfolders, before any other event sink is executed.

You open the Purchase Reqs public folder and you create a message and post it to the folder.

**Problem**
You can’t tell if the event fired or not.

**Solution**
In order to see what happened, turn Diagnostic Logging on.

Select your server in the Exchange System Manager, as shown in Figure 29-8.

![Exchange System Manager](image)

**Figure 29-8:** Select server in Exchange System Manager

Once you have selected your server, you can right-click on it to bring up the Properties dialog box. Select the Diagnostic Logging tab, as shown in Figure 29-9.

Expand the MSExchangeIS service and select Public Folder. In the Categories list box on the right-hand side, select Rules. Set the logging level to Maximum (default is none).
Troubleshooting

Figure 29-9: Diagnostic Logging

Now you can view all the events that occur in a public folder and trace what happens when you post a new message to the folder.

Server Role Assignments

In most organizations, not all servers are assigned the same roles. Even in a quick install, some responsibilities are only assigned to the first server by default. As the installation grows and you add connectors or users, they are assigned to specific servers, further differentiating the servers. Server assignment differences can complicate troubleshooting. Specific server role assignments should be deliberately assigned and well understood in a well-designed system to avoid downstream problems during troubleshooting. This section presents some examples of server roles to scrutinize.

First server in a site

The first server in an organization is the one that you need to specify the Organization name to. The Organization name is case-sensitive and has to be exact.

Also specify the Service Account at this time. All other servers joining an existing organization get this information in a stub from the target server. If the Service Account is specified incorrectly, it cannot be corrected without reinstall. By default, the first server also holds the Schedule plus Free Busy folder and calculates the Link State Routing Table. It is also the Recipient Update Server, Public, Private, and Hybrid Servers.
Some organizations dedicate servers as public folder servers with no users assigned to mailboxes on them, and private servers with no public folder data. Hybrid servers that have no public folder content, but retain a pub.edb, can also make sense; as long as pub.edb is there, you have a database structure to put public folders in while working on some other public folder server, even if the folders only stay there temporarily. Leaving the database structures (even when not using them in production) on servers designated as private store servers for emergencies is a common practice.

**Bridgehead servers**

Connectors require bridgehead servers, through which messages are focused. Multiple bridgehead servers are possible when using multiple connectors. Often, in larger environments, bridgehead servers are dedicated machines, because of where they sit in the topology (sometimes outside of firewalls) and because they often work hard.

**News servers and conferencing servers**

Conferencing Servers and NNTP News Service servers are sometimes set up as dedicated servers because of the load placed on them in very active environments. Especially when streaming video and other high-bandwidth uses are contemplated, this configuration can be an excellent way to ensure normal e-mail is not affected.

**Other roles**

Some companies segregate their POP3 or IMAP4 users onto dedicated servers.

**Backup servers**

The servers that do backups and restores are critical to every organization and can work quite hard. Windows 2000 Server Backup can back up any server in your organization, but doing that across most WAN lines is not common. When assigning backup servers and deciding what they should back up, consider the available bandwidth between the backup server and the servers to be backed up. Also consider whether the LAN bandwidth is adequate, and make sure you have sufficient devices (tape and hard drive spindles) and appropriately configured storage groups to allow parallel backups and restores.

**Additional admin workstations**

It is useful to have Windows 2000 and the Exchange System Administrative program loaded on a second Windows 2000 machine in each physical location, for redundancy. The Administrative program should be at the highest revision level of servers it will administer. This machine can be used for service monitoring, performance monitoring, or logging to multiple machines.
Summary

The troubleshooting examples presented in this chapter give you a sense of the many sources of information built into Exchange and around the Exchange community. You should now have a good idea of how to approach administrative troubleshooting, development administration issues, and other common types of problems, such as controlling message flow. In many cases, you can use the same techniques to solve problems no one knows are there. For instance, controlling message flow with artistry can save money. If messages are flowing no one may be complaining, but costs could still be reduced if messages flowed differently. If you want kudos, baseline costs and then use the troubleshooting methods to lower costs. While there are myriad other ways to procure information on your system and many other problems that you will encounter, you are now equipped to face them without trepidation. You know the basics.

✦✦✦
Exchange 2000 is a development platform for building messaging and collaboration applications. The platform is built on standards, and ties closely to other Windows 2000 development capabilities as well as those of other BackOffice applications. Knowing how to administrate Exchange when it is being used as a development platform is an important part of the role in which many Administrators are cast. Even if you are not a developer, you need to have sufficient knowledge of what developers do on this platform and how they do it to administrate your Exchange environment. By knowing a bit about development, you can also use new features like Active Directory Service Interface (ADSI).

In this part of the book, we explore administering applications on Exchange 2000. We introduce you to the different types of Exchange applications, with a focus on collaborative solutions with Outlook 2000 and server-side collaborative solutions using the Exchange 2000 Collaborative Data Objects (CDO). Then we introduce you to concepts that help you programmatically administer Exchange 2000 applications using the Active Directory Service Interface. We close with a focus on administration of both client-side and server-side applications, addressing issues in folder management, event sinks, and security.
An Introduction to Exchange Applications

People collaborate at work in different ways. Exchange 2000 is built to support a number of collaboration styles. This chapter explores the different types of collaboration possible with Exchange, and shows you how to use Outlook 2000 and Exchange 2000 Collaborative Data Objects (CDO) to build some collaborative solutions.

Overview of Collaborative Solutions

A collaborative computing environment can reduce a number of barriers to cooperation in an organization. You and your colleagues can work together independent of organizational hierarchy, local time, and geography.

A collaborative solution enables groups to work together in virtual teams to collect, organize, distribute, and track vital information across the organization. An effective collaborative solution streamlines workflow so that colleagues can interact efficiently, find and share information, collaborate on documents, and publish information to the company intranet or to the Internet.

Examples of collaborative solutions in the business environment include workflow and sales automation, document collaboration and e-mail, scheduling, and discussion groups and newsgroups. All of these applications are made possible through the collaborative environment provided by Outlook 2000 and Exchange 2000.
Collaborative solutions can be grouped into two categories whose names refer to time: *synchronous* (at the same time) and *asynchronous* (at different times). For example, talking on the phone is a synchronous information flow (especially when you both talk at once). On the other hand, leaving a voice mail message and having the person call you back is an example of an asynchronous information flow.

Synchronous collaborative solutions happen simultaneously. All the users see changes to information immediately (or, if not immediately, at least as quickly as latency in your underlying environment allows). Examples of synchronous collaborative solutions are Chat, Exchange 2000 conferencing, and NetMeeting. In a synchronous collaborative solution, a user can distribute information to other users only when their schedules have been coordinated and they are meeting at a designated time at a virtual location.

Asynchronous collaborative solutions do not happen simultaneously, but according to users’ own schedules. Examples of asynchronous collaborative solutions are voice mail and e-mail. In an asynchronous collaborative solution, a user can distribute information to other users at any time and without requiring any coordination of schedules.

**Types of collaborative solutions**

There are five basic types of collaborative solutions:

- **Instant collaboration**: These solutions use built-in customizable mail, calendar, tasks, journal, and contact folders that are included with Outlook 2000 and combined with Exchange 2000 public or Web folders.

- **Discussion group**: These solutions enable users to conduct online discussions. Users can communicate asynchronously by using post forms to place items in a folder. Users can also post responses to items in a folder. Discussion group solutions are the foundation for threaded conversations in views, whereby a user can view the history of responses for a particular item of interest.

- **Routing and tracking**: These solutions enable a user to record and view information that is being constantly updated. Routing and tracking solutions can pass information to individuals in a sequence or to a group and can track who has approved the information and who has made any changes. It also enables users to vote on an issue associated with the information they are provided. An example of a routing and tracking solution is an expense report: submitted by employees and routed to their boss for approval, then routed to accounting for approval, then routed to payroll for payment.

- **Reference**: These solutions serve as a repository of information in different formats (Microsoft Office documents, graphics, images, voice mail) that can be stored as items in folders. An example of a reference solution is a repository of sales and marketing information that can consist of brochures, PowerPoint presentations, video, and so on.
✦ **Digital Dashboard:** These solutions facilitate the customization of a single Web page for a particular knowledge-worker. The customization involves deciding on which information from which source should be integrated in the Web page. This information becomes permanently accessible. The purpose of a Digital Dashboard is to enable knowledge-workers to better focus on their key business priorities and make decisions that are more informed. In its most basic form, a Digital Dashboard solution is a Web page running within Outlook 2000.

**Collaborative solution components**

Even though collaborative solutions can have different purposes, they generally have several components in common, as shown in Figure 30-1.

![Figure 30-1: Components of a collaborative solution](image)

The following components can be used in a collaborative solution:

✦ **Folders:** Used for online discussion groups and for sharing, recording, tracking, and dissemination of constantly changing information. Collaborative solutions typically use personal folders for testing, even when the application is being assembled for public folders. Both personal and public folders are essential for creating collaborative solutions.
Views: Display sorted, grouped, or calculated data. Views determine how the information contained in an item in a folder is displayed.

Fields: Containers of information to be displayed. Fields store information about a mail message, a task, a contact, or an appointment.

Forms: A graphical interface to present view information to users.

Web pages: Display information through an HTTP protocol. Web pages are typically HyperText Markup Language (HTML) or Active Server Pages (ASP) that create and display content in a browser.

Databases: Store relational information. The collaborative solutions typically use OLE DB and Active Data Objects (ADO) to manipulate data in relational databases.

ActiveX Controls: Display information. As an example, the ActiveX Calendar control allows forms to display date information in the form of a calendar representation.

Component Object Model (COM) add-ins: Share code between different collaborative solutions. The shared code is structured as a COM object that can be used by multiple solutions.

Script: Represents the business logic of a solution. Using VBScript, you can implement the logic behind approving an expense report or denying it.

Designing a collaborative solution

There are three steps in designing a collaborative solution:

1. Develop a collaborative architecture.
2. Derive a logical system from the collaborative architecture.
3. Derive a physical design from the logical system.

Developing a collaborative architecture includes defining both the client-side architecture and the server-side architecture. On the client side, you design around the use of Outlook 2000, perhaps using Visual Studio and Office to develop the client-side solution. On the server side, the architecture includes Windows 2000 Server, Exchange 2000 Server, and SQL Server. The Windows 2000 services include the Internet Information Server (IIS), which enables you to develop Web-based collaborative solutions using Active Server Pages (ASP). You can create Web-based applications with ASP and Collaboration Data Objects (CDO) and database applications with ASP and ActiveX Data Objects (ADO).
When you design the *logical system* for a collaborative solution, you are creating a document that outlines the logical architecture. The document you create should include three key elements:

- **Replication object database**: Because people store data in different information types, and because they often want to share that information with their colleagues, the collaborative solutions you create will work best with a robust replication object database that can store many different types of information, such as Web pages, Microsoft Office documents, drawings, bitmaps, and e-mail messages. The object database usually requires replication from server to server, and from server to client, to help reduce the effect of constraints on wide area network bandwidth.

- **Support for Internet and industry standards**: Your collaborative solution must support and be able to connect to disparate networks and form a cohesive solution.

- **Use of development tools**: Your solution must include tools that can yield powerful solutions in the customer interface.

After you have composed your three key elements, you can create the *physical design*. The physical design consists of the application standards, which include error handling, naming conventions, coding style, User Interface (UI) design, client application design, server application design, database design, and any object modeling required for the collaborative solution.

### Developing Collaborative Solutions with Outlook 2000

Developing client-side collaborative solutions with Outlook 2000 involves designing forms, working with views, and programming forms.

### Designing forms

Outlook 2000 provides a number of built-in forms that can be customized for your collaborative solution, and it also provides the tool to customize these forms — the Outlook Form Designer. To start the Outlook Form Designer, open an Outlook 2000 item; on the Tools menu of the form, choose Forms and then choose “Design this form” as shown in Figure 30-2.
To set up a form, you must follow several steps. First, set up form properties. Form properties determine the form characteristics, such as its category and subcategory, so that forms can be organized. You can configure the form properties by choosing the Properties tab, as shown in Figure 30-3.

Second, add fields to the form. You can add either existing fields or standard controls. Add existing fields to a form using the Field Chooser, as shown in Figure 30-4.
You can also add standard controls to a form from the toolbox. Drag the control from the Toolbox to the form, as shown in Figure 30-5.

If you want to add ActiveX controls to a form, you must customize the Toolbox by right-clicking it and choosing Custom Controls.

Third, lay out the form using the Form Layout Tools located on the Layout menu. Layout tools include Align, Make Same Size, Size To Fit, Order, and more.

Fourth, set the control properties. There are two categories of properties: basic and advanced. Figure 30-6 displays the basic control properties, including name, caption, font, and color.
The advanced properties dialog box, shown in Figure 30-7, enables you to set properties that are specific to ActiveX controls.

Other steps involved in designing forms include binding controls to a form field, creating custom fields (including formula fields and combination fields), creating reply forms, using Office document forms (as shown in Figure 30-8), and saving and publishing a form.
Depending on how your organization is structured, the role of publishing and maintaining forms replication and revisions can fall to a number of different individuals. Often the role falls to the messaging Administrator. Who owns the role is not as important as ensuring that someone owns the role.

**Working with views**

Each folder in Outlook 2000 has a set of standard *views*, or ways of displaying information. There are five standard views: Card, Day/Week/Month, Icon, Table, and Timeline.

If you want to create a custom view for a folder, you must first select the information to be displayed by adding existing fields to a view or creating custom fields. Then, define the sort order of items in the view. You can also define the grouping of items in a view. Finally, you can format views.
Programming forms

You can enhance your Outlook 2000 collaborative solution by allowing a form and/or its controls to respond to events. The response to events is in the form of VBScript code that executes when the form or control is invoked. To create code on the Form menu, click on View Code, and then add code for the event that you want your solution to respond to. For example, if you create a text box with a name FirstName, you can create a script that checks whether the user entered a first name, and prompt if necessary, as shown in Listing 30-1

```
Listing 30-1: Script That Checks for First Name

Sub FirstName_Click()
' test that the field is not blank
If FirstName.text is null then
    MsgBox "You must enter a first name"
End if
End Sub
```

Scripting allows you to apply rules to forms based on the logical and physical design of your collaboration solution.

Developing Collaborative Solutions with Exchange Features

Developing server-side collaboration solutions involves using Collaborative Data Objects (CDO). CDO is a programming interface to the Messaging Application Programming Interface (MAPI). As an interface, CDO simplifies the task of writing server-side applications, whether they are Web-based or not.

CDO includes two libraries: the CDO library (CDOEX.DLL) and the CDO Rendering Library (CDOHTML.DLL). The CDO library enables you to create, maintain, and debug messaging code. The CDO Rendering Library enables you to display information from Exchange 2000 in a browser using HyperText Markup Language (HTML).

The CDO library, as shown in Figure 30-9, consists of objects and collections. Objects contain collections of different types. Collections contain objects of similar types.
Using CDO, you can create a session, log on, and execute any messaging logic associated with messages in a particular folder.

CDO Rendering Library objects, as shown in Figure 30-10, can be classified as top-level, child, or collection. Top-level objects are created directly by code and are not derived from any other object. A child object must be derived from another object, and a collection is a group of objects of the same type.
The RenderingApplication object provides the framework for rendering objects. It has two child objects: the ContainerRenderer and the ObjectRenderer. The ContainerRenderer is used to render the contents of a container object, such as a folder or address book; the ObjectRenderer is used to render one or more selected properties of an object, such as the subject or importance of a message.

Differences in Collaborative Solution Development Since Exchange 5.x

Exchange 5.x enabled you to create collaborative Web-based solutions; Exchange 2000 enhances these capabilities significantly.

Exchange 2000 includes a new version of Outlook Web Access, which includes the following new features: support for public folders containing contact and calendar items, named URLs to reference items (items in the Information Store are now accessed using plain text addresses, such as http://server/exchange/mailbox/inbox), and support for placing audio and video clips directly into a message.

Exchange 2000 itself also has a couple new features that affect the development of Web-based collaborative solutions: WebDAV, a redesigned Web Storage System, a workflow engine, a workflow designer, and a redesigned CDO system.
WebDAV

The WebDAV (Distributed Authoring and Versioning) protocol allows any client to have Read/Write access to the database over HTTP. WebDAV, an extension of HTTP 1.1, is used for general data access over the Internet, where the data is stored on the server but is used on the client. It supports various storage models, including file-system, semi-structured (Exchange), and relational (SQL Server). Exchange 2000 uses WebDAV to provide access to its database to any Internet client. WebDAV includes in its specification definitions of protocol elements that are used to manage file-versioning. The way WebDAV transmits data elements is by using the XML protocol format.

Web Storage System

Exchange 2000 also includes a redesigned storage system that stores both messaging and file information using Web Storage System technology. There are conceptual similarities between the new Web Storage System in Exchange 2000 and the Exchange Information Store model you may be more familiar with from previous versions of Exchange. However, there is a fundamental difference between the two. The Web Storage System holds data from other sources than Exchange. The Web Storage System can hold all the types of information the Exchange Information Store held and then some. The Web Store can hold Word, Excel, PowerPoint, Microsoft Office documents, e-mail messages, Web pages, and multimedia files. Exchange’s Information Store could be accessed through a file system or through Exchange. The Web Storage System can be accessed through any of the following: Exchange public folders, Web browser (using HTTP), file system drive mapping, and Data Access Objects (such as ADO 2.5).

There are other major differences. Earlier versions of the Exchange Information Store were monolithic. There was one priv.edb and one Pub.edb per server. The Web Storage System is different. It can be separated into multiple databases and storage groups. That affords advantages for backup, restore, and administration. It enables parts of the Web Store to be administered separately. For instance, you can have multiple Web Storage Systems providing public folders to your users; one could be backed up offline while others are still online providing access on the same server. You could not do this in earlier versions of Exchange.

The Web Storage System can also be used as both a collaboration server and an Internet publishing platform; it supports Internet protocols and formats including HTTP, WebDAV, Extensible Markup Language (XML), and Multipurpose Internet Mail Extensions (MIME).

Workflow engine

Exchange 2000 includes a feature used for developing solutions called the workflow engine. In Exchange 5.5 and earlier versions, Microsoft provided ways to create collaborative solutions using a variety of separate tools and services. They included routing objects, event services, and scriptable folders. Exchange 2000 approaches
development in a more comprehensive way by providing integrated and comprehensive services, COM objects, and a workflow designer tool. Together, they are called the workflow engine and are used for creating and maintaining collaborative solution applications.

The workflow engine helps you deal with the most common services in collaborative solution applications, such as routing, workflow-state management, and audit tracking. You can use the workflow engine, together with the workflow designer, to map out your business process. The workflow designer allows you to map out the process, while the workflow engine will actually execute the process. The engine provides access to events, routing, and audit trailing services using COM. That means you can access them from your collaborative solutions built in Visual C++ or Visual Basic.

**Workflow designer**

The workflow designer helps you create a collaborative solution quickly. Using the designer you can delineate workflow states to represent the milestones during the workflow. Once you establish the states, you connect them by actions that evoke changes in the workflow state. For instance, a workflow collaborative solution for approving an expense report might include the following states: request, approval/rejection, and payment/denial.

In a workflow application, actions trigger the movement from one state to another. In our example, you must first submit your expense report; when you do, it moves to the approval state. There it must be approved or denied. If approved, it moves to the payment state. If denied, it probably moves back to you for further clarification, receipts, and so on.

The workflow designer makes following, creating, and managing the flow through states and the actions between them easier by providing a graphical tool. Alternatively, you can use the Visual Basic scripting environment. You might want to use VB scripts to program the automation of tasks that are executed when an event or action takes place somewhere in the workflow.

These actions are likely to be one of the seven basic actions that affect your data or move within the workflow to a different state:

- **Create**: The start point for data entering the workflow
- **Delete**: The point where your data exits
- **Enter** and **Exit**: The point where data enters or exits a state and can be assigned to any state
- **Change**: The point where your data is updated
Receive: Anytime your workflow data/document receives mail

Expire: Used to clean up forgotten or abandoned requests by setting an expiration

Collaborative Data Objects

Exchange 5.5 included two distinct versions of Collaborative Data Objects (CDO). CDO 1.2.1 — the Exchange 5.5 version of CDO — was a set of objects, based on MAPI, that enabled developers to access Exchange services such as e-mail, calendaring, and contacts. A version of CDO for Windows NT (CDONTS) enabled access to the SMTP services of Microsoft Internet Information Server.

This changed in the Windows 2000/Exchange 2000 environment. CDONTS is now called CDO for Windows 2000 (CDOSYS). This COM interface includes a new library that is based on Internet standards. CDO supports many of the new services of Windows 2000, such as Network News Transfer Protocol (NNTP).

Exchange 2000 includes a new version of CDO for Exchange 2000 (CDOEX). CDOEX has moved away from a strict basis in MAPI, but it still provides COM-based access to Exchange e-mail, calendaring, and contact functionality. CDOEX does not use any MAPI code in its implementation, and it uses Internet standards and OLE DB data access to access and manipulate data in the Exchange Web Store.

Microsoft went a bit further with CDO by creating some useful additional libraries. The libraries enable you to use COM to reach into other areas of interest in Exchange 2000. For example, CDO can be used for Exchange Management by using CDOEXM.DLL to automate administrative tasks, such as adding mailboxes according to a particular prescription or maintaining a standard server configuration by modifying deviant servers. Another part of the CDO library, called CDO for Workflow objects (CDOWF.DLL), gives you access to the Exchange workflow engine. There you can use its routing for their own workflow applications.

Exchange 2000 also includes two new libraries. CDO for Exchange Management (CDOEXM.DLL) consists of a set of objects for administering Exchange 2000 Server mailboxes and servers. CDO Workflow Objects for Microsoft Exchange (CDOWF.DLL) provides a set of objects for creating workflow collaborative solutions.

FrontPage 2000

FrontPage 2000 can link to the Exchange 2000 Web Store. This means that developers and people less familiar with development can now store Web sites and build applications that reside in the Web Store. They can also use Exchange Public Folders to add collaboration and knowledge-work features to applications and Web sites.
Summary

Exchange 2000 is a collaborative solution development platform. In this chapter, you were exposed to the types of collaborative solutions and the pieces that make up most collaborative solutions. We covered views, fields, forms, Web pages, databases, ActiveX controls, COM add-ins, and scripts. You learned the basics of how to create a collaborative solution using Outlook 2000 and Exchange 2000 server features capabilities, including using forms to make it easy. Because many of you had or have Exchange solutions running on Exchange 5.5, we introduced you to some of the changes that you see in Windows 2000 and Exchange 2000.
Administrating Directory Integration with ADSI

Think of the Active Directory System Interface (ADSI) as a set of COM objects, each of which exposes several interfaces. Each interface provides different functions that enable you to access and manipulate the capabilities of directory services. ADSI objects and their interfaces can be used to manage electronic messaging capabilities that can be provided by different electronic mail providers. In this chapter we review the Active Directory System Interface, show you how to program ADSI objects, and examine the Exchange Object Model.

Exploring the Active Directory System Interface

Active Directory System Interface (ADSI) is an object model that consists of COM objects. The advantage of using ADSI is that it provides developers and Administrators (that can use the ADSI objects and interfaces in a scripting language such as VBScript) with a single set of objects and interfaces. You can write applications and scripts with ADSI that will work with any directory service that has an ADSI provider, including Windows 2000/Exchange 2000. For example, with ADSI you can write an application or script that can access Exchange 2000 objects in the Windows 2000 Active Directory Service (ADS), Lightweight Directory Access program (LDAP), Novell Directory Service (NDS), and other directories with ADSI interfaces, as long as the appropriate service providers are available.
You can use ADSI to manage the resources in a directory service, no matter which network environment contains the resource. In addition, by using ADSI you can automate common tasks, such as adding users and groups, managing printers, and setting permissions on network resources.

Each of the ADSI objects exposes its own set of interfaces. Each interface consists of several methods. When you write a client application or a script, you first have to create an instance of the ADSI object and/or an instance of the interface. Once the instance of the interface is created, you can call one of the methods of the interface and pass it parameters to retrieve or modify items in the underlying objects of the provider. As an example, when Exchange 2000 is installed, new objects are added to the Active Directory. You can manipulate these objects by writing a program or a script that uses ADSI objects to modify the Exchange 2000 objects in the Active Directory.

The reason that you can use ADSI objects in your program or script is that ADSI objects have been developed as COM objects with dual interfaces and consequently can be used either as objects in a compiled language such as Visual Basic (early binding) or as objects in an interpretive language such as VBScript (late binding). As you can see from Figure 31-1, each ADSI object can expose several interfaces. In turn, these interfaces provide methods that allow you to retrieve information about the underlying physical object and to update information for the physical object.
When you create an instance of an ADSI interface, in technical jargon you would call that creating a handle, a reference to the underlying physical object, so that you can manipulate it using the functions that the interface exposes.

All ADSI objects provide you with four basic interfaces that you can use as a programmer. The basic interfaces consist of: IADs, IADsContainer, IADsPropertyList, and IDirectoryObject. In addition, since DSI objects are COM objects with dual interface capability for both early and late binding (explained earlier) they also provide the two standard COM interfaces: IUnknown, and IDispatch.

In Table 31-1, you can find a short description of the important ADSI interfaces that you will probably use in any program or script that you create:

<table>
<thead>
<tr>
<th>ADSI Object</th>
<th>Description</th>
<th>Usage / Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>IADs</td>
<td>Interface that provide access to metadata object properties.</td>
<td>Use for object definitions in the ADSI schema. To access properties and methods that manage the object’s data in the property cache, you will also take advantage of IADs.</td>
</tr>
<tr>
<td>IADsContainer</td>
<td>Interface that controls object.</td>
<td>Use to control binding, enumeration, creation, deletion, copying, and moving.</td>
</tr>
<tr>
<td>IADsPropertyList</td>
<td>Interface that optimizes object’s data property cache management.</td>
<td>Use to improve cache utilization.</td>
</tr>
<tr>
<td>IDirectoryObject</td>
<td>Interface that bypasses object property cache to provide direct access to object properties.</td>
<td>Use for clients that do not use automation.</td>
</tr>
<tr>
<td>IADsCollection</td>
<td>Interface for managing directory element collections.</td>
<td>Use when of the same data type.</td>
</tr>
<tr>
<td>IADsGroup</td>
<td>Interface for managing the special case collections of objects.</td>
<td>Use when supporting the IADsMembers interface.</td>
</tr>
<tr>
<td>IDirectorySearch</td>
<td>Interface for supporting directory service query methods.</td>
<td>Use to accommodate directory service calls.</td>
</tr>
<tr>
<td>IADsUser</td>
<td>Interface for supporting users.</td>
<td>Use for many user support activities.</td>
</tr>
<tr>
<td>IADsComputer</td>
<td>Interface for supporting computers.</td>
<td>Use for many computer support activities.</td>
</tr>
</tbody>
</table>
Programming the Active Directory Object Model

This section shows you how to use the ADSI objects in small scripts that can perform administrative tasks that are typically associated with the management of simple computer operations. You would write the administrative programs almost the same way whether you are using Visual Basic as your development tool or writing scripts using VBScript.

ADSI fundamentals

The first step in learning how to use ADSI in programs or script is to learn three basic COM programming concepts:

- Binding
- Getting and setting properties
- Containers and children

What is binding?

In order for you to be able to invoke methods and properties in your program or script, you must first bind to an object. You can bind to different types of objects such as a computer, a domain controller, a user, or some other item in a machine’s directory structure. Once you bind to an object in your program or script, you can then read or change the properties of the object, or you can invoke any of the methods available on that particular type of object. The following line of code presents an example of how to create an ADSI binding. In this case, you are creating a variable myObject that is being bound.

```
SET myObject = GetObject("WinNT://Domain/Machine/Object,Class")
```

Using the SET statement, the object myObject is bound to a binding string using the function GetObject. The argument to GetObject is called the binding string. An ADSI binding string consists of two parts: the provider (in this case WinNT:) and the path to a computer or object (//Domain/Machine/Object,Class).

The provider indicates what type of namespace to bind to. A namespace identifies the naming convention being used for a particular binding. Objects that reside within a given namespace are identified by a unique name. For example, files stored on a PC disk drive reside in the file system namespace. The unique name of a file is based on where it is stored in the file system namespace. For example:

```
C:\public\documents\adsi\adsi_spec.doc
```
ADSI ships with the four different providers shown in Table 31-2.

<table>
<thead>
<tr>
<th>Provider</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP</td>
<td>The LDAP provider is used for communicating with server services supporting LDAP version 3, including Exchange 2000 Directory and Windows 2000 Active Directory. This is the provider for choice for applications created for standards compatibility, robust technology and a view to the future.</td>
</tr>
<tr>
<td>NDS</td>
<td>The NDS provider is employed to communicate with Novell Directory Services and NDS compatible server services.</td>
</tr>
<tr>
<td>NWCOMPAT</td>
<td>The NWCOMPAT provider provides legacy communication with Novell NetWare or NetWare compatible server services.</td>
</tr>
<tr>
<td>WinNT</td>
<td>The WINNT provider provides legacy communication with Windows NT 4.0 domain controllers (both primary and backup).</td>
</tr>
</tbody>
</table>

Be careful and specific with regard to the case sensitivity and designation you supply. For example, if you supply only the provider in the binding string (for example, `WinNT:`), ADSI will bind to the root of the provider's namespace and allow access to all objects in the enterprise. This may not be your intention. Remember that the provider name is also case-sensitive.

The path allows you to bind to a specific object within the namespace, rather than to the entire namespace. The following line of code shows how you can bind to a specific computer without specifying the domain name. If the computer you bind to is not a domain controller, the provider will bind to the local machine accounts.

```vbscript
Set myObject = GetObject("WinNT://Server01,computer")
```

In the bind statement, you can also use a path to bind to objects below the level of computer, with the following format:

```vbscript
Set myObject = GetObject("WinNT://MyDomain/dc01/MyUser,user")
```

The above statement will bind to a user `MyUser` `dc01` in the domain `MyDomain`.

You should use the optional `class specifier` (the part of the binding string after the comma) to avoid ambiguity when more than one object has the same name. In the previous line of code, you can avoid ambiguity by using the `class specifier` `user`, which ensures that if there is also a computer with the same name `MyUser` in the `dc01` domain controller, you bind to a user `MyUser` instead of to a computer.
Getting and setting properties

After you bind to an object, you can retrieve and set properties for that object. To retrieve and set object property values, you use the IADs interface. The IADs interface has six properties:

✦ **Name**: Object name
✦ **Class**: Object schema class name
✦ **GUID**: Object Globally Unique Identifier structure for unique identity
✦ **AdsPath**: Object’s unique directory service path string form
✦ **Parent**: Object’s parent container ADSPath name
✦ **Schema**: Object’s schema class ADSPath

The IADs interface also has four methods, which enable you to retrieve and set the value of a property:

✦ **Get**: Retrieves property value
✦ **Put**: Sets property value
✦ **GetInfo**: Retrieves object’s property values from the directory service, putting them into the local property cache
✦ **SetInfo**: Saves object’s property changes to the directory service

You can use the VBScript, shown in Listing 31-1, to display the standard IADs properties for a specific computer.

### Listing 31-1: VBScript for Displaying IADs Properties

```vbs
Dim myObject
Set myObject = GetObject("WinNT://MyDomain/mymachine,computer")
WScript.Echo "Name is " & myObject.Name
WScript.Echo "Class is " & myObject.Class
WScript.Echo "GUID is " & myObject.GUID
WScript.Echo "AdsPath is " & myObject.ADsPath
WScript.Echo "Parent is " & myObject.Parent
WScript.Echo "Schema is " & myObject.Schema
```

You can use Notepad as an editor and type the above VBScript into a MyFile.vbs. The .vbs designates that this is a VBScript file. Double-clicking on a .vbs file executes it.
Running the script on Listing 31-1 will produce the following output

```
Name is MYMACHINE
Class is Computer
GUID is {DA78BDC0-1q71-12gF-B343-02987C9E7553}
ADsPath is WinNT://mydomain/MYMACHINE
Parent is WinNT://mydomain
Schema is WinNT://mydomain/Schema/Computer
```

Please note that the value of the GUID line may be different when you run this example on your computer.

Although the basic ADSI properties are useful for identifying an object and where it is located in the directory hierarchy, most of the useful information about an object is stored in other properties that are specific to an object’s class. In other words, different objects have different properties that can be retrieved using the particular object’s properties and methods.

You can use the VBScript example in Listing 31-2 to return the full name of a user given the user name.

### Listing 31-2: VBScript Sample for Retrieving a Full Name

```vbnet
Dim myObject
Set myObject = GetObject("WinNT://mymachine/Administrator,user")
WScript.Echo "The full name for Administrator is " & myObject.FullName
Dim myObject
Set myObject = GetObject("WinNT://MyDomain/mymachine,computer")
WScript.Echo "Name is " & myObject.Name
WScript.Echo "Class is " & myObject.Class
WScript.Echo "GUID is " & myObject.GUID
WScript.Echo "ADsPath is " & myObject.ADsPath
WScript.Echo "Parent is " & myObject.Parent
WScript.Echo "Schema is " & myObject.Schema
```

As you can see from the fourth line of code, we are retrieving the full name for the Administrator user because in the third line, we are binding to the Administrator user. When you execute the above script, the output should look like this:

```
The full name for Administrator is Built-in account for administering the computer/domain
```
Another way to retrieve the value of a property is to use the `Get` method to read the value of a given property. You can use the VBScript in Listing 31-3 to retrieve the `FullName` property of a user object:

Listing 31-3: VBScript Sample for Reading a Property Value

```vb
Dim sFullName
Dim myObject
Set myObject = GetObject("WinNT://mymachine/Administrator,user")
sFullName = myObject.Get("FullName")
WScript.Echo "The full name for " & sUserName & " is " & sFullName
```

This example returns the same output as the previous example:

The full name for Administrator is Built-in account for administering the computer/domain

You can cache properties locally when you access ADSI objects. The advantage of caching locally is to avoid unnecessary and slow network calls. When you first bind to an object with `GetObject`, the local cache does not yet contain any of the object's properties. The first time you retrieve property values with the `object.property` syntax, or with the `Get` method, ADSI automatically retrieves all the property values for that object and places them in the local cache. All further retrieval of properties, using either syntax, will retrieve the values from the local cache.

You can, however, explicitly retrieve an object's properties to local cache by calling the `GetInfo` method. The `GetInfo` method will query the namespace and refresh the cache with the values of all the properties on the object. You can use the VBScript in Listing 31-4 to retrieve the properties into cache using `GetInfo`.

Listing 31-4: VBScript Example of GetInfo Use

```vb
Dim sFullName
Dim myObject
Set myObject = GetObject("WinNT://mymachine/Administrator,user")
MyObject.GetInfo
sFullName = myObject.Get("FullName")
WScript.Echo "The full name for " & sUserName & " is " & sFullName
```
In Listing 31-4, you first retrieve all the properties to cache with `GetInfo`, and then you can retrieve from cache any of the properties either by name or with a `Get` statement.

The process of setting a property value is similar to retrieval, but in reverse. You first set the property with the `Set` statement and then call `SetInfo`. Listing 31-5 sets the property of `FullName` for the Administrator account:

```
Listing 31-5: VBScript Example of Setting the FullName Property

Dim sFullName
Dim myObject
Set myObject = GetObject("WinNT://mymachine/Administrator,user")
myObject.GetInfo
sFullName = myObject.Get("FullName")
WScript.Echo "The full name for " & sUserName & " is " & sFullName
myObject.Put "FullName", "New Administrator name"
myObject.SetInfo
```

In Listing 31-5, you are changing the `FullName` property for the Administrator user to “New Administrator name” with a `Put` statement. This changes the `FullName` in cache. To make the change permanent, you must call `SetInfo`, as shown in the last line in the above example.

You must always call the `SetInfo` method to commit property changes to the directory service namespace. `GetInfo` overwrites the property values in the local cache. If you call `GetInfo` after making changes to properties, but before calling `SetInfo`, you will lose all the changes.

**Note**

Containers and children

In ADSI, objects can relate to each other in two ways: containers can relate to their members, and objects can relate to their children.

Typically, the root of a namespace is an object. An object can have multiple collections related to it. Different collections have different characteristics. Each collection can have multiple objects. All the objects within a collection have the same characteristics. The root is an object. The root can have different multiple collections. Each collection can have similar multiple objects.
As an example, think of the Sun star system. The root of this hierarchy is the Sun object. The Sun object is related to the Planets collection. The Planets collection has an object, Earth. The Earth object has two collections: Continents and Oceans. The Continents collection has an object, America. As you can see, the pattern continues, alternating between an object and a collection.

Examples of containers and their members include a collection of domains and the domain objects in them, a collection of computers and the computer objects in them, and a collection of users and the user objects in them.

The children of an object are all the elements one level below the object in the directory structure. An object’s children do not share the same Class, but their ADsPath attributes will be directly related. For example, the children of a domain object include the collections of users, computers, global user groups, and other collections whose position in the directory structure is directly beneath the domain.

All the Active Directory and Exchange collections expose an interface, IADsContainer, that enables us to programmatically retrieve the count of the objects within the collection and select which objects are going to be retrieved from the collection:

- **Filter:** This property allows you to select which objects will be retrieved from the collection based on the value set in this property.

- **Count:** This property allows you to retrieve the count of the number of objects in a collection. If the count is retrieved after the filter property is set, the count will reflect the subset of the collection that meets the filter selection criteria.

In addition to the two properties, the IADs Container interface exposes the following methods:

- **GetObject:** This method enables you to bind to an object or a collection with the specified AdsPath.

- **Create:** This method enables you to create a new object of a specified class in the current collection.

- **Delete:** This method enables you to remove an object of the specified class from the current collection.

- **Copyhere:** This method enables you to create an object copy with a specified ADsPath in the current container. The object being copied must be in the same directory namespace.

- **Movehere:** This method enables you to move an object with a specified ADsPath from its original location to the current container. The object being moved must be in the same directory namespace.

The following VBScript example, Listing 31-6, shows you how to bind to a collection, Administrators, and retrieve for each object in the collection its Class, Name, and Description.
Listing 31-6: **VBScript Sample to List Group Members and Description Attributes**

```vbs
Dim myGroup
Dim myUser
Set myGroup = GetObject("WinNT://mydomain/mymachine/administrators")
For Each myUser In myGroup.Members
    WScript.Echo myUser.Class & ": " & myUser.Name
    WScript.Echo myUser.Description
    WScript.Echo
Next
```

Listing 31-7 shows you how to bind to a root object and retrieve all the collections that are related to it. For each collection, you can retrieve its Class and its Name.

Listing 31-7: **VBScript Example to List Children and Attributes of a Domain**

```vbs
Dim myDomain
Dim myItem
Set myDomain = GetObject("WinNT://mydomain")
For Each myItem In myDomain
    WScript.Echo myItem.Class & ": " & myItem.Name
Next
```

Listing 31-8 shows you how to set the `Filter` property to limit the collection being returned to only computers and users in the domain:

Listing 31-8: **VBScript Example Using Filter**

```vbs
Dim myDomain
Dim myItem
Set myDomain = GetObject("WinNT://mydomain")
myDomain.Filter = Array("computer", "user")
For Each myItem In myDomain
    WScript.Echo myItem.Class & ": " & myItem.Name
Next
```
The Filter property has to be set to an array. Even if you need only one value for a filter, it must still be made into an array. Passing the schema class name by itself, without using the Array function, does not raise an error, but no filtering will be applied.

**High-level objects**

This section explains how to program the following high-level objects with ADSI: ADS Namespaces container, domains, and other high-level objects.

**ADS Namespaces container**

At the root of an Active Directory tree is the Namespaces container. The Namespaces container supports the IADsNamespaces interface. Binding to the Namespaces container enables a VBScript application to access WinNT, LDAP, NDS, or other providers without having to write a separate script for each.

Listing 31-9 shows you how to enumerate the objects in a Namespaces collection to find out what providers are available; then the code enumerates for each provider namespace object the domain collections that exist:

Listing 31-9: **VBScript Sample for Enumerating Namespace Containers**

```vbnet
Dim myADS
Dim namespace
Dim domain

Set myADS = GetObject("ADs:")

On Error Resume Next
For Each namespace In myADS
    WScript.Echo "Domains in " & namespace.Name
    For Each domain In namespace
        If Err.Number = &H800704B8 Then
            WScript.Echo "This namespace contains no domains"
        ElseIf Err.Number <> 0 Then
            WScript.Echo "Unexpected error: " & Err.Number
            WScript.Echo Err.Description
            WScript.Quit(1)
        Else
            WScript.Echo "    " & domain.Name
        End If
    Next
Next
```
In the example above, the `Set` on the fourth line binds to the Ads collection. The code beneath iterates through all the namespaces in the Ads collection. For each namespace, we retrieve domain name.

**Note**

If a namespace does not contain any domains, ADSI will return the error 800704B8 (an "Extended Error") and stop execution of the script. You can avoid this situation by adding an `On Error Resume Next` statement and then testing for the error 800704B8 explicitly. This enables the code to continue even when errors are detected.

**Domains**

If you wanted to programmatically find all the methods and properties that an object supports under a particular namespace, you can use the schema property of an ADSI domain object. In the schema property, you store the ADsPath to the object’s schema class. To reiterate: the object has a schema property in which you store the path to a schema class object. A schema class object can be thought of as metadata that contains information about the object. The information stored in a schema class object identifies whether the object is a collection, what types of objects this collection can hold and what mandatory and optional properties this object supports.

Use the script in Listing 31-10 to display the properties available in a WinNT domain called `MYDOMAIN`.

**Listing 31-10: VBScript Example of Displaying Domain Properties**

```vbnet
Dim myDomain
Dim mySchemaClass
Dim member
Set myDomain = GetObject("WinNT://MYDOMAIN")
Set mySchemaClass = GetObject(myDomain.Schema)
WScript.Echo "Properties for the " & myDomain.Name & " object"
If mySchemaClass.Container Then
    WScript.Echo myDomain.Name & " may contain the following objects:"
    For Each member In mySchemaClass.Containment
        WScript.Echo "    " & member
    Next
Else
    WScript.Echo myDomain.Name & " is not a container."
End If
```

Continued
In the above example, the fourth line of code binds to the MYDOMAIN object. The fifth line of code binds to the content of the schema property of the MYDOMAIN object. The schema property points to an object of the schema class type. Consequently, we can test the value of the property container of the schema class object to determine whether this is a collection or not.

**Other high-level objects**

ADSI includes other high-level objects in addition to domains. Some of these high-level objects are Country, Locality, Organization, and Organizational Unit.

These high-level objects are collections that serve as a mechanism to organize other objects either by their physical location or by their relationship to a particular part of an organization hierarchy.

As an example of how to use the other high-level objects, Listing 31-11 shows you how to create a new Organizational Unit and set some of its properties.

**Listing 31-11: VBScript for Creating a New OU**

```vbscript
Dim myDomain
Dim newOU
' Use the Active Directory namespace
Set myDomain = GetObject("LDAP:")
' Create the organization unit
Set newOU = myDomain.Create("OrganizationalUnit", "SalesSouth")
newOU.SetInfo
' Set the properties
newOU.Description = "Sales Division for the Western Region"
newOU.LocalityName = "Sales Offices"
```
newOU.PostalAddress = "5678 AStreet St., Anywhere, WA"
newOU.TelephoneNumber = "434-1313"
newOU.SetInfo

In the fourth line of code, you are binding to an LDAP provider. Once you bind to the provider, in the sixth line you can create an OrganizationalUnit with the name SalesSouth. As you might recall from an earlier discussion in this chapter, the changes are made to cache. Before we can use the organizational unit we just created, we must make the change permanent. That is why in the seventh line we call SetInfo. Once the change is recorded in the Active Directory, we can now set some properties of the organizational unit. Once we are finished with setting properties, we have to make what we have recorded in cache permanent by calling SetInfo again.

Groups and users

Because users and groups are such an important and integral part of any namespace provider, it is very important that you learn how to program them. This section explains how to program groups, users, and associating users with groups.

Groups

ADSI provides four methods and one property to manage groups. The four methods for managing groups are

✦ Members: Returns a collection of the group’s members
✦ IsMember: Returns True if the given ADsPath is a member of the group
✦ Add: Allows you to add an object to a group by adding an ADsPath to the group
✦ Remove: Allows you to remove an object from a group by removing an ADsPath from the group

Every group has one property, Description, that allows you to enter a string comment about the group.

To create or delete a group, you must call the appropriate methods of the group parent (Create or Delete). You can use the following VBScript sample, shown in Listing 31-12, to create a group of users on a computer called mymachine.
Listing 31-12: VBScript Sample to Create a Group

Dim Target
Dim NewGroup

' Bind to Computer object
Set Target = GetObject(WinNT://mymachine")

' Create the new group
Set NewGroup = Target.Create("group", "MyGroup")
NewGroup.SetInfo

' Set the description property of the new group.
NewGroup.Description = "This is a new user group"
NewGroup.SetInfo

Creating a group could return an error message that a group with the same name exists.

Listing 31-13 shows you how to delete the group you added in the previous example:

Listing 31-13: VBScript for Deleting a Group

Dim Target
Set Target = GetObject("WinNT://mymachine")
Target.Delete "group", "MyGroup"

You can also list all the groups in a particular domain by applying a group filter to the domain object and enumerating the domain object. You can use Listing 31-14 to list all the groups on mymachine.

Listing 31-14: VBScript Example Filtering a Group List

Dim myMachine
Dim group
Set myMachine = GetObject("WinNT://mymachine")
myMachine.Filter = Array("group")
For Each group In myMachine
    WScript.Echo group.Name
Next

You can enumerate groups using the VBScript example in Listing 31-15.

Listing 31-15: Enumerating Groups

Dim Container
Dim ContainerName
Dim Group
ContainerName = "Container_Name_To_Manage"
Set Container = GetObject("WinNT://" & ContainerName)
Container.Filter = Array("Group")
For Each Group in Container
    Response.Write Group.Name
Next

Users

As with groups, you can use ADSI to manage users programmatically. The code in Listing 31-16 sets up a new user and a few important properties for that user.

Listing 31-16: VBScript for Setting up and Deleting a User

Dim myComputer
Dim newUser
' Create the new user account
Set newUser = myComputer.Create("user", "NewAccount")
newUser.SetInfo
' Set properties in the new user account
newUser.FullName = "A New User Account"
newUser.Description = "A Happy New Account"
newUser.Password = "password"
newUser.SetInfo

Similarly, the following VBScript example deletes the user account created above.

Dim myUser
Dim userParent
In order to delete a user account, you have to use the `Delete` method on its parent.

You can rename a User Account using a VBScript Active Server Page, as shown in Listing 31-17.

### Listing 31-17: Renaming a User Account

```vbnet
dim container
Dim ContainerName
Dim OldName
Dim User
Dim newUser
Dim NewName
OldName = "Old_Account_Name"
NewName = "New_Account_Name"
ContainerName = "Target_Domain_Name"
set Container = GetObject("WinNT://" & containerName)
set User = GetObject("WinNT://" & containerName & "/" & OldName & ".user")
set newUser = Container.MoveHere(User.ADsPath, NewName)
set User = nothing
```

Finally, you can retrieve a user’s properties programmatically using the schema class object, as shown in Listing 31-18.

### Listing 31-18: VBScript Retrieving a Schema Class Object Example

```vbnet
dim myuser
Dim userclass
Dim property

set myUser = GetObject("WinNT://mymachine/NewAccount.user")
set userClass = GetObject(myUser.Schema)

wscript.echo "Mandatory properties for " & myUser.Name & ":"
for each property in userclass.MandatoryProperties
```

Associating users with groups
In this section you will learn how to associate users with groups: adding a user to a group and removing a user from a group.

Listing 31-19 shows how to add a user, NewAccount, to the Administrators group:

**Listing 31-19: VBScript Sample for Adding a New User to a Group**

```vbnet
Dim myGroup
Dim myUser
Set myGroup = GetObject("WinNT://mymachine/Administrators,group")
Set myUser = GetObject("WinNT://mymachine/NewAccount,user")
myGroup.Add(myUser.ADsPath)
```

The following VBScript example in Listing 31-20 shows how to remove a user, NewAccount, from the Administrators group.

**Listing 31-20: VBScript Sample for How to Remove a User**

```vbnet
Dim myGroup
Dim myUser
Set myGroup = GetObject("WinNT://mymachine/Administrators,group")
Set myUser = GetObject("WinNT://mymachine/NewAccount,user")
myGroup.Remove(myUser.ADsPath)
```
In both cases (adding a user to a group and removing a user from a group) no call to `SetInfo` is required.

Listing 31-21 shows you how to enumerate members of a group.

```vbnet
Dim Group
Dim GroupName
Dim GroupDomain
GroupName = "Target_Group_Name"
GroupDomain = "Target_Group_Domain"
Set Group = GetObject("WinNT://" & GroupDomain & "/" & 
GroupName & ",.group")
For Each Member in Group.Members
    Response.Write Member.Name
Next
```

Use the script shown in Listing 31-22 to query user membership in a group.

```vbnet
Dim Group
Dim GroupName
Dim GroupDomain
Dim User
Dim UserName
Dim UserDomain
GroupName = "Target_Group_Name"
GroupDomain = "Target_Group_Domain"
UserName = "Target_User_Name"
UserDomain = "Target_User_Domain"
Set User = GetObject("WinNT://" & UserDomain & "/" & 
UserName & ",.user")
Set Group = GetObject("WinNT://" & GroupDomain & "/" & 
GroupName & ",.group")
Response.Write Group.IsMember(User.ADsPath)
```
Error handling
In this section, you will learn how to trap ADSI errors and how to identify the common ADSI errors you should be concerned about.

Trapping ADSI errors
With VBScript, you can program inline error handling only. The statement `On Error Resume Next` turns on error handling in VBScript. After each statement that can cause an error, you test for errors by calling the `Err` object, as shown in Listing 31-23.

Listing 31-23: VBScript Sample Showing Testing for Errors

```vbnet
Dim myComputer
Dim newUser
' Turn on error handling
On Error Resume Next
Set myComputer = GetObject("WinNT://mymachine")
' Create the new user account
Set newUser = myComputer.Create("user", "NewAccount")
newUser.SetInfo
If Err.Number = &H800708B0 Then
    WScript.Echo "The user " & username & " exists."
End If
```

Common ADSI errors
All ADSI errors have a hexadecimal representation. Table 31-2 lists the common ADSI errors and their descriptions.

<table>
<thead>
<tr>
<th>ADSI Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;H80005000</td>
<td>An invalid ADSI pathname was passed.</td>
</tr>
<tr>
<td>&amp;H8000500D</td>
<td>The ADSI property cannot be found in the property cache.</td>
</tr>
<tr>
<td>&amp;H8000500E</td>
<td>The ADSI object already exists.</td>
</tr>
</tbody>
</table>
For a more detailed description of ADSI programming, refer to the ADSI Reference at the following URL:

What Is the Exchange Object Model?

As mentioned earlier, when you install Exchange, additional collections and objects are added to the Windows 2000 Active Directory. In this section, we will discuss what those additional Exchange 2000 objects are and how you can programmatically use ADSI to access these objects. To illustrate, when you install Exchange 2000, the following collections are added to the Active Directory: mailboxes and recipients.

When accessing Exchange 2000 objects in the Active Directory, you can use either ADSI or Active Data Objects (ADO), as shown in Figure 31-2.

![Figure 31-2: A collaborative solution can access the Active Directory through either ADSI or ADO.](image)
Active Directory

Normally, the Windows 2000 Active Directory contains Windows 2000 user account information. When Exchange 2000 is installed, the Active Directory is expanded to include Exchange configuration information and user messaging information.

Exchange configuration information is stored in a special Exchange container in the Active Directory Database.

As mentioned earlier, the Active Directory contains information about Exchange 2000 objects such as mailboxes, distribution lists, servers, topology, and system configuration information. It provides a unified, structured view of all users and resources within the Exchange 2000 environment. Use the LDAP namespace v2 and v3 (RFC2251) and Internet Domain Name Service (DNS) to access the Exchange 2000 directory. LDAP is natively implemented within the Exchange 2000 directory service. DNS is the locator service that ADS uses.

Exchange 2000 schema

The Exchange 2000 schema is a collection of descriptions, or rules, that define characteristics of Exchange objects in the directory. The schema defines available object classes in the directory, relationships between object classes, attributes of each object class, and specific characteristics of attributes and classes.

The Active Directory contains Exchange 2000 object definitions called classes. Classes define the characteristics of objects created with a particular class type. You can have many objects in the directory with the same class. Different classes can also be stored in the directory.

Each class describes a set of attributes (or fields of information) that are either mandatory or optional. The class definition includes all the data for the class attributes, including attribute name, length, data type, and so on.

Each mandatory attribute must have a value that is valid in terms of length, data type, and so on. Optional attributes do not need to contain a value. However, if the value is present, it must conform to the length and data type dictated by the object class. Attributes can contain single or multiple values depending on the class definition. For example, the Members field on a Distribution List contains multiple values; the Directory Name field contains a single value.

The full list of object classes and the definition of each class is defined in the schema. Class definitions are constructed hierarchically. Each derived class inherits all of its parents’ attributes. All directory classes are derived from a class called Top. The Exchange 2000 schema is not extensible: you cannot add new classes or attributes to the directory.

Do not confuse ADS, which is extensible, with the Exchange schema, which is not extensible.
Managing recipients

In this section we will review some of the Exchange 2000 collections and objects that are added to the Active Directory. You will learn how to create a recipient, as shown in Listing 31-24.

Listing 31-24: VBScript Sample of Custom Recipient Creation

```vbnet
Dim strDisplayname
Dim strAlias
Dim strTelephone
Dim objCont As
Dim objNewCR

strDisplayname = "James Smith"
strAlias = "jsmith"
strTelephone = "867-5309"

Set objCont = CreateObject("LDAP://Server/cn=Recipients,ou=Site,o=Org")
Set objNewCR = objCont.Create("Remote-Address", CStr("cn=" & stralias))
objNewCR.Put "cn", CStr(strdisplayname)
objNewCR.Put "uid", CStr(stralias)
objNewCR.Put "telephoneNumber", CStr(strtelephone)
objNewCR.Put "Target-Address", "SMTP:jsmith@microsoft.com"
objNewCR.SetInfo
```

In the example above, we use the CreateObject method to bind to a Recipients collection. Once you have the binding to the collection, you can use the Create method to create a recipient object. Then you can use the Put method to set the properties for the recipient object you created. In the example above, we set the \textit{cn}, \textit{uid}, \textit{telephoneNumber}, and \textit{Target-Address} properties. Then, as usual, to make the changes permanent, we call the SetInfo method.

Retrieving e-mail addresses

The script shown in Listing 31-25 enables you to retrieve the properties of an e-mail address in a Recipients collection.

Listing 31-25: VBScript Sample for Retrieving E-Mail Addresses

```vbnet
Dim objMailbox
Dim varAddrs As
Set objMailbox =
```
CreateObject("LDAP://Server/cn=Mailbox,cn=Recipients,ou=Site,o=Org")

' Retrieve and print the smtp address
Debug.Print objMailbox.Get("mail")
Debug.Print objMailbox.Get("rfc822Mailbox")

' Retrieve and print the X.400 address
Debug.Print objMailbox.Get("textencodedORaddress")

Set objMailbox = Nothing

In the third line of code in the above example, we bind to a mailbox within the Recipients collection. Using the binding variable, we can print the value of the mail and rfc822mailbox (for SMTP) and textencodedORaddress property (for X.400) attributes of the mailbox.

How to Retrieve Active Directory Information

This section explains how to use Active Directory tools to retrieve information about the Exchange Server. It also explains how to use the Active Directory Editor to retrieve and modify schema information, and the Active Directory Browser to retrieve and modify object information.

ADSIEdit

You can use the Active Directory Editor (ADSIEdit), a Microsoft Management Console snap-in, to browse the Active Directory and, in particular, the Exchange 2000 information. You can use this browser to view, change, and delete the attributes of any object in the Active Directory.

To use the Active Directory Browser, execute the following steps:

1. Log on as Administrator (if you haven’t already).
2. From the Start menu, select Run.
3. In the Open: field, type mmc and then click OK.
4. When the Microsoft Management Console (MMC) appears, select Add/Remove Snap-in from the Console menu.
5. Click the Add button.
6. From the displayed snap-in list, select the ADSIEdit Snap-in, click Add, and then click Close.
If ADSIEdit is not in the list, install the Windows 2000 Resource Kit from the Windows 2000 CD in the folder \Support\Tools.

7. Click OK to close the Add/Remove Snap-in dialog box.
8. Right-click the ADSIEdit node beneath Console Root and select the Connect to... option.
9. In the Connection dialog box, verify that the Name: field contains Domain NC, and then click OK.
10. To view the top-level containers in your domain, expand the entry for your domain tree.

The containers Builtin, Computers, Domain Controllers, and Users are part of every Active Directory tree. You will see these containers when using the Active Directory Users and Computers administrative snap-in. You will see some of the other containers, such as System, only if you view the advanced features in the Active Directory Users and Computers administrative snap-in.

To view the Exchange 2000 Active Directory information, expand the entry CN=Microsoft Exchange System Objects. The list will appear in the right pane, as shown in Figure 31-3.

![Figure 31-3: MMC snap-in with ADSIEdit for Exchange Server objects](image)
Active Directory Browser

You can use the Active Directory Browser (ADSVW) to examine the namespaces of the various ADSI providers.

To use the Active Directory Browser, you should execute the following steps:

1. Log on as Administrator (if you haven’t already).
2. Start the program ADSVW.exe.
3. In the New dialog box, select ObjectViewer and click OK to display the New object dialog box. Figure 31-4 shows the dialog box that should appear.

![Figure 31-4: Object Viewer dialog box](image)

4. To view all the ADSI providers currently installed, type ADs: in the Enter ADs path: field and then click OK. If you receive an error message, clear the Use Open Object check box and try again. You will now see a list of the ADSI providers currently installed on your computer, as shown in Figure 31-5.

![Figure 31-5: Entering the ADs path](image)

5. Double-click LDAP to open the LDAP namespace. Double-click your domain tree entry to show the top-level containers in your domain.
6. Double-click Exchange Domain Servers to display the Exchange 2000 domain server, as shown in Figure 31-6.
You can browse through other namespaces that interest you. For example, browse through the namespace until you find the Exchange Services object.

**Summary**

ADSI is relatively easy to use and can make your job as an Administrator or as a developer easier. What you have learned in this chapter — how to use ADSI COM objects to perform administrative tasks, how to use ADSI interfaces to access Exchange 2000 objects programmatically in the Active Directory, and how to use Active Directory tools to retrieve information about the Exchange Server — is somewhat applicable to other directory services, such as Novell NDS and Microsoft Meta Directory Service, but its real use is in combination with the other features of Exchange 2000 to build low-cost, high-horsepower applications. With some development talent, and perhaps more detailed books on development, you can take what you have learned in this chapter quite a bit further.
This chapter introduces you to the administrative tasks of collaborative solutions, including client-side (Outlook 2000) collaborative solutions, server-side (CDO) collaborative solutions, and Web-based collaborative solutions. The chapter explains each of the components and clarifies what you should be administrating and how to do it. Although many of these components are covered from a different administrative perspective in other sections of the book, the focus here is primarily on administrative roles directly associated with collaborative solutions.

**Administrating Outlook 2000 Collaborative Solutions**

This section explains how to create and manage team folders, use Outlook Web Access, configure information stores, and create a Digital Dashboard.

**Team folders**

A team folder is a special type of public folder, in which someone has restricted user access to members of a team. It is a cornerstone of the Web-based collaboration tools for Outlook 2000. The Team Folders Kit (TFK) is an add-in tool that bridges the gap between the public folder collaboration paradigm and the Internet paradigm. Using TFK, internal and external development teams can meet business customers’ needs for messaging-based collaboration and communication. TFK takes flexibility and collaboration features of public folders and combines them with the intuitiveness of the Web; you can create a
location for shared information that has the feel of the Web. Using the Network News Transport Protocol (NNTP) in Exchange 2000, you can also make folders accessible on the Web.

The extent of access you provide, and to whom, takes careful consideration and deft handling of permissions. With TFK you do not have to choose between constituencies who prefer the so-called Internet feel and messaging feel, or even between the so-called Internet route or public folder route. You can easily provide access to a broad constituency with both.

**Deploying the Team Folder Wizard**


TFK includes an invitation form, ActiveX controls, a Team Folders Wizard Setup file, and supporting documentation.

**Invitation form**

You can create a team folder using one of the Microsoft Outlook 2000 Team Folders Templates. You can then send a message to team members that (1) alerts them about the creation of the team folder and (2) provides a button they can click to add a shortcut for the team folder on their Outlook Bar.

Before sending an invitation message, publish the form the message is based on in the Microsoft Exchange Server Organizational Forms Library. This prevents users from receiving a warning about macros in the invitation message.

How far ahead of time you should publish the forms depends on the latency in your overall Exchange Design and WAN. You need to ensure that replication has had time to complete (if necessary) and that the form is available in the server where the client will look.

To publish the form, TFK includes a personal folder file (Invite.pst) containing the form and a utility (Vbinvite.exe). If you are a Microsoft Exchange Server Administrator, you can use the utility to publish the form automatically.

**ActiveX controls**

ActiveX is an important part of TFK. The folder home pages for the Team Folders Templates use two ActiveX controls: the Microsoft Outlook View Control and the Microsoft Outlook Permissions Control. The default is for the folder home pages to download these controls from the Microsoft Web site ([www.microsoft.com](http://www.microsoft.com)). Obviously, this model assumes your users have Internet connectivity. For organizations whose users do not have access to the Internet (or if you simply do not wish to employ this model), you can configure the folder home pages to download the controls from a server on the organization’s LAN. TFK includes .cab files that would otherwise be downloaded. It also contains the controls and utility files (Codebase.bat and Codebase.ini) that specify the location of the source files.
If you decide not to download the ActiveX controls from the Internet, you must manage updates of the .cab files and controls on your local server, so that you are running code that is as recent as you determine is appropriate. For some organizations, the ability to control updates is enough reason to manage them locally.

**Team Folders Wizard Setup**

Before a team folder can be created, the Team Folders Wizard and the Team Folders Templates must be installed on the user machines. For these installations, Microsoft provides the Team Folders Wizard Setup, a self-contained executable file (Oltfwiz.exe) that a user can run either from a network share or from an e-mail attachment.

**Supporting documentation**

TFK also contains programmer’s reference files for the View Control (Ovctl.chm) and the Permissions Control (Oltfacl.chm). They are located in the Docs folder. As with most Microsoft products, a Readme file in the installation folder contains information that developed too late to be included in the body of the guide.

**Team folder setup**

Team folder setup proceeds as follows: Team Folders Templates and supporting files are copied to a folder on the user’s hard drive. After that, the Team Folders Wizard is installed as a COM add-in for the Outlook 2000 client. The Wizard creates a folder named TFWizard in the Microsoft Office installation folder (the default for its location is C:\Program Files\Microsoft Office\Office). There the Wizard Setup creates a subfolder. The name of the subfolder will be a number representing the user's language-specific version of Office. In that subfolder the Wizard copies the Team Folders Templates.

Templates are located in their own subfolders. Each template consists of

- An Outlook personal folders (.pst) file
- One or more Web pages
- An initialization (.ini) file

The personal folders file contains one or more folders also copied to the destination public folder by the Team Folders Wizard. These folders often contain nuggets of customization. Typically, they have custom forms published in their forms libraries and often have custom fields and views defined.

Usually the Web pages provide the user interface for the team folder application. Often they will contain documentation (such as online help) to assist the user.

**Tip**

Make sure you remember to build the online help files for the users, and establish a culture in which users consult online help first. If not, the support phone will ring off the hook with easily answered questions.
The initialization file contains critical information including template structure, folder home page names and locations, and strings in the Web pages that are to be replaced for localization. The Wizard uses these settings when copying the contents of the personal folders file and the Web pages to their respective locations.

The TFWizard folder also contains dynamic-link library (.dll) files (the ones that make up the Team Folders Wizard itself). The Wizard Setup installs the Wizard as an Outlook COM add-in by creating entries in the system registry. Those entries instruct Outlook where to find the files and how to load them. Once they are in place, the next time the user starts Outlook a command is added to the File menu for running the Wizard.

**Using the Team Folders Wizard**

Using the Team Folders Wizard simplifies the creation of team folders. It walks the user through the process of selecting the type (template) of team folder, naming the team folder, specifying where the components of the team folder are to be located on the network, and selecting the initial list of users. After selections are made, the Wizard does most of the rest. It creates the team folder in the location specified and configures it according to the choices provided. Finally, the Wizard displays the team folder Administration page so that you can finish setting up the team folder and send an invitation message to the team folder’s users. Figure 32-1 shows the welcome screen of the Wizard.

![Figure 32-1: Welcome screen of the Outlook Team Folders Wizard](Image)

The rest of this section describes the selections you can make using the Wizard.

**Selecting the team folder type**

As shown in Figure 32-2, the Team Folders Wizard allows you to select from six templates of team folders. The templates are included with the Wizard and designed to cover the needs of most types of basic collaborative solutions. As you might expect, you can create additional custom templates and add them to the list of templates to satisfy more customized requirements.
Naming the team folder
As shown in Figure 32-3, if you are not satisfied with the default name the system provides, you can provide your own name for the team folder you are creating. The one that the Wizard creates consists of the word My with the name of the selected template appended to it. The name is used for the name of the public folder that the Wizard creates for the team folder components. If the name is already used by an existing public folder, you will be asked to change the name.

Specifying team folder locations
As shown in Figure 32-4, the next step in creating a team folder is to specify two locations where the team folder files will be created. Choose from a Microsoft Exchange Server public folder or a Web (HTTP), FTP, or file server share.
Figure 32-4: Selecting team folder location

Selecting users
As shown in Figure 32-5, the final step in creating a team folder is specifying, from the Global Address List (GAL), team users who will be allowed to use the team folder. Users selected from the GAL are granted Author rights as a default, and the user who runs the Wizard is granted Owner rights. Default and Anonymous users (shown as Everyone in the Wizard) are granted Reviewer rights.

Figure 32-5: Selecting team folder users

Tip
Be sure to assign the right permission to the right people and groups. When in doubt, supply more limited permissions until requested to open them up, but do not be so restrictive that the folder utility is diminished.
Creating and configuring the team folder

Once the Wizard has your selections, it copies the template’s Web files to a temporary directory. It uses your selections stored in the initialization file to modify and customize the files. The Wizard copies the Web files to the server location you specified. Next, the Wizard makes a copy of the template’s personal folders file in a temporary directory. There it performs some preprocessing. When the preprocessing is complete, the Wizard again copies the contents of the personal folders file, this time to the specified public folder. It then sets the permissions as specified, creates a contact item for each user, and sets the addresses of the folder home pages. Just before finishing, the Wizard opens an Outlook Explorer window displaying the team folder Administration page, which enables you to put the finishing touches on your effort: specifying the text that you wish to appear in the Welcome and Top News sections of the home page, providing links to team information, and defining areas for tasks. (The Administration page itself is structured somewhat like a Wizard, leading you through the steps to help ensure none are missed.) The last step is for you to send other users an invitation message. The message contains a button that, when clicked, creates a link to the team folder on the user’s Outlook Bar and makes participation easy for the other team members.

Customizing team folder templates

You can customize any of the sample six templates of team folders to meet your specific needs. Customization of a team folder template is not difficult. It involves customizing the home page, creating a custom template, and adding a custom extension to the Wizard.

Customizing the home page

The Team Folder home page uses Dynamic Hypertext Transfer Markup Language (DHTML). With a rudimentary knowledge of DHTML, you can change the appearance of the Team Folders home page, remove allowed elements, and add support for an additional subfolder in a team folder application. You can further customize by making changes to the Outlook View Control, altering the name of a folder, dictating a page’s purpose, modifying the navigation bar, customizing view tabs, and adding a folder.

Creating a custom template

The easiest way to create a custom template is to modify the files that go with the sample template located in the Sample folder in the TFK installation folder (C:\TFKIT). The files you should focus on are the Outlook personal folders file (.pst), folder home page, and template initialization file.

After you make the changes to the template, use the Microsoft Office 2000 Developer’s Package and Deployment Wizard to install and register your template.
Adding a custom extension to the Wizard

If you want users to use a modified version of the Wizard, customized for your needs, you can develop a Team Folders Wizard extension that can accept input from the user and modify the template based on that input. A Wizard extension is an ActiveX dynamic-link library (.dll) registered as an extension. It implements a specific set of Component Object Model (COM) interfaces.

To help you with this, the TFK contains a sample template, located in the Sample folder, that includes a Wizard extension. Use the source files for this sample extension to create your own extension and to learn more about extensions.

Administering Outlook Web Access

Exchange 2000 includes an enhanced Outlook Web Access (OWA) that enables users to access e-mail, personal calendars, group scheduling, and your collaboration applications through a Web browser. The degree of functionality depends on which advanced features are supported by the browser; Internet Explorer 5.0 supports the most.

See Chapter 15 for more information.

OWA and Internet Explorer 5.0 are often the best choice for a front end to applications that include kiosks or other public areas, roaming users, and users behind firewalls with limited ports open.

Exchange 2000 includes a new version of OWA that overcomes performance and scalability difficulties of previous versions. OWA for Exchange 2000 includes four new features:

✦ Support for Internet Explorer 5.0 that enhances performance
✦ Support for public folders containing contact and calendar items
✦ Access to the information store using URL addressing (http://server/exchange/mailbox/inbox)
✦ Adding audio and video clips directly into a message

Managing Outlook Web Access in Exchange 2000 is like managing any other Hypertext Transfer Protocol (HTTP) virtual servers in Internet Information Services (IIS).

Administration tasks that you can perform with OWA include start, pause, and stop the server; enable OWA for a user; and configure a front-end server. You can configure a front-end back-end topology. If you choose to configure a server to be a front-end server, you are dedicating that server to receive requests from clients and relay the request to the appropriate Exchange 2000 back-end server.
After identifying a server as a front-end server, examine its Exchange configuration. Removing unnecessary components improves performance and usually enables the front-end server to relay client requests more efficiently, making applications appear snappier to the users.

**Information store administration**

Information stores are Exchange 2000 databases. Those databases, and the service that manages them, are crucial to almost any application developed for Exchange. Exchange 2000 includes a new feature that supports multiple databases contained within a storage group. A storage group includes one to five databases, and a single set of transaction log files for those databases. You are limited to a maximum of four storage groups on one server. If each storage group contains the maximum of five databases, you are limited to a maximum of 20 databases on one server.

In general, you should create multiple databases to store your folders. Using multiple databases and deftly mapping your software to your hardware may enable you to increase the number of simultaneous users on a server and lessen the risk of a damaged database. In addition, with multiple databases the size of each database is decreased, and data recovery is much faster.

See Chapters 8 and 19 for more information on this topic.

All storage groups are managed by the Information Store service. Each storage group contains two kinds of databases: *mailbox stores* and *public folder stores*.

In most collaborative solutions, someone takes ownership of the Information Store architecture. This person spends quality time delineating how the solution will be mapped to storage units, taking care to understand the implications for access speed, speed of restoration in case of failure, and the effects of latency.

**Mailbox stores**

Mailboxes are the delivery location for all incoming mail messages for a designated owner. A mailbox can contain received messages, message attachments, folders, documents, and other files.

If the user has a PST or OST, the mail may stay in the user’s mailbox for a very brief or long period, but it will always go through the mailbox.

In Exchange 2000, information in a user’s mailbox is stored in a mailbox store. Each mailbox store must be associated with a public folder store that you specify; the default is the server that the mailbox is on but can be altered on a server property page. The public folder store installed by default on each server contains the MAPI public folder hierarchy, which is the All Public Folders branch of the Public Folders tree. Users get their view of the hierarchy from their public folder server. You are allowed only one MAPI public folder hierarchy in your Exchange organization and
other non-MAPI public folder hierarchies. The MAPI hierarchy is associated with only one default public folder store.

**Public folder stores**

Exchange 2000 supports multiple public folder hierarchies; each hierarchy (or tree) is stored in a public folder store. The default server installation includes one public folder store that contains one public folder hierarchy. You are limited to one public folder hierarchy in a public folder store.

Before you create a public folder store, you must first create a public folder hierarchy to store in it. If you try to create a public folder store without an available public folder hierarchy, you will see the following error message:

All the public folder trees already have an associated public store on the server. You need to create a new public folder tree before creating this new public folder store.

The default public folder store contains the All Public Folders hierarchy. This store must be associated with every mailbox store on a server to ensure that the All Public Folders hierarchy is displayed in IMAP and MAPI mail clients.

Any additional hierarchies that you create are accessible from standard Windows applications in which the folders in the hierarchies are mapped as network drives using IFS, WebDAV, and NNTP clients. Use the extra hierarchies as file repositories for departments, groups, or projects.

**Web Storage System**

One of the greatest improvements in Exchange 2000 is its ability to integrate into the Internet paradigm seamlessly. A large part of this Internet integration is the enhancement to the message storage technology called the **Web Storage System**. The Web Storage System allows you to organize your information for access with a Web browser to any item in a public folder hierarchy such as messages, sub-folders, appointments, meetings, and so on. Access to any item in the Web Storage System is made by using any one of the following protocols and formats: HTTP, WebDAV, Extensible Markup Language (XML), or Multipurpose Internet Mail Extensions (MIME).

In addition to direct access through the Web Storage System, you can use Web Store Forms to access items in any public folder. Web Store Forms enable users to enter and view information. They associate a form with a particular item in an Exchange 2000 public folder; then, whenever an item is accessed with a Web browser, the item is displayed using the associated Web Store Form. Every item has a default Web Store Form associated with it to allow it to be displayed in a Web browser.

A Web Store Form is actually an Active Server Page (ASP) that includes logic to display the content of an item in a particular format.
You can make an ASP page adaptive to different languages, different browsers, and the workflow state of the item (for example, an initial entry or response) in two ways. First, you can include conditional logic in each ASP page to test for each of the different conditions, and display the item accordingly. Second, you can use a different form for each combination of conditions. This second solution, the one adopted by Exchange 2000, requires more forms but is much more efficient in displaying the contents of an item. Once you choose a form, there is no additional logic to be executed.

Web Store Forms uses a mapping table, as shown in Figure 32-6, to determine which form to use when you request that an item in an Exchange 2000 public folder be displayed. Based on conditions such as language, Web browser being used, and the workflow state of the item, the appropriate form will be selected to display your item.

![Figure 32-6: Web Store Form operation](image)

The main advantage of Web Store Forms is speed. They have no conditional code and therefore display any item you choose quickly. And since they are specifically tailored to a particular set of conditions, they can be cached for even better speed in retrieval.

**Administrating the organization forms library**

For each item in an Exchange 2000 public folder, you need a set of forms that matches the combination of conditions for the item and its presentation.

To make forms easier to manage, Exchange 2000 includes an organization forms library, a repository of all the forms that can be used to access the items in Exchange 2000 public folders.

*Note* You can use multiple organization forms libraries, one for each language you use. Each organization forms library can have only one language assigned to it.
In Exchange 2000 the organization forms library is a public folder. By making the forms library a public folder, Exchange 2000 allows you to change the properties of the folder and consequently of the forms library. In addition to the language property, you can also set other properties, such as access permission, and assign an e-mail address for the forms library itself.

**Digital Dashboard**

Outlook 2000 is a popular messaging system client for Exchange 2000. Exchange 2000 helps Outlook 2000 become a knowledge-management system for day-to-day information needs. It is axiomatic that knowledge-workers typically look for a tool that allows them to consolidate all the different types of information they use daily into a single view. The type of information that people want to consolidate and view depends on the nature of their work and organizational responsibilities; often it includes personal, team-organizing, and even some external information.

Many believe the solution to knowledge-management needs is the Digital Dashboard. The Digital Dashboard comes packaged as a resource kit (Digital Dashboard Resource Kit 2.01) that you can download for free from the Microsoft Web site. The use of the Digital Dashboard requires that you also be using Internet Explorer 5.0 or later as your default browser.

**Capabilities of Digital Dashboard solutions**

The Digital Dashboard allows you to extend the capabilities of Outlook 2000 to include additional information that makes up your personal knowledge-management system. To do that, the Digital Dashboard allows you to define Web parts. A *Web part* is the definition or information that you want to display as part of your Outlook 2000 view. You can create your own Web part, or you can import a Web part from libraries on the Internet or on corporate intranets.

In order to facilitate your use of a Digital Dashboard solution, the Digital Dashboard Resource Kit includes a library of pre-built Web parts. These pre-defined Web parts include samples for displaying SQL Server data and the contents of files.

An easy way to get familiar is to customize all of the samples to meet your personal information needs. You can also, at any time, change the placement of a Web part in your Outlook 2000 view as well as which Web part is displayed. As the nature of your work changes, you can modify your personal Digital Dashboard to reflect the changes in your information needs.

You can include data from SQL Server databases, file systems, legacy systems Exchange 2000 folders, and many more sources of data in your personal Digital Dashboard. The information you display as a Web part can either be dynamic (up-to-the-minute) or static (in the form of a snapshot taken at regular intervals).

In addition, you take advantage of the use of filters with Web parts to identify conditions that you want to track. For example, you define a Web part to track your favorite stock. You can add a filter to this Web part that alerts you whenever the
stock falls to a certain level. Assuming there is little latency, you may have time to buy or sell (or take an aspirin). In general, the availability of filters associated with Web parts turns the Digital Dashboard from a passive knowledge-management system to an active one. Your messaging system puts information in front of you, enabling you to take action when potential problems are detected or opportunities present themselves.

For collaboration, Web parts can also be created to view items in an Exchange 2000 public folder for team members. Members can have instant access in their Outlook 2000 view to any information being generated by any of the team colleagues. Such instant access to team information can keep users in the loop at all times.

Finally, Web parts also enable users to interact with the information being displayed. The Digital Dashboard allows them to use such tools as the PivotTable component of Excel 2000, which can manipulate information being retrieved from a SQL Server database or other sources as a Web part. NetMeeting, a peer-to-peer conferencing software that enables users to conduct online meetings on the Internet, including chats and sharing of applications, can also be used as a Web part. Users can also include a media player with streaming video and audio content.

Streaming video, and even sound, takes transmission-system bandwidth.

Imagine media content in a Digital Dashboard that includes a presentation on a technical topic that you want to learn about. In a sense, your Digital Dashboard becomes a classroom for your personal trainer.

**Integrating Web content**

The Digital Dashboard is made up of Web parts—that is, reusable components that can contain any kind of Web-based information. Web parts are relatively easy to build. In fact, the Digital Dashboard was designed to allow end-users to create their own simple Web parts. For more sophisticated development, the Digital Dashboard includes a Web Part Builder for more complex (that is, beyond the capability of end-users) Web parts, such as those that integrate other data sources: Office 2000, Exchange 2000, SQL Server, and various types of information from the Internet.

**Digital Dashboard samples**

The Digital Dashboard Resource Kit includes several Web part samples that can be used to create personal digital dashboards. Two of the samples are the Windows 2000 File System Sample Dashboard and the SQL Server Sample Digital Dashboard.

**The Windows 2000 File System Sample Dashboard**

The Windows 2000 File System Sample Dashboard allows access to the contents of any file on your Windows 2000 file system as a Web part. It installs additional components in the folders for you to access as Web parts and changes the properties of the folders to include WebDAV access properties.
You can install the Windows 2000 File System Sample Dashboard only on a computer running Windows 2000 Server and IIS 5.0.

As shown in Figure 32-7, you can install the Windows 2000 File System Sample Dashboard from the Building Digital Dashboards page of the Digital Dashboard Resource Kit.

**Note**

You can install the Windows 2000 File System Sample Dashboard only on a computer running Windows 2000 Server and IIS 5.0.

As shown in Figure 32-7, you can install the Windows 2000 File System Sample Dashboard from the Building Digital Dashboards page of the Digital Dashboard Resource Kit.

During installation, the Wizard will prompt you for the name of the folder or virtual root that will hold your users’ digital dashboards. The default virtual root is Dashboards. The Wizard will also prompt you for the name of the virtual Web root that contains the Dashboard factory. The default virtual Web root is Factory.

**The SQL Server Sample Digital Dashboard**

The SQL Server Sample Digital Dashboard stores Web parts and digital dashboards in a SQL Server database. This sample dashboard includes a special IIS extension supporting the HTTP 1.1 and WebDAV methods that handle HTTP requests for dashboard contents compatible with Exchange Server and Microsoft Office.
Before you install the SQL Server Sample Digital Dashboard, create an empty SQL database to store your dashboard resources. During installation, Setup adds tables and stored procedures, and triggers to the database you specify.

You must install the SQL Server Sample Digital Dashboard on a computer running SQL Server version 7.0 or later, Windows 2000 Server, and IIS 5.0. The SQL Server Web part catalog server, the catalog database, SQL Server, and IIS must reside on the same computer.

During installation, the Wizard will prompt you for the following information: the name of the SQL Server and an authentication mode, the name of the Web root that users will use to attach to a dashboard, the name of the Web root that will be used to access the catalog, and the name of the database.

**Creating a new Digital Dashboard**

You can download the Digital Dashboard Resource Kit from the Microsoft Web site: [www.microsoft.com/business/DigitalDashboard/](http://www.microsoft.com/business/DigitalDashboard/). After you install it and install a sample digital dashboard, you will see the Welcome dashboard and an Administration dashboard, as shown in Figure 32-8. You can use the Administration dashboard to create a new digital dashboard application and set its properties. You can add Web parts to your digital dashboard by downloading existing Web parts or building new ones.

![Welcome screen for a Digital Dashboard](image)

**Figure 32-8: Welcome screen for a Digital Dashboard**

To create a new Digital Dashboard application from the Administration dashboard as shown in Figure 32-8, choose New. In Dashboard Properties, as shown in Figure 32-9, specify some or all of the following properties: the name for the new dashboard, the title for the new dashboard, and the description of the dashboard.
After you create a dashboard, you can tailor its functionality by adding Web parts, as shown at the bottom of Figure 32-9.

You can add Web parts to a digital dashboard in four different ways:

- End-users can use the dashboard’s built-in customization pages to create relatively simple Web parts.
- Administrators can take advantage of the built-in customization pages of the Administration dashboard with some ease.
- End-users and Administrators can use the dashboard Web part Catalog Listing page to import existing Web parts from a corporate catalog.
- Developers can use the Web Part Builder to create a complex Web part. That component is a Microsoft Visual InterDev add-in included in the Digital Dashboard Resource Kit.

**Customizing a Digital Dashboard**

After you create a dashboard application, you can customize it by modifying its properties. You do not have to define all properties for a given dashboard, because properties that you do not define will use default values.
Integrating a Digital Dashboard with Outlook

If you run digital dashboards as folder home pages in Outlook, users can access their dashboards from within their Outlook environment. Running a digital dashboard within Outlook provides offline capabilities and allows developers to leverage the Outlook object model to create more sophisticated collaborative dashboards.

Running the Digital Dashboard samples in Outlook Today

Outlook Today hosts a special version of Internet Explorer with one drawback: Outlook Today has a smaller feature set than Outlook Folder home pages. You can run the Digital Dashboard samples in Outlook Today, but you must first change an entry in the Windows registry that enables customization of the dashboard. Alternatively, you can use the full version of Internet Explorer to display Outlook Today by adding the following two settings to the registry:

```
[HKEY_CURRENT_USER\Software\Policies\Microsoft\Office\9.0\Outlook\WebView\mailbox]
"url"=<http://digidash/home.htm>  
(Type the location of your digital dashboard here.)
"navigation"="yes"
```

Administrating Collaborative Data Object Solutions

Exchange 2000 introduces Web Storage System event sinks. You can use event sinks to customize how Exchange responds to events. Event sinks are associated with store events.

Store events

Store events overcome the two main limitations of the Exchange 5.5 scripting model. The first limitation is that events are asynchronous; other applications may modify or delete a message before you get a chance to process the message in a script. The second limitation is that the scripting model doesn’t scale well for high-volume applications that would like to scan every incoming or outgoing message.

Store events, introduced in Exchange 2000, provide the following features:

- They fire when an item or a folder is saved, moved, or deleted within the store.
- They are scoped to a particular folder.
- They run synchronously or asynchronously.
They pass OLE DB interfaces.

Event sinks associated with them can be written in C++, Visual Basic, or VBScript.

The store event architecture allows event sinks to run in its own process, as shown in Figure 32-10. Each event sink is a free, threaded process, with the limitation that an event sink must run on the same machine as the store.

![Figure 32-10: Store event and event sink](image)

The store event architecture supports synchronous and asynchronous events as well as system events. You can handle synchronous events before an item is committed to the store; in other words, the sink process can occur within the context of a transaction. There are three important features of Exchange 2000 Synchronous Store Events that you should consider when using event sinks:

- Synchronous events fire twice: once during the Begin phase of a transaction, and again during the Commit/Abort phase of a transaction.
- They are read-committed.
- They are fired in the order specified in the event registration.

Exchange 2000 also supports asynchronous events, which are similar to the functionality provided by the Exchange 5.5 scripting model. They occur after the item has been committed to the store, firing only after the item has been saved or deleted. They are not fired in any particular order, and they are guaranteed delivery. Moving a message from one folder to another causes a Save event, followed by a Delete event, to be fired, while copying a message to another folder causes only a Save event to be fired.

The several events available are classified into three categories: synchronous, asynchronous, and system. The synchronous events include OnSyncSave and OnSyncDelete. The asynchronous events include OnSave and OnDelete. The system events include OnTimer, OnMDBStartup, and OnMDBShutdown.
Event sinks

Event sinks are the glue that holds an Exchange 2000 collaborative solution together. An event sink is the application — the script or Dynamic Link Library (DLL) — that runs when an event fires. You can write event sinks in C++, Visual Basic, or VBScript. Event sinks can handle document workflow, scan for viruses, and prevent large attachment downloads.

With a bit of creativity, you can address many valuable administrative tasks or business issues using event sinks. Suppose you want your firewall to stop most access, but you still want to get a list of active users. By using an event sink that returns a list of active users only when some specific characters are found in an e-mail in some specific place, you can achieve both security and access.

You can register an event sink for an event by saving an item with a content class of urn:content-class:storeeventreg in the folder in which the event should fire. The registration items are hidden by the event system and move with the folder. Several registration parameters are associated with an event sink: Method, SinkClass, Scope, MatchScope, Criteria, Priority, StartTime, StopTime, Interval, and Enabled.

You must consider two security issues when thinking about events: who can install event sinks and what the event sink can access. Only users who have access to the server can install an event sink. These users must be folder owners, since only a folder owner can create an item with a content class of urn:content-class:storeeventreg. In addition, in the event sink code you can implement ICreateRegistration to validate a registration request. When an asynchronous event is called, it is passed a URL to the item. When a synchronous event is called, it is passed the item and a session. The session contains the exchange logon of the user that caused the event.

When an event is fired, such as when an item is read, written, deleted, moved or copied, in the code for the event, you can access anything that the exchange logon can access; this includes creating a new session using the current security credentials of the event sink.

Managing security

Microsoft Exchange 2000 Server security features provide control over who can access Exchange objects in your organization and administrative groups and who can connect to your mail system. For your applications to have the right levels of security, it is crucial to set access to objects correctly.

As an Exchange Administrator, you can use Windows 2000 security options to protect your Exchange system and your Exchange applications. Windows 2000 Server offers security features such as user accounts, group policies, security and authentication protocols, and security logs. Windows 2000 also contains a set of tools to simplify configuring and auditing security.
To manage folder and message security, Exchange 2000 provides an Exchange Administration Delegation Wizard, a tool that simplifies delegating permissions to Exchange Administrators. When you start Exchange Administration Delegation Wizard, it prompts for users and groups to which you want to apply the administrative permissions. You can delegate administrative permissions at the organization level in System Manager, or at an administrative group level. The scope of permissions you set is determined by the place from which you launch the Wizard. If you launch the Wizard from the organization level, the groups or users that you specify will have administrative permissions at the organizational level. You can also require different levels of user authentication before granting access to your system. Authentication levels can range from no requirements, to requiring a password, to requiring a security certificate. You can further block designated users and computers from accessing your Exchange system. You can audit the messages sent to your Exchange mail system by logging the client IP addresses, domain names, dates and times of messages, or message sizes.

For more information on Windows 2000 and Exchange security architecture and components, see Chapters 6 and 22.

Summary

This chapter explains the components and some of the common tasks of creating and administrating collaborative solutions using Exchange 2000—team folders, OWA, information stores, Digital Dashboards, and event sinks. You now have a framework to use the development features to automate or extend your reach in some of the everyday administrative tasks. This framework allows you to build solutions that take all the advantages of both the Internet and the underlying store and forward messaging system. All this is done with Exchange 2000 and the rest of Microsoft BackOffice and Office—the same components that are used to develop full-featured applications, as covered in other books on Exchange development.
What’s on the CD-ROM

This appendix provides you with information on the contents of the CD-ROM that accompanies this book.

What’s on the CD

The CD-ROM contains source code examples, applications, and an electronic version of the book. Following is a summary of the contents of the CD-ROM, arranged by category.

Source code

Every program in any listing in the book is on the CD in the folder named Source Code.

Applications

The following applications are on the CD-ROM:

- Commercial version of Microsoft Internet Explorer
- Freeware version of HyperSnap DX
- Freeware version of Eudora Email
- Shareware version of WinZip
- Evaluation version of Acrobat Reader
- Commercial version of Netscape Communicator
Browsers
A browser is the client software you use to access files on the Internet or to read local HTML files.


✦ Netscape Communicator: A Web browser for Windows 9x or later; a version is also available for Linux. Freeware. For more information, see http://www.netscape.com.

Third-Party Information
Exchange was built as a development platform both for you and for other messaging industry manufacturers. Many companies have jumped at the opportunity to develop for Exchange, and many more will while this book is in press or on the shelf. This CD contains a sampling of information from various companies whose products expand and enhance Exchange 2000 messaging environments. Some of them are mentioned below. For more vendors or more current information on these vendors, consult the Web, industry conferences, newsgroups, and industry press. The following list is far from exhaustive, but is provided to give you an idea of what some vendors have:

✦ FAX & mobile connectivity: Fenestrae B.V. (Headquarters), Loire 128-130, P.O. Box 77, NL-2260 AB Leidschendam, Netherlands; phone (31) 70-3015100 (31) 70-3015151; Web site http://www.fenestrae.com

✦ Reliable storage: EMC Corporation, Hopkinton, MA 01748-9103; phone 508-435-1000; Web site http://www.emc.com

✦ Storage management: CommVault Sales & Technical Support, 2 Crescent Pl., P.O. Box 900, Oceanport, NJ 07757; attn.: David Edelstein; phone 732-870-4104; fax 732-870-4524; e-mail headquarters@commvault.com; Web site http://www.commvault.com


✦ Wireless messaging: Research In Motion, 295 Phillip St., Waterloo, Ontario N2L 3W8, Canada; phone 519-888-7465; fax 519-888-7884; Web site http://www.rim.net
Electronic version of Exchange 2000 Server Administrator’s Bible

The complete (and searchable) text of this book is on the CD-ROM in Adobe’s Portable Document Format (PDF), readable with the Adobe Acrobat Reader (also included). For more information on Adobe Acrobat Reader, see http://www.adobe.com.

URLs

Here are some other URLs of potential interest to you. Some were mentioned in the book; others are mentioned only here. URLs are often changed by the owners; as this book went to press, all were good URLs.

<table>
<thead>
<tr>
<th>URL</th>
<th>Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.ietf.org/rfc.html">http://www.ietf.org/rfc.html</a></td>
<td>RFCs</td>
</tr>
<tr>
<td><a href="http://www.microsoft.com/exchange">http://www.microsoft.com/exchange</a></td>
<td>Microsoft Exchange home page</td>
</tr>
<tr>
<td><a href="http://www.microsoft.com/isn">http://www.microsoft.com/isn</a></td>
<td>Microsoft’s Internet Service Network home page</td>
</tr>
<tr>
<td><a href="http://www.msexchange.org">http://www.msexchange.org</a></td>
<td>FAQs, third-party tools and products</td>
</tr>
<tr>
<td><a href="http://www.pcwebopedia.com">http://www.pcwebopedia.com</a></td>
<td>Definitions</td>
</tr>
<tr>
<td><a href="http://www.exchangeplus.com">http://www.exchangeplus.com</a></td>
<td>Information resource</td>
</tr>
<tr>
<td><a href="http://www.exchangefaq.com">http://www.exchangefaq.com</a></td>
<td>Information resource</td>
</tr>
<tr>
<td><a href="http://www.swynk.com/exchange">http://www.swynk.com/exchange</a></td>
<td>Information resource</td>
</tr>
<tr>
<td><a href="http://www.ntfaq.com">http://www.ntfaq.com</a></td>
<td>Information resource</td>
</tr>
<tr>
<td><a href="http://www.compaq.com">http://www.compaq.com</a></td>
<td>Compaq home page</td>
</tr>
<tr>
<td><a href="http://www.emc.com">http://www.emc.com</a></td>
<td>Storage hardware manufacturer</td>
</tr>
<tr>
<td><a href="http://www.storcase.com">http://www.storcase.com</a></td>
<td>For information on Kingston DataExpress removable drive carriers</td>
</tr>
<tr>
<td><a href="http://www.slipstick.com">http://www.slipstick.com</a></td>
<td>Slipstick Exchange information site</td>
</tr>
<tr>
<td><a href="http://www.win2000mag.net/Channels/Exchange/">http://www.win2000mag.net/Channels/Exchange/</a></td>
<td>Information resource</td>
</tr>
<tr>
<td><a href="http://www.cdolive.com/exchange2000.htm">http://www.cdolive.com/exchange2000.htm</a></td>
<td>Information resource; CDO Objects for Exchange</td>
</tr>
</tbody>
</table>
### Table A-1 (continued)

<table>
<thead>
<tr>
<th>URL</th>
<th>Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.rim.net/products/handhelds/index.shtml">http://www.rim.net/products/handhelds/index.shtml</a></td>
<td>Research in Motion Wireless home page</td>
</tr>
<tr>
<td><a href="http://www.hyperionics.com">http://www.hyperionics.com</a></td>
<td>Hyperionic (HyperSnap creator) home page</td>
</tr>
<tr>
<td><a href="http://www.whatis.com">http://www.whatis.com</a></td>
<td>Definitions</td>
</tr>
<tr>
<td><a href="http://www.tuxedo.org/~esr/jargon/">http://www.tuxedo.org/~esr/jargon/</a></td>
<td>Humorous definitions</td>
</tr>
<tr>
<td><a href="http://www.dell.com">http://www.dell.com</a></td>
<td>Dell home page</td>
</tr>
<tr>
<td><a href="http://www.ics.uci.edu/pub/ietf/webdav">http://www.ics.uci.edu/pub/ietf/webdav</a></td>
<td>WebDAV specification</td>
</tr>
<tr>
<td><a href="http://www.washington.edu/pine">www.washington.edu/pine</a></td>
<td>Pine information</td>
</tr>
<tr>
<td><a href="http://www.Microsoft.com/Exchange">http://www.Microsoft.com/Exchange</a></td>
<td>Microsoft Exchange Site</td>
</tr>
<tr>
<td><a href="http://support.microsoft.com/servicedeskstechnet/">http://support.microsoft.com/servicedeskstechnet/</a></td>
<td>Microsoft Technet</td>
</tr>
<tr>
<td><a href="http://www.microsoft.com/exchange/techinfo/deploymigrate.htm">http://www.microsoft.com/exchange/techinfo/deploymigrate.htm</a></td>
<td>Microsoft Exchange Migration &amp; Deployment</td>
</tr>
<tr>
<td><a href="http://www.microsoft.com/hcl">http://www.microsoft.com/hcl</a></td>
<td>Microsoft Hardware Compatibility List</td>
</tr>
<tr>
<td><a href="http://support.microsoft.com/support/kb/articles/q262/0/68.asp">http://support.microsoft.com/support/kb/articles/q262/0/68.asp</a></td>
<td>Article on Exchange 2000 Hardware Requirements</td>
</tr>
</tbody>
</table>
Newsgroups

There are many Internet newsgroups for Exchange. Here is a list of a few of our favorites:


Resource Kits

If you subscribe to Microsoft TechNet, you will also have access to the Exchange 2000 Server Resource Kit, a very valuable source of information and tools for Exchange 2000. You can also procure the Resource Kit separately.

System Requirements

Make sure that your computer meets the minimum system requirements listed in this section. If it does not, you may have a problem using the contents of the CD.

For Microsoft Windows ME, 9x, or Windows 2000, you will need

✦ A PC with a Pentium processor running at 120 MHz or faster
✦ At least 6MB of RAM
✦ An Ethernet network interface card (NIC) or modem with a speed of at least 28,800 bps
✦ Internet connection
✦ A CD-ROM drive — double-speed (2x) or faster
You may need 300MB of hard drive space to install all the software from this CD.

**Using the CD with Microsoft Windows**

To install the items from the CD to your hard drive, follow these steps:

1. Insert the CD into your computer’s CD-ROM drive.
2. Click Start ➪ Run.
3. In the dialog box that appears, type `d:\setup.exe`, where `d` is the letter of your CD-ROM drive.
4. Click OK.

**Troubleshooting**

If you have difficulty installing or using the CD-ROM programs, try the following solutions:

* Turn off anti-virus software: Installers sometimes mimic virus activity and can make your computer incorrectly believe that it is being infected by a virus. Be sure to turn your antivirus software back on immediately after installation.

* Close all running programs: The more programs you are running, the less memory is available to other programs. Also, installers typically update files and programs; if other programs are running, installation may not work properly.

For more sophisticated users: Set up your machine with drive carriers (DataExpress from companies such as Kingston) and removable drives. Doing so enables you to load some of the demo programs on fresh disks without affecting any of your normal production software. This technique delivers maximum utility out of a small number of chassis, especially in pilot programs. You also can use it to eliminate problems with demo programs.
This glossary supplements the many definitions of terms found in the book. In some cases we have extracted definitions from the main text and added them here. Sometimes we have lengthened or shortened the comments. In other cases the terms are defined only in this glossary. For definitions of terms not included here, consult one of many Web sites that provide technical definitions, such as www.netdictionary.com/html/index.html.

**Access control entry (ACE)** An object in an access control list (ACL).

**Access control list (ACL)** A list delineating the security permissions for an object that contains membership (ACE) and the actions available to each member for that object.

**Active Directory** A component of Windows 2000 Server that is utilized by Exchange 2000 Server. It replaces the Exchange 5.5 directory, providing a single repository for objects for both the operating system (Windows 2000 Server) and the application (Exchange 2000 Server). It provides unified security and replications services. Active Directory is a Lightweight Directory Access Protocol (LDAP)–compliant directory service and provides distributed access to database objects from anywhere in the organization via Global Catalog servers.

**Active Directory Connector (ADC)** A connector, provided on the Exchange 2000 CD, that synchronizes directory information from legacy Exchange 5.5 directories to Active Directory in Windows 2000.

**Active Directory System Interface (ADSI)** A set of COM interfaces used to access the capabilities of directory services from different network providers in a distributed computing environment.

**Active Server Pages (ASP)** A server-based scripting environment designed to utilize ActiveX objects that provide dynamic capabilities to static HTML pages.
**Address list**  A feature of Exchange 2000 similar in function to an address book. Address lists are a logical subset of the Global Address List (GAL) and can be used to limit the addresses viewed by clients.

**Address space**  A logical path to the foreign site or system. The address space helps control the flow of the messages across the connector.

**Administrative group**  A collection of Exchange 2000 Active Directory objects that are grouped together for the purpose of managing permissions.

**Anonymous access**  Access to resources in the network for which authentication at login is not required.

**Authentication**  The process by which permission to perform a function and/or access a resource is given.

**Bridgehead server (BHS)**  A server that transfers messages between routing groups or between a routing group and an external mail system. Bridgehead servers focus traffic through a single server on each side of the connector.

**Certificate Authority (CA)**  A third-party organization that issues digital certificates and public key pairs that are used for encryption and decryption of data. The CA verifies and certifies the identity of the party to whom the certificate is granted.

**Certificate Revocation List (CRL)**  A list of unacceptable certificates for a user and/or server.

**Certificate Trust List (CTL)**  A list of trusted certificates for a user and/or server.

**Circular logging**  A method of writing data to a log file, wherein previously written data in the file is eliminated based on one of two different criteria: when the log file reaches its maximum amount of information (the oldest log is overwritten first), and when the data reaches a certain age. In Exchange, circular logging causes the transaction log to overwrite itself, making restoration of data whose transaction logs have been overwritten potentially impossible.

**Collaborative Data Objects (CDO)**  A programming interface to the Messaging Application Programming Interface (MAPI). As an interface, CDO simplifies the task of writing server-side applications, whether they are Web-based or not.

**Collection**  An object that holds similar objects.

**Conference Access Web Pages**  A set of Web application pages hosted by IIS. Conference participants use these pages to participate in online conferences.

**Conference Management Service (CMS)**  A Microsoft Conferencing Server service that accepts reservations for online conferences, maintains conference schedules, and controls conference technology providers.
**Conferencing Manager** A Microsoft Management Console tool for managing and configuring conferencing services.

**Connection Agreement (CA)** An object that specifies how replication will occur. A CA defines a single replication connection between the Windows 2000 AD and the Exchange 5.5 Directory. It is configured with information, such as the local and remote server names, source and destination recipient containers, replication schedule, and connection security information.

**Connector** The means by which Exchange servers communicate with each other and with external mail systems across the enterprise. These include the Routing Group Connector (Exchange to Exchange), SMTP, and X.400. In earlier products such as MS Mail, connectors were called gateways.

**Container** An object that holds a collection of other objects.

**Default Global Address List** A list comprising all recipients in an organization — mailbox-enabled users, mail-enabled users, groups, contacts, and public folders.

**Digital Dashboard** A customized solution for knowledge-workers that consolidates personal, team, corporate, and external information and provides single-click access to analytical and collaborative tools.

**Digital ID** A digital signature generally granted by a public authority (such as Verisign) that uniquely identifies the originator. The Windows 2000 Certificate Service can also grant digital IDs for use in Exchange signing and sealing; these are also known as keys.

**Directory Log** The log for the Active Directory Service. Because Exchange 2000 relies on Active Directory Service and does not have its own directory, this log becomes very important in most troubleshooting.


**dirsnc** Directory synchronization; the real-time process by which multiple Exchange servers maintain up-to-date directories across the enterprise.

**Discretionary access control lists (DACLs)** Lists that protect and secure all objects in the Active Directory Service (ADS) of Windows 2000 Server, a service used by Exchange 2000 Server. Every access to any object or attribute in the Active Directory by any user or service is validated against the DACL.

**Distinguished Name (DN)** An object name used for routing messages within an organization. It is based on the X.500 standard and includes all the components of an X.500 name, differing in that an abbreviation of the component type is added antecedent to the component. Thus, cn=common name; a sample DN is cn=MEaves, ou=Consulting, dc=Valimar, dc=com.
Domain Name System (DNS)  A system whose servers can resolve a fully qualified domain name to a specific IP address. DNS is based on a standard, RFC 1035, and is like a virtual telephone directory. If the local DNS server cannot provide resolution, it will contact other DNS servers for help.

Domain Name System (DNS) Server Log  A log that provides insights into many mail-flow and connector problems. If DNS entries are wrong or inadequate, many Internet and connector protocols will suffer and may fail.

Dr. Watson Logs  Logs provided by Dr. Watson, an application installed with Windows 2000 Server. Dr. Watson monitors applications and dumps vital information when an application crashes. It does not dump information from a low-level catastrophe, such as blue screen (such catastrophes are rare in Windows 2000). Dr. Watson logs may provide vital information for many Exchange problems involving software bugs and conflicts.

Dynamic Buffer Allocation (DBA)  The process by which Exchange Server 2000 dynamically increases the size of the buffer cache for ESE.

Dynamic HTML (DHTML)  An extended version of HTML that is object-based and adds interactive capabilities to Web pages.

Exchange System Manager  A Microsoft Management Console (MMC) snap-in that can provide a framework for containing all other Exchange snap-ins, which enables you to manage an entire Exchange enterprise from a single console.

ExIFS  A file system that stores streaming data, such as audio and video, and also enables clients such as Office to read and write documents. ExIFS uses a data model that supports both hierarchical collections (folders) and heterogeneous collections (folders that accept any file type).

Extensible Markup Language (XML)  A markup language that allows programmers to develop custom schemata for describing data.

Extensible Storage Engine (ESE)  The database engine used by Windows 2000 Server Active Directory. An improved version of the JET engine used in Exchange 5.5 Server, ESE manages the transaction logging system used when writing to the Information Store.

File Replication Service log  The log to which File Replication Service makes entries. It is not as useful as the other logs in most Exchange troubleshooting.

Fully Qualified Domain Name (FQDN)  A DNS domain name that fully and uniquely defines the location of a server within a domain.

Global Address List (GAL)  The master address list for an Exchange environment. It contains information on all users, groups, public folders, contacts, and conferencing resources in the organization.
Globally Unique Identifier (GUID) A value that uniquely identifies an object.

Group A logical collection of users created to facilitate administration. In earlier versions of Exchange some of the functionality of a group was handled by a distribution list.

Hardware Compatibility List (HCL) The list of Windows 2000–certified hardware that Microsoft provides on its Web site.

Hypertext Transfer Protocol (HTTP) version 1.1 Version 1.1 of the universal Web standard that enables information to be formatted and communicated over the Web. It is the mechanism by which Web servers and clients respond to commands, such as GET and POST.

Ilmarin A “mansion of the high air” on Taniquetil in Tolkien’s world.

Instant messaging A service enabling individuals to communicate with each other instantaneously.

Internet Message Access Protocol version 4 (IMAP4) A protocol that, like POP3, allows a messaging client to connect to a messaging store remotely. Unlike POP3, IMAP4 allows messaging clients to access their mail without downloading it to their computer. To enhance the capabilities of remote access to e-mail, IMAP4 provides additional features, such as search by keyword, access to public folders, access to more than one e-mail folder, viewing message headers before downloading, downloading specific messages, downloading an attachment separately from a message, and flagging messages as unread or read.

Kerberos Key Distribution Center (KDC) service A service responsible for granting Kerberos security tickets to clients.


Lightweight Directory Access Protocol (LDAP) A service that provides connection to X.500-based directories. Directories can be multipurpose and store hierarchical information about a variety of objects.

Local Security Authority (LSA) A Windows 2000–protected subsystem whose job is to provide local computer security.

Mailbox The location on an Exchange server in which a user’s e-mail is stored.

MAPI A Microsoft-based Messaging Programming Application Interface (API).

Message Transfer Agent (MTA) The Exchange component and X.400 term that routes messages to other MTAs, the Information Store, a connector, or a gateway.
**Mixed mode**  A mixed Exchange 5.5 and Exchange 2000 Server environment.

**Multicast Address Dynamic Client Allocation Protocol (MADCAP)**  A part of the Windows 2000 Dynamic Host Configuration Protocol (DHCP) service that assigns a multicast group IP address to each video conference.

**Native Content Store**  A store containing data from non-MAPI clients in native MIME format. No conversion is performed on the data unless a MAPI client accesses the data.

**Native mode**  An environment running Exchange 2000 Server without any prior versions of Exchange Server, such as Exchange Server 5.5.

**Net Logon service**  A service that validates user credentials against a Windows 2000 Server domain controller (DC), returning domain Security Identifiers (SIDs) and user rights for the user.

**Network News Transfer Protocol (NNTP)**  The protocol used to post and distribute USENET news. The Network News Transfer Protocol (NNTP) facilitates posting, distributing, and viewing news items and group discussions between NNTP clients and news servers. The standard for NNTP is RFC 977. NNTP includes commands that allow access to newsgroups posts that are managed by a USENET provider, such as UUNET or MCI.

**Newsgroups**  Online public forums where Internet users can post questions or answers, discuss general or specific topics, or post topic-related information for other users’ education.

**Non-Delivery Report (NDR)**  A notification delivered to the sender of an e-mail that the e-mail was unable to be delivered.

**NTFS**  The file system specifically designed by Microsoft for the Windows NT family that is used by Windows 2000 Server.

**NTLM**  Windows NT LAN Manager Challenge/Response authentication method (Secure Password Authentication [SPA]) used by Exchange 2000 Server. It uses a randomization algorithm and an encrypted password to authenticate users. Kerberos has replaced NTLM as the default authentication protocol in Exchange 2000 Server.

**Offline Address Book (OAB)**  A text-based copy, stored in the system public folder, of all or part of the address list. It is downloaded by remote users and used in conjunction with other local address books for addressing while not online.

**Offline Storage Files (*.OST)**  Files for synchronizing local messages with server messages.
**Open System Interconnect (OSI)**  A standard adopted by the International Standards Association (ISO) that views the interconnection between devices as made up of seven layers, with each layer providing services to the layer above it:

- **Application layer:** Provides network access to applications.
- **Presentation layer:** Ensures that both senders and receivers of messages use the same data format, such as UNICODE, and also includes any required encryption or compression.
- **Session layer:** Responsible for establishing a communication session (all messages between senders and receivers occur within a session).
- **Transport layer:** Responsible for the quality of service of the communication system; ensures that messages are not lost or duplicated.
- **Network layer:** Decodes addresses and routes messages appropriately.
- **Datalink layer:** Packages message data into frames, controls the flow of frames from source to destination, acknowledges the receipt of transmissions, and retransmits if necessary.
- **Physical layer:** Defines the hardware, including the terminal devices, the male and female connectors used, the pin configuration, and the transfer of the data as a serial bit stream.

**Outlook Web Access (OWA)**  A server component that publishes Exchange 2000 mailbox content in a format that can be accessed from a Web browser across the Internet. The new OWA provides Web browsers with the same capabilities that previous versions provided: accessing e-mail, public folders, calendar, creation and management of password changes, name checking, and conversion of Outlook forms to HTML.

**Permission inheritance**  The process by which a child object inherits permissions from its parent object by default.

**Personal Address Book (PAB)**  The Personal Address Book (PAB) file originally used with Microsoft Schedule+. Outlook 2000 supports both PABs and Outlook contact formats for Personal Address Books but will try to convince you to import your PAB to Outlooks contacts.

**Personal Folder File (*.PST)**  Local storage folder (file) for downloading messages and contact information for Outlook clients.

**Post Office Protocol version 3 (POP3)**  A protocol used by clients to receive messages from a remote server. The Post Office Protocol (POP) comes in three versions: the original POP; POP2, which requires SMTP for electronic messaging; and POP3, which does not require SMTP (however, application soften couples POP3 with SMTP, where SMTP is used for transmission of e-mail and POP3 is used for retrieving e-mail). POP3 consists of ten core commands that enable a client to download mail for offline use. POP3 is defined in RFC 1088.
**Public folder** A repository for many different types of information that can be shared among users in an Exchange organization.

**Routing Engine Service** A service that coordinates the transfer of messages.

**Routing group** A collection of connected servers running Exchange. At least one server in each routing group must be running Exchange 2000. Messages sent between any two servers within a routing group are routed directly from source to destination.

**Routing Group Connector (RGC)** Exchange-to-Exchange unidirectional connector expecting high-speed, reliable connection.

**Schema** A collection of descriptions, or rules, that define characteristics of objects in the directory. The schema defines available object classes in the directory, relationships between object classes, attributes of each object class, and specific characteristics of attributes and classes.

**Secure Sockets Layer (SSL)** A method of encrypting a session using a public/private key technology to ensure communication privacy and data integrity. It must be set up on the server and supported and enabled on the client. SSL protects the session between the client and the server.

**Security Accounts Manager (SAM)** Location for storage of local security accounts; also enforces locally stored policies and supports APIs.

**Setup Log** Log generated when Exchange Server is installed. It is found at the root of the destination drive in which Exchange Server was installed and is called Exchange Server Setup.log. It can be read in Notepad text form.

**Simple Mail Transfer Protocol (SMTP)** The native messaging protocol of the Internet, as defined in the RFC822 standard. SMTP consists of a small set of commands that facilitate the transfer of mail between SMTP-compliant systems: DATA, EXPN, HELO, HELP, MAIL, NOOP, RCPT, RSET, SAML, SEND, SOML, TURN, and VRFY. The SMTP protocol is typically used by messaging clients to send mail.

**Site Replication Service (SRS)** An Exchange 2000 Server service that makes the Exchange Server look like an Exchange 5.5 server when directory connections are being made from Exchange 5.5 systems. An SRS server has a copy of the Exchange 5.5 directory, even though it is running Exchange 2000.

**Storage group** A group of one or more stores.

**Store** An Exchange logical database consisting of two database files: the streaming file (.stm) and the MAPI store (.edb).

**System Attendant (SA)** A background service that performs duties such as server and link monitoring, message delivery link state tables, and feeding information to other monitoring systems (such as the Event Viewer).
**T.120 MCU/H.323 Conference Bridge** A Conferencing Server component that provides an MCU for T.120 data conferencing clients and a bridge that allows H.323 clients to participate in audio and videoconferences.

**T.120 Multipoint Control Unit (T.120 MCU)** A server required for data conferences and to interconnect multiple conference participants. It synchronizes and distributes conference data between participants. The two possible configurations for an MCU are peer-to-peer and client-server.

**TCP/IP** The predominant Internet protocol suite. It includes several protocols and applications that belong to three of the OSI layers:

- **Application layer**: Includes e-mail protocols SMTP, IMAP4, and POP3, as well as FTP for file transfer, HTTP for allowing browsers to retrieve data from Web servers, and the Network News Transfer Protocol (NNTP).

- **Transport layer**: Includes two protocols that define the type of message context, TCP and User Datagram Protocol (UDP). TCP is a connection-based protocol, and UDP is a connectionless protocol. UDP is less reliable because it does not guarantee delivery of messages after a connection is established.

- **Internetwork layer**: Includes the IP protocol. IP, like UDP, is connectionless and is responsible for routing of data. It uses routing tables to determine where to forward messages, so that they can use the shortest route to reach a destination.

**Team folder** A public folder that has restricted user access to members of a team. It is a Web-based collaboration tool for Outlook 2000.

**Virtual Private Network (VPN)** A virtual non-public connection between a user and a host that utilizes public and/or private physical circuits. It can be used to encrypt an authentication and data session, normally between the client and the external interface of a VPN router. Session data on the internal network is not normally encrypted, so the client/server session is not entirely secure.

**Virtual server** A division of a physical server into distinct logical servers, which appear to the outside world as distinct Exchange 2000 Servers although physically located on a single Windows 2000 Server.

**Web Store** A storage platform that provides a single repository for managing multiple types of information within one infrastructure. There are four key components to the Web storage system: ExIFS, Native Content Store, URL addressing, and Web Distributed Authoring and Versioning (WebDAV). WebDAV enables the implementation of Web folders so that files can be opened and saved using Port 80.

**WebDAV** An extension to the Hypertext Transfer Protocol (HTTP) version 1.1. It facilitates file operations such as copy and move, as well as document tracking over the Internet.
**X.25** A protocol for transmitting data over a WAN (wide area network). Also known as TP0.

**X.400** An ISO standard for global e-mail including the addressing and transporting of e-mail messages over a WAN (wide area network).

**X.500** An ISO hierarchical directory standard.

**XML** See *Extensible Markup Language*. 
Symbols & Numbers
\ character in SMTP commands, 50
3DES encryption algorithm, 506

A
AbandonRequest command, 65
access control entries (ACEs), 119, 484, 705
access control list (ACL), 705
access control methods for NNTP, 211–212
access permissions for objects, 99
access points in TP4, 69
Access property tab
for HTTP, 224–225
for IMAP4, 221
for NNTP, 211–212
of the Virtual Server Property box, 218
Access tab of the SMTP Virtual Server Properties
dialog box, 508
access token, 485
Access Units (AUs), 38
accounts, identifying duplicate, 415
ACEs (Access Control Entries), 119, 484, 705
ACL (access control list), 705
Acrobat Reader on the CD-ROM with this book, 699
Active Directory, 123–124
adapting design and topology for
Exchange 2000, 528
administering keys using, 501, 502
categories of Exchange 2000 information stored
in, 90
configuring, 322–323
considerations for an Exchange 2000
installation, 439–440
container for objects in, 131
database size limitations, 123
deploying Outlook 2000, 455–456
Exchange configuration and user messaging
information in, 673
installing, 319
integration with, 90–91
preparing for Exchange 2000 installation, 368
schema, 439
storing security policies and account
information, 484
synchronizing with the MS Mail GAL, 398–399
updating the Schema during installation,
448, 449
verifying the installation of, 323
Active Directory Account Cleanup Wizard, 383
Active Directory Browser, 677–678
Active Directory Connector. See ADC (Active
Directory Connector)
Active Directory Connector Management
components, 374
Active Directory Connector Manager, 375–376
Active Directory Connector Service
component, 374
Active Directory Editor, 675–676
Active Directory Server, scanning for the best, 186
Active Directory Service. See ADS (Active Directory
Service)
Active Directory System Interface. See ADSI (Active
Directory System Interface)
Active Directory tools, retrieving Exchange Server
information, 675
Active Directory trees, containers in, 676
Active Directory Users and Computers snap-in, 18
active logs, 600
Active Server Pages. See ASPs (Active Server
Pages)
active/active clustering, 87
active/passive clustering, 87
ActiveX controls
adding to forms in Outlook 2000, 641
in collaborative solutions, 638
managing, 681
setting properties specific to, 642
in TFK, 680
activity, monitoring after installation, 362
ad hoc-driven application development, 557
ADC (Active Directory Connector), 372, 705
creating the service account for, 368
installing, 374, 433
prerequisites, 374
ADC service account
  creating, 368
  specifying, 375
ADClean utility, 415
AddRequest syntax in LDAP, 65
address lists, 96, 706
  building with OUs, 131
  creating, 107–108
  custom, 106
  default, 105–106
  maintaining, 186
  managing, 104–110
  offline, 106
  securing, 109
  types of, 104–106
  updating, 109–110
  updating with new recipient information, 110
address providers in Outlook 2000, 233–235
Address shared folder, 450
address space, 706
address space entries, 197–198
Address Space property page
  scoping connectors, 592
  for the SMTP Connector, 197–198
Address Space tab
  of the cc: Mail Connector, 388
  of the Connector for Lotus Notes, 407
  of the Connector for MS Mail, 398
Address Spaces, establishing, 190–191
addressing in X.400, 42–43
ADE exchange with the cc: Mail environment, 388
ADMD (Administrative Management Domain), 39, 40
admin workstations, 630
admin.exe
  in earlier versions of Exchange, 15–16, 17
  MS Mail administration program, 394–395, 398
administration costs of Web Store, 172
Administration dashboard, creating a new digital dashboard application, 693–694
Administration Delegation Wizard, 474
administrative account, 438
administrative assignments, 558–559
administrative boundaries, 527–528
Administrative DMD (ADDMD) in X.500, 57–58
administrative groups, 91, 101–103, 137–138, 706
  for Conferencing Server, 480
  creating, 102
design considerations for creating, 526
  displaying, 99
  planning, 523–524
  selecting for Conferencing Server, 477–478
Administrative Groups container
  creating administrative groups, 102
  granting permissions, 103
  for an Organization object, 100
Administrative Management Domain (ADMD), 39, 40
administrative models, 17, 23, 514, 524–525
administrative notices, 618, 619
administrative permissions, 131
Administrative program, loading on a second Windows 2000 machine, 630
Administrative property page for KMS, 496
administrative responsibilities, transferring after deployment, 364
administrative roles, 120
administrative security, 119–120
administrative tasks, scheduling, 561, 565–570
administrative tools, 17
Administrative Tools programs, 323–324
Administrator e-mail account text box for NNTP, 210
administrators
  adding to KMS, 496
  granting permissions to, 120
Administrators group, VBScript adding a user to, 669
Administrators Rights permission category for public folders, 114
ADO (Active Data Objects), 672
ADS (Active Directory Service), 17–18, 21–22
  as the address provider for Exchange 2000, 233
  leveraged by Exchange 2000, 124
  log for, 617
Namespaces container, 662–663
naming in, 516
as part of the Windows 2000 security subsystem, 483–485
recovering, 609
relationship to X.500, 38–39
schema, 22, 515
support for X.500, 53
verifying an object for the server to be restored, 611
ADSI (Active Directory System Interface), 651–653, 705
accessing Exchange 2000 objects in Active Directory, 672
binding, 654
COM object, 652
Edit utility, 102, 103
errors, 671–672
interfaces, 653
objects, 652, 658
providers, 677
providers shipped with, 655
Reference URL, 672
ADSIEdit, 675–676
ADSIEDIT/LDP, 18
ADSVW.exe, 677
Advanced delivery property page, 219–220
advanced properties dialog box in Outlook 2000, 642
Advanced property page
for the SMTP Connector, 196–197
for the X.400 TCP Connector, 204–206
Advanced Security object, 495
Advanced tab
of a CA, 378
of the cc: Mail Connector, 389
of the Connector for Lotus Notes, 408, 409
alerts
providing access to, 563
setting up for message queues, 158–159
tracking down and dealing with, 566
Algorithm property page, 506–507
All Address Lists container, 107–108
All Contacts address list, 105
All Global Address Lists container, 107–108
All Groups address list, 105
All Public Folders hierarchy, 688
All Users address list, 106
alternate port numbers, configuring in Outlook Express, 270
annual tasks, 568–569
anonymous access, 212, 706
in OWA, 305, 307
for the SMTP Connector, 196
API gateway, facilitating connections to GroupWise, 411
AppCenter 2000, 546
Apple, OSI model and, 36
Application Converter, 408, 410
application development in a messaging environment, 557–558
Application Event log, examining after installation, 452
application layer
of the OSI model, 35, 36, 534, 535, 711, 713
of the TCP/IP protocol stack, 534, 535
application layer protocols, 70, 536–537
application log, 616
application settings, creating consistent with OUs, 131
applications
assigning to users, 456
on the CD-ROM with this book, 699
checking configuration settings for, 290
converting from Lotus Notes to Exchange, 410
making available using Active Directory, 455–456
publishing, 456
AQE (Advanced Queuing Engine), 141
arbitration, 431
architectural models for conferencing, 458
architecture of Exchange 2000, 81–84
Archive Sink, 585
Arpa domain name, 73
Ashby, Erik, 66
ASPs (Active Server Pages), 688, 705
first appearance of, 15
in previous versions of OWA, 297, 298
assigned applications, 456
assistants in Outlook 2000, 251–253
association parameters on the X.400 TCP Connector Override property page, 204
asymmetric algorithm, 506
asynchronous collaborative solutions, 636
asynchronous events, 696
attachments, allowing to go as file transfer body parts, 205
attributes, described in classes, 673
“Audit Failure” in Event Viewer, 566
auditing security-related events, 486–487
AUs (Access Units), 38
authenticated state for an IMAP4 server, 76
authentication, 706
  methods of, 212, 293–294
  options for Outlook Express, 277–278
  options for OWA, 305–308
  options in LDAP, 61
  planning for, 520
  setting for NNTP, 211–212
Authentication Methods property tab for HTTP, 225
auto signature, 243
AutoArchive, 245
"automagical" information transfer, 140
automated tools for creating profiles, 238
automating the installation process, 452–456
"Available virtual memory" monitor, 178
AWP tool, 585

B
backbone, Exchange 2000 as, 593
back-end performance, monitoring after
  installation, 362–363
back-end server, 82, 83
back-end services, implementing for the pilot
  phase, 358
backing up, 603–604
BackOffice, 13–14
  components, 529–530
  servers, 522–523
backup and restore, standards and conventions
  for, 559
backup employees, 559
backup generators, 569
backup plan, 602
backup servers, 630
backup strategies, 601–602
backup tapes, 567
backups
  hardware for, 426
  for PIMs, 258
  of PST files, 241
types of, 604
  verifying, 568
  verifying/validating, 606
backups/restores, Exchange-aware, 596–597
"Bad IP address or Unknown Host" response, 291
badmail directory, 218–219
bandwidth
  considerations for profiling POP3 and
    IMAP4, 296
  managing, 530
  profile of Outlook 2000, 254–255
  between servers, 518–519
basic authentication, 212
  for OWA, 305–306
  SSL with, 307, 308
basic control properties, 641–642
BDC (Backup Domain Controller), 133
“best of breed” model, 555
BHS. See bridgehead servers
bi-directional trusts, 129
binding in ADSI, 654–655
binding string, 654
BindRequest command in LDAP, 61
BinHex for Macintosh compatibility, 222
BIOS versions for PIMs, 257
Blackberry Exchange Service property page, 594
Blackberry wireless devices, 259, 594–595
block algorithm, 506
Blue Book for the X.400 standard, 34
body part number assignments in the X.400
  standard, 46
body parts, specifying for the X.400 TCP
  Connector, 205
bottlenecks, testing for, 351
boundaries, determining for the implementation
  project, 348
BP-15, allowing, 205
bridgehead servers, 191, 377, 630, 706
browsers, 700
  supported by OWA, 299
budget
  in an implementation plan, 341
  working within, 531
built-in forms in Outlook 2000, 639
bulk encryption key, 503
bulletin board replication strategies, 390
bursting in SMTP, 47–48
business organization, model for, 514–515

C
CA (Certificate Authority), 494, 706
caching DNS servers, 128
CALs (Client Access Licenses) for Conferencing
  Server, 28
canonical names, 126
capitalization of mailbox names, 517
CAs (Connection Agreements), 372–374, 375–378, 707
CAST-40 encryption algorithm, 506
CAST-64 encryption algorithm, 506
Categorizer, 141
cc: Mail
  address type, 389–390
  checking the database version of, 386
  coexistence with and migration from, 383–392
  compared to MS Mail, 392
  configuring a connection to, 386–390
  as the master location, 390
  migration tools, 391–392
  reasons for migrating from, 384
cc: Mail Archive Importer, 391
cc: Mail Automatic Directory Exchange (ADE), 384
cc: Mail connector, 384
  connecting to a dedicated connector PO, 384–385
  as an installation component, 444
  requirements for, 386
cc: Mail information store for Outlook 2000, accessing legacy e-mail data, 232
cc: Mail Migration Directory Toolkit, 391
cc: Mail systems, creating multiple connectors to, 385
CCITT. See ITU (International Telecommunication Union)
CD key
  entering for Conferencing Server, 475
  entering for Exchange 2000, 324
  writing on the CD with permanent marker, 443
CD Rendering Library, 644, 645–646
CDO (Collaborative Data Objects), 649, 706
  libraries, 25, 644–645
  selections, 695–698
CDO for Exchange Management, 649
CDO for Windows 2000 (CDOSYS), 649
CDO for Workflow objects, 649
CDO Workflow Objects for Microsoft Exchange, 649
CDOEX, 649
CDOEXM.DLL, 649
CDONTS, 649
CDOWF.DLL, 649
CD-ROM with this book
  evaluation of HyperSnap, 553
  information on useful URLs and newsgroups, 622
  LDAP program example, 66
  links to the HCL (Hardware Compatibility List), 435
  storage requirements checklist and Excel spreadsheet, 425
  summary of the contents of, 699–701
  system requirements for running, 703–704
CDs, “home-burned” as software copies, 436–437
central administration, allowed by ADS, 22
centralized directory in X.500, 58
centralized management model, 524, 525
centralized management, standards and techniques for, 7
Certificate Authority (CA), 494, 706
Certificate Revocation List (CRL), 489, 706
Certificate Server, 489
Certificate Services, 488–489
Certificate Trust List (CTL), 706
certificate Web screen wizard, 490
Certificate Wizard, 218
certificates
  generated by users, 498–499
  revoking, 501, 502
  chaining in X.500, 58
  challenge/response authentication protocol, 487
  Change Security Settings dialog box, 505
  chat communities, 118
  chat in data conferences, 464
  Chat Service, 19, 445
  checklists for installation and follow-up, 357
  checkpoint file, 601
  child containers in the Organization object, 100
  child object in the CDO Rendering Library, 645
  children in ADSI, 659–663
  circuits, alternatives for, 68
  circular logging, 706
    enabling, 163
    setting, 600
    warning on turning off, 601, 602
class C address, 70
classes, 673
“clear text” authentication, 293
Client Access Licenses. See CALs (Client Access Licenses)
client configuration settings, 290
client performance, monitoring after installation, 363
client policy for IP security, 491
client protocols in earlier versions of Exchange, 15
Client security settings, 505
client software, automating the installation and setup of, 455–456
client-based rules, 252
clients
   for conferencing, 469–470
   connecting with, during installation, 333–337
   supported by Exchange 2000, 83–84
   virus protection for, 511
client/server access over the Internet, 540–545
client-server conferencing, 458
client/server messaging system, 81, 82
CLNP (Connectionless Network Layer Protocol), 69
cloning process, 454
clump, 170
cluster quorum files, 598
cluster server node, backing up data on, 606–607
clustering, 86–87, 170
   considerations about, 441
   four-way active, 427–428
clusters, 170
CMS (Conference Management Service), 28, 460, 461–462, 477, 706
collaboration
   personal folders and, 240
   collaborative architecture, 638
Collaborative Data Objects. See CDO (Collaborative Data Objects)
collaborative knowledge working. See groupware
collaborative solutions, 635–636
   administrating Outlook 2000, 679–695
categories of, 636
   components of, 637–638
designing, 638–639
developing with Exchange features, 644–646
developing with Outlook 2000, 639–644
   enhancements in the development of, 646–649
types of, 636–637
collections, 706
   in the CDO Rendering Library, 645
   related to objects in ADSI, 659
   VBScript for binding to, 660–661
columns, sorting in Outlook, 254
Com domain name, 73
COM programming concepts, 654–662
command set of LDAP, 59–66
common namespace, 515
compaction, 573
CompanionLink, 257
CompareRequest command in LDAP, 65
“compatibility and personal connection conundrum”, 7
complementary newsfeeds, 213
Complete setup type for Conferencing Server, 476
Component Object Model (COM) add-ins, 638
Component Selection screen
   in the Exchange 2000 Installation Wizard, 326
   in the Installation Wizard, 443–445
   installing connector software, 380
component selection screen for installing the ADC, 374
Component Summary dialog box in the Installation Wizard, 447–448
components, installation options for, 444–445
compression, 573
Computer dialog box, restricting access to the Virtual Server, 509
concept phase of the deployment plan, 354–355
conditions for building filter rules, 108–109
conference access pages, 467
Conference Access Web Pages, 477, 706
conference calendar mailbox, 467
calendar mailbox
conference host, finding, 458
conference invitations, coordinating, 458
conference organizer, software requirements for, 471
conference panel, 466
conference resources
reserving, 466
scheduling, 460
conference topology, 458
conference window, 466
conferences. See also data conferences
creating, 467
joining, 460–461
scheduling, 460
Conferencing Administrator, 474
conferencing capabilities, 95
conferencing clients, types of, 469
Conferencing Manager, 477, 707
Conferencing Server, 28–29, 85, 457, 521
administration of, 479–481
best practices for deploying, 473
features of, 458–459
initial screen for installation, 474
installing and configuring, 473–478
key components of, 461–468
post-installation tasks, 478–479
security provided by, 467
using in your organization, 460–461
Conferencing Server Administrator, 462
Conferencing Server site
hardware and software requirements, 469–471
planning, 472–473
Conferencing servers as dedicated servers, 630
conferencing sites, creating multiple, 473
conferencing topology, automating, 465
configuration partition, 90
Configuration Wizard, accessing within Outlook Express, 268
Connected Routing Groups property page for the SMTP Connector, 148, 198
Connection Agreements (CAs), 372–374, 375–378, 707
connection control on the Access property tab, 218
connection credentials, overriding Exchange 5.x, 194
Connection dialog box, 508–509
connection retry values on the X.400 TCP Connector Override property page, 204
connection-based protocol, 536
Connectionless Network Layer Protocol (CLNP), 69
connectionless protocol, 536
Connections tab
of a CA, 377
of the Connector for MS Mail, 396, 397
connectivity, checking, 290–291
connectivity options for PIMs, 257
connector files, 598
Connector for Lotus Notes, 404
configuring, 406–408
as an installation component, 444
Connector for MS Mail, 392
components of, 393
preparing for the implementation of, 394–395
Connector for Novell GroupWise, 93, 410–411
configuring, 413
as an installation component, 444
coster ID (ExConnect), 405
Connector Interchange in Connector for MS Mail, 393
Connector MTA in Connector for MS Mail, 393
Connector MTAs tab of the Connector for MS Mail, 397
Connector PO in Connector for MS Mail, 393
connector protocols in earlier versions of Exchange, 14–15
Connector server, 521
connectors, 12, 84, 92, 707. See also enterprise connectors; routing group connectors
creating, 117–118
for each routing group, 144
in earlier versions of Exchange, 14
evaluating for migration, 519–520
in Exchange 2000, 24
installing, 380–381
in intermediate-size organizations, 589
moving to Exchange 2000, 371
notices sent by, 618, 619
types of, 92
Connectors container for an Organization object, 100
consultants, filling engineer positions with, 344
consulting skills versus administrative skills, 551
contacts, 96. See also mail-enabled contacts
Contacts module in Outlook 2000, 234
ContainerRenderer object, 646
containers, 707
  in ADSI, 659–663
  choosing for ADC replication, 377
  in a DIT, 56
  specifying for directory replication to cc: Mail, 389
Content Restrictions property page
  for the Routing Group Connector, 147, 194
  for the SMTP Connector, 196
contract human resources, 349
Control Message check box for NNTP, 210
control messages, used by NNTP servers, 210
control properties, setting for forms, 641–642
controlled application development, 557
controls, adding to forms in Outlook 2000, 641
copy backup, 604
Copyhere method, 660
core directory services for Exchange 2000, 122
Corporate or Workgroup mode of Outlook 2000, 287
Corporate or Workgroup option for storing data in the Exchange Information Store, 239–240
corporate standards installation plan, 341
corporate WAN, 6
costs. See also administration costs of Web Store
  assigning to enterprise connectors, 189–190
  assigning to network routes, 151
Count property, 660
counter log for monitoring message queries, 158
country codes in Internet domain names, 73
CPU Utilization monitor, 178–179
Create method, 660
/CreateUnattend switch during install, 454
criticality analysis in the backup plan, 602
CRL (Certificate Revocation List), 489, 706
\CRLF\ in SMTP commands, 50
.CSV format for saving event logs, 567
CTL (Certificate Trust List), 706
CTRL+SHIFT+F, 254
custom address lists, 106
custom fields for PIM synchronization software, 258
Custom installation mode, 443, 445–446
Custom installation type for Conferencing Server, 477
custom schedule for a Routing Group Connector, 194
custom searches of recipients, 109
Custom setup type for Conferencing Server, 476
custom-built EDK connectors, 371
D
DACLs (discretionary access control lists), 119, 483–484, 707
daily tasks, 566–567
DAP (Directory Access Protocol), 53, 58, 59. See also LDAP (Lightweight Directory Access Protocol)
data
  encrypting directly on volumes using NTFS, 490
  ensuring the integrity and security of, 294–295
  verifying after backup, 604
data access standards, 27
data and voice conferencing, 95
Data Communications Equipment (DCE), 67
data conferences, 463–464. See also conferences
data conferencing clients
  hardware requirements for, 469
  software requirements for, 471
Data Conferencing Provider, 29, 462–464
data consistency, regaining with ISInteg, 576–578
data link layer of the OSI model, 35, 36
data management, effect on the design of physical placement of servers, 526
data storage in Outlook 2000, 239–242
data store databases, 163–164
Data Terminal Equipment (DTE), 67
data throughput, balancing across I/O channels, 426
data transfer between Exchange and Lotus Notes, 405
database logs, size of, 599
database pages, zeroing out deleted, 163
databases
  backing up selective, 603
  in collaborative solutions, 638
  content of, 166–167
  creating, 162, 163–164
  impact of multiple on RAID structures, 425
  multiple in a storage group, 25
  physically separating, 435
  repairing with ESEUtil, 573–576
restoring, 605
using multiple smaller, 435
datalink layer of the OSI model, 534, 535, 711
DBA (Dynamic Buffer Allocation), 590, 708
DC (Domain Controller). See domain controller
DCE (Data Communications Equipment), 67
DCE devices, connecting to DTE devices, 68
Debug Symbol files, 617
Dedicated Recovery System (DRS), 607–608
default address lists, 105–106
default administrative group, renaming, 91
default Exchange virtual server, stopping, 312
Default Global Address List, 106, 707
default password for KMS administration, 496
Default SMTP Virtual Server Properties dialog box, 508
Default Virtual Server in the IIS snap-in, 508
“default_first_site”, 134
defragmentation, 573, 574
recommendation of, 620
Delete method, 660
deleted database pages, zeroing out, 163
Delivery Options property page
for a Routing Group Connector, 193–194
for the SMTP Connector, 196
Delivery property tab of the Virtual Server Property box, 219
Delivery Restrictions property page
for a Routing Group Connector, 193
for the SMTP Connector, 197
DelRequest command in LDAP, 65
DelResponse command in LDAP, 65
deployment. See rollout
deployment plan
building, 354–359
components of, 342
deployment team, 346
DES encryption algorithm, 506
design of Windows 2000 and Exchange topologies, 356
destination information, SMTP responses for incorrect, 49
Details property page for the SMTP Connector, 197
developers, standards-based needs of, 7
development
capabilities of Exchange 2000, 19
phase of the implementation plan, 356–357
platform, 94–95
resources, 17
DFS root server, returning from a local site, 136
DHCP Administration Tool, 465
DHCP service, 464–465
DHTML (Dynamic HTML), 545, 708
diagnostics
enabling logging for the Exchange 2000 Server, 566
setting to minimum, 614
Diagnostics Logging tab, 292
dialogue, minimum number of commands in an, 52
dial-up networking, integration of Outlook Express with, 275–276
Dial-Up Networking phone book entries, 275–276
Dial-Up Networking tab of the Exchange Server dialog box, 249
DIB (Directory Information Database), 54, 55
differential backup, 604
digital certificates
generated for users, 499
required for private conferences, 467
Digital Dashboard, 637, 690–695, 707
creating, 693
customizing a, 694
integrating with Outlook, 695
digital IDs, 707
e-mail message to users, 500
in Outlook Express, 274–275
procuring, 501
digital signature
adding, 501
selecting in the Message Options page, 504
verifying, 505–506
Dijkstra, Edsger, 151
directories, topologies of X.500, 58
directory
in earlier versions of Exchange, 15
X.400 standard and, 38
X.500 rules enforced, 57
directory data, migrating to Windows 2000, 381
Directory Information Database (DIB), 54, 55
Directory Information Tree (DIT), 55–56
Directory Log, 707
Directory Management Domains (DMDs), 57–58
Directory Name Service. See DNS (Directory Name Service)
directory protocols in X.500, 58
directory replication
  between Exchange 5.5 and Windows 2000, 372
  schedule for replicating changes from the AD to cc: Mail, 389
Directory Rights permission category for public folders, 114
Directory schema, 57
Directory Service (DS), 233
Directory Service Agent (DSA), 55
Directory Service Module, 485, 707
directory services
  configuring in Outlook Express, 270
  network traffic generated by, 255
Directory Store, 124
directory structure of Exchange 2000, 22
directory synchronization configuration in MS Mail, 394
Directory System Agent (DSA), 39, 54–55
Directory System Protocol (DSP), 59
Directory User Agent (DUA), 39, 54
Dirsync, 707
Dirsync configuration in MS Mail, 394
Dirsync Options tab of the Connector for Lotus Notes, 408
Dirsync process in MS Mail, 398
Dirsync requestor
  configuring Exchange 2000 as, 399–400
  creating in Exchange, 399
  in MS Mail, 398
Dirsync schedule, 408
Dirsync Schedule tab of the Connector for Novell GroupWise, 413
Dirsync Server
  configuring Exchange 2000 as, 401–403
  configuring the general properties of, 401
  in MS Mail, 398
  re-configuring the MS Mail environment to support, 402–403
Dirsync synchronization numbers, resetting, 403
disaster simulation, 569
discretionary access control lists (DACLs), 119, 483–484, 707
discussion groups, 636
disk defragmentation. See defragmentation
disk I/O channels, balancing data throughput across, 426
disk space, running out of, 619
disk storage systems, calculating the size of, 422–425
Distinguished Name (DN), 56, 707
distributed database hierarchy of DNS, 125
distributed directory in X.500, 58
distributed management model, 524, 525
distribution groups, 105, 331
Distribution List Manager program, 66
distribution lists, 38
  migrating to Exchange 2000, 371
  updating in LDAP, 66
Dirsync (Directory Synchronization), 707
DNS (Domain Name System), 71–73, 124–128, 537, 708
  hierarchy, 125
  resolver model, 71–72
  Server log, 617
  servers, accessing, 71
  service, 71
DNS (Directory Name Service), compared to X.500, 53
documentation, 549–550
  of ongoing support, 551–552
  of planning and design, 550–551
domain controller, 133, 134
  creating in a new domain, 319–320
  overriding the default, 98
  upgrading from Windows NT 4.0, 493
Domain local group, 331
domain names
  conforming to DNS character standards, 516
Internet, 73
  for nodes in DNS, 72
Domain Name System. See DNS (Domain Name System)
Domain Name System (DNS) Server Log, 708
Domain Naming Master, 132
domain partition, 90
domain tree, 320
domains
  compared to zones, 126–127
  container for, 128
  in DNS, 72
  in the DNS hierarchy, 126
  programming with ADSI, 663–664
  in SMTP, 50
  in Windows 2000, 130
DOS, limitations of, 13
down time, minimizing, 351
Dr. Watson logs, 617, 708
drivers, obtaining Windows 2000-compatible, 435
DRS (Dedicated Recovery System), 607–608
DS (Directory Service), 233
DSA (Directory System Agent), 39, 54–55
DSCAccess cache, 585
DSCFlush tool, 585
DSP (Directory System Protocol), 59. See also LDAP (Lightweight Directory Access Protocol)
DTE (Data Terminal Equipment), 67
DUA (Directory User Agent), 39, 54
dumb terminals, emulating with PCs, 4
dUN (Dial Up Networking) phonebook entries,
  configuring in Outlook Express, 275–276
duplicate accounts, identifying, 415
Duplicate Proxy tool, 585
DXAFIX tool, 586
Dynamic Buffer Allocation (DBA), 590, 708
Dynamic Host Configuration Protocol. See DHCP service
Dynamic HTML (DHTML), 545, 708
dynamic information in Web parts, 690

E
early binding objects in a compiled language, 652
.edb file, 87, 166, 167, 598
edb.chk file, 601
Edu domain name, 73
EFS (Encrypting File System), 490
EFS certificate, procuring, 490
EHLO command, 51
Eicon X.25 card, 200
EIDE controllers, 420
electronic directories, 52
electronic version of this book, 701
e-mail. See also messages
  leaving on the server, 592
  restoring lost, 595
  retention rate for, 424
  securing the content of, 510
  solving delivery problems, 626
e-mail account, adding, 541–545
e-mail addresses
  allowing multiple for each object, 515
  VBScript for retrieving, 674–675
E-mail Addresses tab, enabling the cc: Mail address type, 390
e-mail messages, created by Link and Server monitors, 183
e-mail notifications, 180–181
e-mail protocols, 536
E-mail Server Names dialog box in Outlook Express, 268
E-mail Server Names screen in the Internet Connection Wizard, 542
e-mail services options, selecting for Outlook 2000, 334
e-mail systems
  connecting between companies, 5–7
  early, 3–5
  “Enable Message Tracking” check box, 159
  “Enable offline access” check box, 246
  encrypted PST, 572
Encrypting File System (EFS), 490
encrypting messages, 503–504
encryption
  algorithms, 506
  in Outlook Express, 274–275

Continued
encryption (continued)
    selecting in the Message Options page, 504
    for sessions, 294–295
    turning on, 256
Ends with condition for building filter rules, 108
definition of, in the Message Options page, 504
    for sessions, 294–295
    turning on, 256
Ends with condition for building filter rules, 108
definition of, in the Message Options page, 504
    for sessions, 294–295
    turning on, 256
Ends with condition for building filter rules, 108
engineer positions, filling with outside consultants, 344
English mnemonic address fields, 42–43
Enroll Users in KMS, 498
Enrollment property page of KMS, 497
Enterprise Admins group, 129
enterprise connectors, 189–191. See also connectors; routing group connectors
Enterprise Edition
    compared to the Standard Edition of Exchange, 436
    of Exchange 2000, 20–21
Enterprise Server, multiple databases allowed
    by, 25
envelopes in X.400, 41
environments, mixed Exchange, 592–593
Err object, 671
eroof handling in ADSI, 671–672
Error utility, 583
escalation section of the ongoing support plan, 552
ESE (Extensible Storage Engine), 123, 161, 590, 708
ESEUtil, 18, 573–576
Eudora
    on the CD-ROM with this book, 699
    as an Internet mail client for POP3 and IMAP4, 285–287
    location of configuration settings, 290
    as an SMTP client, 83
evaluation and testing team, 345–346
event logs
    clearing, 567
    deleting, 567
    examining, 582–583
    formats for saving, 567
    overwriting, 567
    event sinks, 19, 94, 697
    preventing the misuse of, 510
    running in the store event architecture, 696
    setting up and monitoring, 627–629
Event Viewer
    examining the application log after installation, 452
    logs in, 566–567, 616
    viewing the Event Log, 582–583
events, 94, 582
.evtx extension, 582
.EVT format for saving event logs, 567
Exchange 4.x/5.x, differences from Exchange 2000, 608–609
Exchange 5.5
    address book as an address provider in Exchange 2000, 233
    compared to Exchange 2000, 15–20
    directory coexistence and migration from, 371–372
    Information Store, 231
    installing Exchange 2000 to coexist with, 432–433
    migration scenarios for coexisting with, 514
    migration scenarios for removing all servers, 514
    naming in, 515–516
    shortcomings of, 121–122
    sites, 370
    upgrading clients, 20
    upgrading OWA from, 302–303
    verifying readiness for coexistence with Exchange 2000, 368–369
Exchange 5.5 Administrator, 445
Exchange 5.5 directory. See Active Directory
Exchange 2000
    adding to Active Directory, 329
    administrative of, 22–23
    administrative model, 23
    advantages created by using ADS, 22
    architecture of, 81–84
    as a backbone, 593
    clients supported, 83–84
    compared to Exchange 4.x/5.x, 15–20
    components of, 94–96
    configuring as a Dirsync requestor, 399–400
    configuring as the Dirsync Server, 401–403
    configuring to connect to GroupWise, 413
    connecting to MS Mail, 395–398
connectors, 24
core directory services provided by Windows 2000, 122
data storage in, 18–19
data transfer with Lotus Notes, 405
development improvements, 19
as a development platform, 94–95
development resources in, 17
differences from Exchange 4.x/5.x, 608–609
directory structure of, 22
as an External Foreign Domain in the GroupWise system, 412
housing the master copy for cc:Mail replication, 390
IMAP4 and, 284–289
IMAP4 interface, 285
as an installation component, 444
installing, 324–332
installing in a new messaging environment, 432
installing in an existing Exchange 5.5 environment, 432–433
installing on new servers, 433
installing to coexist with other mail systems, 433–434
instant communication in, 19
integration with Windows 2000, 89–94
internationalization of, 28
introducing into an existing Exchange environment, 370–371
leveraging of ADS, 124
migration tools, 24
minimum practical requirements for a server, 418–420
multilingual capabilities of, 20
naming in, 516
overview of, 20–21, 85–96
OWA features in, 299–300
planning the implementation of, 339–366
POP3 and, 284–289
POP3 connector, 75
POP3 interface, 285
real-time communication in, 26
security features of, 494–507
server roles, 521
setting up an account, 540–545
snap-ins available with, 92–93
transferring data to GroupWise, 411
transferring data with MS Mail, 393–394
transferring messages from MS Mail, 393–394
upgrading to, from Exchange 5.5, 15–20
versions of, 20–21, 85
Windows DNA and, 29–30
workflow capabilities of, 25–26
Exchange 2000 ADS Catalog, 18
Exchange 2000 Conferencing Server. See Conferencing Server
Exchange 2000 Enterprise server, 85
Exchange 2000 Information Store, 232
Exchange 2000 Installation Wizard, 325
Exchange 2000 NNTP. See NNTP (Network News Transfer Protocol)
Exchange 2000 Resource Kit, 585–586
Exchange 2000 schema, 673
Exchange 2000 Server, 85
appearing as an ADS object, 611
recovering, 610–612
unattended setup of, 454
Exchange 2000 servers
building as member servers, 611
installing into an existing Exchange 5.5 site, 370–371
Exchange 2000 SMTP Connector. See SMTP connector
Exchange accounts, replicating to cc:Mail, 389–390
Exchange Administration Delegation Wizard, 102, 120, 698
Exchange Administration tool, 309
Exchange Administrator role, 120, 510
Exchange Advanced Security, 93
Exchange Application Services for Lotus Notes, 410
Exchange bridgehead server, 377
Exchange cc:Mail snap-in, 93
Exchange client, configuring Outlook as, 236–251
Exchange, coexistence with earlier versions of, 367–378
Exchange Conferencing Server services role, 471
Exchange Conference Services, 93, 479
Exchange directory, 123
Exchange Dirsync Requestor, 399–400
Exchange Domain Servers, 677–678
Exchange environment, defining as a foreign domain in Lotus Notes, 406
Exchange Full Administrator role, 120, 510
Exchange General tab for new Exchange objects and attributes, 450, 451
Exchange Information Store
  compared to the Web Storage System, 647
  scheduled maintenance run by, 574
  storing data in, 239–240
Exchange Installable File System. See ExIFS
Exchange Mailbox Merge tool, 586
Exchange Message Tracking Center, 93
Exchange Migration Wizard. See Migration Wizard
Exchange modeling tool, 351
Exchange Monitoring and Status Tool, 562–563
Exchange Object Model, 672
Exchange Organization
  creating a new, 326, 327
  entering in the Installation Wizard, 447
Exchange Router service for GroupWise, 413
Exchange routing groups. See routing groups
Exchange Server CD, 454
Exchange Server Management Console, managing server-centric components, 23
Exchange servers
  configuring access, 335
  earlier versions of, 14–15
  keeping separate from BackOffice and Windows 2000 servers, 522
  stopping and restarting, 181
  virus protection packages for, 512
Exchange Server Services monitor, 177
Exchange Server Setup.log, 617, 712
Exchange Servers snap-in, 93
Exchange Server technical resource kit, 362
Exchange service account, 438
Exchange services, running on the local machine, 438
Exchange SMTP snap-in, 93
Exchange System Manager, 92, 93, 97, 708
  creating public folders, 113
  creating system policies, 103–104
  directing to a specific domain controller, 98
  GAL in, 105
  starting, 97–98
Exchange System Manager console, 17
Exchange Tasks Wizard, 111
Exchange users. See recipients
Exchange View Only Administrator role, 120, 510
Exchange X.400 snap-in, 93
Exchange-aware backups/restores, 596–597
Exchsrvr folder, 597
.exe programs, identifying the version number of, 584
executive sponsorship and buy-in for the implementation project, 347
executive summary in an implementation plan, 341
ExIFS, 112, 708
  client, 84
  in the Web Storage System, 89
Expiration policies, setting in NNTP, 214
Export Containers tab
  of the Connector for Lotus Notes, 407
  of the Connector for Novell GroupWise, 413
  of the Exchange Dirsync Requestor, 400
Export Container tab of the cc: Mail Connector, 389
Export Wizard in Outlook 2000, 253
extended permissions, 119
Extensible Markup Language. See XML (Extensible Markup Language)
Extensible Storage Engine (ESE), 123, 161, 590, 708

F
  failure and recovery modes, 419
  fault tolerance
    in Exchange 2000, 86–87
    increasing with additional processors, 421
  FAX service with Outlook, 235
  Feeds container, 213
  Field Chooser, 640–641
  fields
    adding to forms in Outlook 2000, 640–641
    in collaborative solutions, 638
  File Replication Service log, 617, 708
  file transfers in data conferences, 464
  files, required for backups, 597–598
  FileVer utility, 584
  Filter property, 660
    setting to an array, 662
    VBScript for limiting the collection returned, 661
filter rules
   conditions for building, 108–109
   creating for address lists, 107, 108–109
filtering, enabling in the Virtual Server Property box, 217
filters
   adding to Web parts, 690
   for PIMs, 258
Find Custom Search dialog box for an address list, 109
find utility in Outlook 2000, 253–254
firewalls
   conferencing across, 459
   getting by, 612
First Administrative Group, 101, 102, 137, 478
First Routing Group, 143
First Storage Group, 162
fiscal resources for the implementation project, 349
flags. See message flags
Flexible, Single Master Operation (FSMO), 132
Flexible, Single Master Operation roles, 129, 132
Folder Rights permission category for public folders, 114
folders in collaborative solutions, 637
foreign domain, defining Exchange as, in Lotus Notes, 406
forest root domain, 129, 440
ForestPrep, 368, 439
forests, 128–129, 130
   creating, 321
   preparing for an Exchange installation, 439
Form Layout Tools in Outlook 2000, 641
form properties, setting up, 640
forms, 95
   in collaborative solutions, 638
   designing in Outlook 2000, 639–643
   in earlier versions of Exchange, 15
   programming in Outlook 2000, 644
forward path in SMTP, 50
four-way active clustering, 427–428
FQDN (Fully Qualified Domain Name), 126, 220, 708
fragment of an X.500 directory, 58
free and busy information, network traffic generated by, 255
Free Disk Space monitor, 179
Free Space Threshold, 179
Free-Busy information, synchronizing, 411
front-end servers, 82, 83, 309, 310
front-end/back-end architecture
   configuring for OWA, 304
   of Exchange 2000, 82–83
   scalability with, 428
FrontPage 2000, 649
FSMO, 132
FSMO Roles, 129, 132
FTP (File Transfer Protocol), 70
full backups, determining how often to create, 602
Full Exchange permissions, granting to a named account, 439
full server recovery, 610
full-text indexing, 95–96
Fully Qualified Domain Name (FQDN), 126, 708
Fully Qualified Domain Name text box, 220
functionality, avoiding for server names, 516
G
GAL (Global Address List), 104, 708. See also Exchange 2000 ADS Catalog
   in the Exchange System Manager, 105
   synchronizing with the MS Mail GAL, 398
Galaxy from CommVault, 602
gateways, 12, 707. See also connectors
GCs (Global Catalog servers), 134, 233, 485
GDI settings for the X.400 TCP Connector, 206
general policies, 555
General property page
   of the Dirsync Server, 401
   of KMS, 496
   of the routing group connector, 145–146
   of the SMTP Connector, 195–196
General property tab
   for IMAP4, 221
   for NNTP, 209
   for POP3, 222
   of the Connector for Lotus Notes, 406–407
   of the Virtual Server Property box, 217–218
   snap-in, 223–224
generic connectors, 414
geographical name structure, 515
geography-based naming, avoiding for servers, 516
Get a Digital ID property dialog box, 501
Get method
- of IADs, 656
  retrieving the value of property, 658
GetInfo method
- example of, 658–659
  of IADs, 656
GetObject method, 660
Ghost, 454
Global Address List. See GAL (Global Address List)
Global Catalog servers (GCs), 134, 233, 485
Global Domain Identifier fields, 206
Global groups, 331
Global Settings container for an Organization object, 100
Globally Unique Identifiers (GUIDs), 18, 709
Gold tape or disk, 437
Gov domain name, 73
grace periods for conferences, 466
granularity of replication, 133
group policies, application of, 131, 132
Groupnames in MS PC Mail, 38
groups, 91–92, 709. See also mail-enabled groups
  adding members to, 332
  creating, 331–332
  programming in ADSI, 665–667
  VBScript for enumerating members of, 670
  VBScript for querying user membership in, 670
groupware
  as a business demand, 12–13
  early products, 13
GroupWise
  coexisting and migrating, 410–415
  configuring Exchange to connect to, 413
  migrating to Exchange 2000, 413–414
  transferring data to, 411, 412
  transferring messages to Exchange, 411–412
GroupWise API Gateway NLM Version 4.1, 412
Guaraldi, Bob, 6
GUIDs (Globally Unique Identifiers), 18, 709
GWART (Gateway Address Routing Table), 140, 151

H
half-full drive model, 608
handshakes, required for RPCs, 225
hard drives
  choices for, 420
  mirroring, 419–420
  required for an Exchange 2000 server, 418–420
hard limits for individual storage of e-mail, 568–569
hardware
  choosing for Exchange 2000 Server components, 417–427
  preparing for installation, 435–436
  replacing, 569
  requirements for Exchange Server, 418–420
  as a source of problems, 620
Hardware Compatibility List (HCL), 417–418, 709
hardware RAID, 425
hardware requirements for Conferencing Server, 469–470
hardware server platform, 417
hash algorithms, 506
hash, calculating for a message, 504
HCL (Hardware Compatibility List), 417–418, 709
header fields, 45, 84
headers in SMTP, 52
hierarchical naming structure, in DNS, 72
high-level objects, 662–665
historical information, moving to personal folders, 245
“home-burned” CDs, using for copies of software, 436–437
hop count, specifying the maximum, 219
hops, 190
HotSite, 624
Hotsync manager, 258
HTML (Hypertext Markup Language), 172
HTTP (Hypertext Transfer Protocol), 70, 74, 539, 709
  controlling access, 222–225
  as an Internet mail protocol, 266
  in Web Store, 171
  WebDAV as an extension to, 539
HTTP client, 84
HTTP default virtual server objects, 223
HTTP Server property box, 224
HTTP virtual servers in OWA, 310–311
HTTP-based mail, 266
hub and spoke topology for routing groups, 149–150
human resources for the implementation project, 349
hybrid mode, running Exchange 2000 in, 23
HyperSnap DX on the CD-ROM with this book, 699
Hypertext Markup Language (HTML), 172
Hypertext Transfer Protocol. See HTTP (Hypertext Transfer Protocol)
hypothesis, building, 624

I
IADs, 653
IADs Container interface, 660
IADs interface, 656–657
IADsCollection, 653
IADsComputer, 653
IADsContainer, 653
IADsGroup, 653
IADsNamespaces interface, 662
IADsPropertyList, 653
IADsUser, 653
ICANN accredited registration company, 322
ICMP (Internet Control Message Protocol), 291
ID file, preparing for the Connector for Lotus Notes, 405
IDE hard drives, 420
ideal distribution of data and services, 369
identities, creating for newsgroups in Outlook Express, 273–274
IDirectoryObject, 653
IDirectorySearch, 653
IDP, 536
IE Root Store, 500
IEEE (Institute of Electrical and Electronics Engineers), RFCs maintained by, 47–52
IFS (Installable File System), 169–170
IIS (Internet Information Server), 14
integration with, 93
OWA and, 298
Ilmarin, 709
imaging software, 454
IMAP folders, configuring in Outlook Express, 276–277
IMAP servers, states for, 76
administering, 220–221
configuring for Outlook Express, 267
connectivity model, 282–284
Exchange 2000 and, 284–289
SMTP with, 264
standard port numbers for, 285
IMAP4 account, configuring in Outlook Express, 268
IMAP4 client and store in Outlook 2000, 232
IMAP4 clients, 83, 285–289
connectivity model for, 283–284
implementation plan, 341–342
areas for special attention, 364–366
components of, 341
elements of, 347–353
implementation project, identifying the scope of, 348
Import Container tab
of the cc: Mail Connector, 389
of the Connector for Lotus Notes, 407
of the Connector for Novell GroupWise, 413
of the Exchange Dirsync Requestor, 400
Import/Export Wizard in Outlook 2000, 253
inbox repair tool (scanpst.exe), 242
inbox rules, combining with wireless, 595
inboxes, repairing with ScanPST, 571–573
incoming messages, performing reverse DNS on, 220
inconsistent mail delivery, 625
incorrectly addressed messages, 620
incremental backup, 604 indexes
created by Exchange 2000, 171
creating high-speed, 95–96
Indexing Service, 95
information management applications for PIMs, 257
information services
in Outlook profiles, 237
selecting for Outlook 2000, 335
Information Store database in the original Exchange, 14
Information Store event sinks, 94
Information Store event database, 94
information stores, 687
administering, 687–689
distributing among different storage groups, 165
in Exchange 2000, 24–25
network traffic generated by, 254
separating from the logs, 419
utilities for compressing or repairing, 18
wider distribution of, 165
Information Store service, 87
Infrastructure Master, 132–133
inheritance of permissions, 711
in-house human resources, 349
initial view, 244
initialization file in the Team Folders
   Templates, 682
inline error handling in VBScript, 671
in-place upgrade, 370, 433
Installable File System (IFS), 169–170
installation
   automating the process, 452–456
   modes of, 443
   performing a quick, 317–337
   performing a smart, 431–456
   preparing for, 434–441
   scenarios for, 432–434
   summary of preparation required, 442
   verifying the success of, 449–452
installation components, 444–445
Installation dialog box, 319
installation engineer, 344
installation media, 436–437
Installation Type dialog box, 327
installation wizard, 325
instance storage, effect on storage group
design, 165
instant collaboration, 636
instant communication in Exchange 2000, 19
Instant Messaging, 8, 19, 26, 709
Instant Messaging service, 95
Instant Messaging Services installation
cOMPONENT, 445
instant-messaging addresses, 220
Institute of Electrical and Electronics Engineers.
   See IEEE (Institute of Electrical and
   Electronics Engineers)
integrated Windows authentication, 212, 306
Intelligent Drive Electronics hard drives, 420
Intellisync, 257
Intellisync synchronization software, 258
Inter Personal Message (IPM), 40–42, 44, 45
Interchange component. See Connector
   Interchange
Interchange tab of the Connector for MS Mail,
   395–396
interfaces
   exposed by ADSI objects, 652
   provided by ADSI objects, 653
intermediate-size organizations, improving
   performance and scalability, 588–589
internal newsgroups, 266
International Organization for Standards, 53–54
International Telecommunication Union (ITU), 6,
   34, 67
International Telegraph and Telephone
   Consultative Committee (ITU), 34
Internet, 533
   as a backbone for a routing group, 537–538
   client/server access over, 540–545
   as an Exchange server backbone, 533–540
   impact of, 6
Internet business card, 243
Internet client connectivity, 289–293
Internet communication protocols, 7
Internet Connection Wizard, 541–544
Internet Control Message Protocol (ICMP), 291
Internet Corporation for Approved Names and
   Numbers, 322
Internet Directory Server Name dialog box,
   270, 271
Internet directory (LDAP) servers, 271
Internet domain names, 73
Internet E-mail Address screen, 541
Internet Explorer, 700
Internet Explorer 5.x, 298, 299
Internet Information Server. See IIS (Internet
   Information Server)
Internet Mail Access, configuring for
   Outlook 2000, 336
Internet mail client, configuring Outlook on, 236
Internet mail lists, 207
Internet Mail Logon dialog box in
   Outlook Express, 269
Internet Mail Logon screen in the Internet
   Connection Wizard, 542–543
Internet mail services, protocols supported for, 267
Internet Mail, setting up on Outlook Express, 336
Internet Message Access Protocol version 4. See
   IMAP4 (Internet Message Access
   Protocol version 4)
Internet Message Formats General property tab, 214–215
Internet News Group folder, setting the language for, 208
Internet News Service (INS). See NNTP (Network News Transfer Protocol)
Internet Protocol (IP), 69–70
Internet Protocol Security (IPSec), 490–491
Internet protocols, supported by Web Store, 171–172
Internet Security and Acceleration Server 2000, 546
Internet Services Manager
   administering the default Exchange virtual server, 312
   coordinating with Exchange Administration tool, 309
Internet standards, supported by
   Exchange 2000, 27
Internet traffic pattern, 351
Internet Voice Messaging Protocol (VPIM), 27
Internet-Only mode of Outlook 2000, 287
Internetwork layer
   of the TCP/IP protocol stack, 534, 535
Inter-Organizational Connection Agreement, 378
Interpersonal Messaging System (IPMS), 38, 44
Interpersonal Notification (IPN), 44
Interprocess Communication Environment (IPCE), 47
intersite traffic pattern, 351
intranets, private, 6
intraserver traffic pattern, 351
intrasite traffic pattern, 351
invitations to conference resources, 460
I/O boards, adding to servers, 592
IP (Internet Protocol), 69–70
IP addresses, 70
   resolving from hostnames, 71
   restricting, 507–509
IP and TCP/SSL port combinations, assigning for NNTP, 209
IP filtering, 491–492
IP transportation layer, 490
IPCE (Interprocess Communication Environment), 47
IPM (Inter Personal Message), 40–42, 44, 45
IPMS (Interpersonal Messaging System), 38, 44
IPN (Interpersonal Notification), 44
IPSec (Internet Protocol Security), 490–491
Is (exactly) condition, 108
Is not condition, 109
ISA 2000, 546
ISINTEG, 18
ISInteg
   regaining data consistency, 576–578
   running, 577–578
ISO/OSI model, 534, 535
ISPs, understanding the inherent routing of, 520
Item recovery, setting on, 609–610
iterative queries, 127–128
ITT, 34
ITU (International Telecommunication Union), 6, 34, 67

J
JET (Joint Engine Technology) database engine, 123, 161

K
KCC (Knowledge Consistency Checker), 134–135
KDC (Kerberos Key Distribution Center) service, 484, 488, 709
Kerberos protocol with OWA, 306
Kerberos V5 authentication protocol, 484, 487–488, 709
Key Management Service (KMS), 444, 494–498
keys, 707
kiosks, OWA with, 302
KMS (Key Management Service), 444, 494–498
KMS files, 598
KMSerer.pwd, 495
Knowledge Consistency Checker (KCC), 134–135
knowledge management platform, 26
knowledge management storage strategy, 25
knowledge-workers, 8

L
LAPB, 68
large organizations, improving performance and scalability, 589–590
large systems, building, 427–428
late binding objects in an interpretive language, 652
Launch.exe, installing Conferencing Server, 474
layers

- nesting OUs into, 131
- of the OSI model, 34–36, 534
- of the TCP/IP protocol stack, 534

Layout tools in Outlook 2000, 641


- clients, 83
- connecting to X.500-based directories, 266
- example program, 66
- provider, 655, 665
- Search command with filtering, 62–63
- service as an address book, 233–234
- session syntax, 60

LDAP server connection, configuring in Outlook Express, 270, 271

LDAPMessage, 59–60

LDAPResult syntax, 61–62

leaf objects in a DIT, 56

“least cost” route, 151

legacy environments, 592–593

legacy geographical and hierarchical name structures, 515

legacy messaging systems, truncation in, 517

legacy system migration, 518–520

legacy systems, special requirements of, 520

License code, filling in, 443

Licensing Agreement dialog box, 328

licensing, required for installation, 441

“light” messaging client, OWA scenario for, 301


Link and Server monitors, 183, 562

link failures, communicating, 152

Link Monitor Notifications, 183–184

Link Monitors, 95, 183–184, 618

Link State Information, exchange of, 151–153

link state routing

- algorithm, 190
- finding alternative paths, 86
- model, 530

links, monitoring the state of, 183–184

list servers with information about Exchange, 622

Listds.exe utility, 403

lists, expanding in SMTP, 49

LMHOST files, 125

load balancing

- facilitated by front- and back-end servers, 83
- with OWA, 304–305

load simulator, 429

LoadSim, 586

LOADSIM.exe, 429

local offline address book, 106

Local Post Office tab of the Connector for MS Mail, 396

Local Security Authority (LSA), 484–485, 709

local support liaison for the rollout, 360

local support staff for the rollout, 360

local system security, 493

lock box, 503

log files, 598, 599–600

- for Conferencing Server, 481
- for database transactions, 163
- deleting manually, 600
- viewing, 582–583

logging

- configuring for POP3 and IMAP4, 292
- enabling, 217
- in Web Store, 172

logic, applying to problems, 623

logical aspects of a messaging environment, 557

logical database in Information Store, 87

logical drives, 418

logical system, designing for a collaborative solution, 639

logon, sites and, 136

logout state for an IMAP4 server, 76

logs

- searching for, 617
- separating from the information store, 419

lost information, restoring, 595

Lotus cc: Mail. See cc: Mail

Lotus cc: Mail Migration Directory Toolkit, 391

Lotus Notes

- coexistence with and migration from, 404–410
- configuring a connection to, 405–408
- data transfer with Exchange, 405

LPADC tool, 586

LSA (Local Security Authority), 484–485, 709

LSA server service, 484

lsmexin.exe, 404–405

lsmexnts.exe, 404–405

lsmexout.exe, 404

lsntsmex.exe, 404
M

M drive, 169–170, 449
macros, security levels for, 510–511
MADCAP (Multicast Address Dynamic Client
Allocation Protocol), 465, 710
MADCAP server, 464
Mail Account Properties screen, 544–545
MAIL command in SMTP, 47, 51
mail connections, configuring in Outlook Express,
267–270
mail delivery, inconsistent, 625
Mail Exchanger records, 127
mail, queuing for remote-triggered delivery, 196
mail systems, coexistence with other, 441
Mailbox limits, maintaining, 186
mailbox names, 517
Mailbox (Private) Server, 521
MailBox Statistics tool, 586
MailBox Store Finder tool, 586
mailbox stores, 87, 687
creating, 164
deleting, 220
mailbox-enabled users, 106, 110–111
mailboxes, 709
accessing, 625–626
adding in Exchange 2000, 23
cleaning up, 568–569
creating for a conferencing site, 479
in SMTP, 50
upgrading, 371
mail-enabled contacts, 110. See also contacts
mail-enabled groups, 110
mail-enabled public folders, 106
mail-enabled users, 106
MailQ tool, 586
Mainframe e-mail models, 4–5
Management Domains (MDs), 39
Managing Exchange Conferencing Server role, 471
Managing resources role, 472
mandatory attributes, 673
many-to-many interaction, 8
MAPI, 12, 709
MAPI client, 83
MAPI store (.edb), 87
MAPISend tool, 586
Masquerade domain, text box for, 219
master-slave configuration for NNTP, 208, 539
Maximize MS Mail 3.x Compatibility option, 395
maximum videoconference extension time, 465
MCUs (Multipoint Control Units), 462–463, 477
configurations for, 463
in data conferences, 464
installing, 478
MD-4 hash algorithm, 506
MD-5 hash algorithm, 506
MDBEF (Message Database Encapsulated Format),
166, 205
Mdbvue32, 584
MDs (Management Domains), 39
members, adding to a group, 332
memory
managing, 590–591
requirements for an Exchange 2000 server,
420–421
standard specified for, 556
mesh topology for routing groups, 150
message body, 84
Message Database Encapsulated Format (MDBEF),
166, 205
Message delivery object in the Global Settings
container, 215
message digest, 504
Message Encoding property tab for POP3, 222
message flags, setting for newsgroups, 273
message flow, problems in an Exchange enterprise
layout, 626–627
Message Format property tab for IMAP4, 221
Message Handling Environment (MHE), 37
Message Handling System (MHS), 37
message header, 84
Message History dialog box, 184, 186
message log files, 159
Message Options page, 503–504
message processing between Exchange and
GroupWise, 411–412
message queues
managing with SMTP, 141
monitoring, 157–159
reviewing and repairing with MTACheck,
580–582
viewing, 159
message restrictions, setting on the routing group
connector, 147
Message Rights permission category for public folders, 114
message routing, 117
  administrative models and, 17
  in Exchange 2000, 537–539
  fine tuning, 530
  model, 514
  to servers in other routing groups, 156–157
  within a single server, 153–154
  between two servers in a routing group, 154–155
Message Store (MS), 37, 39, 41
message store providers in Outlook 2000, 230–232
message synchronization in Outlook Express, 276–277
message tracking, 184–186, 615
Message Tracking Center properties page, 184, 185
message tracking files, 598
Message Transfer Agents. See MTAs (Message Transfer Agents)
Message Transfer System (MTS), 37, 39, 41
message transport, 139
  changes since Exchange 5.5, 139–140
  managing, 157–159
messages. See also e-mail
  delivering between different routing groups, 538
  delivering in the same routing group, 537
  delivering outside the organization, 538
  delivering to personal folders, 250–251
  digitally signing, 504–505
  digitally signing in POP3 or IMAP applications, 295
  digitally signing or encrypting, 275
  encrypting, 295, 503–504
  filtering, 94
  incorrectly addressed, 620
  routing on the same server, 537
  securing the content of, 510
  sending an original copy of, 618
  setting the maximum size of outgoing and incoming, 215
  transferring between servers in different routing groups, 117
  types supported by Exchange clients, 84
  Messages property tab of the Virtual Server Property box, 218
message-tracking center, 159
Messaging and Collaboration Services installation component, 444
messaging and directory standards, using characters compatible with, 516
Messaging Application Programming Interface (MAPI), 12, 709
Messaging Application Programming Interface client. See MAPI client
messaging protocol for Exchange 2000, 19
messaging standards, 6, 7
messaging systems. See also legacy messaging systems
  coexistence with other, 379–380
messaging transfer schedule for a routing group connector, 146–147
methods of interfaces, 652
MHE (Message Handling Environment), 37
MHS (Message Handling System), 37
Microsoft Conferencing server, 95
Microsoft Exchange 5.5 Administrator, 445
Microsoft Exchange 2000 Resource Kit, 585–586
Microsoft Exchange cc:Mail Connector. See cc: Mail Connector
Microsoft Exchange Chat Service installation component, 19, 445
Microsoft Exchange Connector for Lotus Notes. See Connector for Lotus Notes
Microsoft Exchange Connector for Novell GroupWise. See Connector for Novell GroupWise
Microsoft Exchange Directory Synchronization (DXA.exe), 399
Microsoft Exchange Instant Messaging Services installation component, 445
Microsoft Exchange Key Management Service, 444, 494–498
Microsoft Exchange Messaging and Collaboration Services, 444
Microsoft Exchange MSMail Connector, 444
Microsoft Exchange Services screen, 177
Microsoft Exchange System Management Tools, 445
Microsoft Exchange Web site, 615
Microsoft Foundation model, 341
Microsoft Hardware Compatibility List (HCL), 417–418, 709
Microsoft IIS Log File Format, 217
Microsoft Internet Explorer on the CD-ROM with this book, 699
Microsoft Mail. See MS Mail
Microsoft Mail for Personal Computing Networking. See MS PC Mail
Microsoft Management Console (MMC), 16, 22–23
Microsoft Meta Directory Service, 528
Microsoft Office, 14
Microsoft Outlook Permissions Control, 680
Microsoft Outlook View Control, 680
Microsoft Premier Support, 620–621
Microsoft Web site, checking for software updates and notices, 567
migration, OWA use during, 302
migration process, testing, 383
migration tools
  for cc: Mail, 391–392
  in Exchange 2000, 24
Migration Wizard, 381–383, 391–392, 403–404
  migrating from Lotus Notes to Exchange, 408–410
  performing GroupWise migrations, 413–414
milestones, setting, 364
MIME format, storing Streaming Internet content, 167
MIME-like message in HTTP, 74
minimal server-hardware profile, 418
Minimum installation mode, 443
mission statement in an implementation plan, 341
mixed Exchange environment, 592–593
mixed mode, 710
  design restrictions imposed by, 529
  versus native mode, 378–379
running in, 514
MLP (Multilink Protocol), 68
MMC (Microsoft Management Console), 16, 22–23
MMC snap-ins. See snap-ins
mnemonic address fields in English, 42–43
mnemonic address in X.400, 41
model for a business organization, 514–515
moderated group server, setting for NNTP, 210
modifications during rollout, 363
ModifyRDNRequest command in LDAP, 65
ModifyRequest syntax in LDAP, 64
modular services in Outlook Express, 262–266
modulo 8.128 window sizes, 68
monitoring
  post-installation, 362–363
  stopping all without deleting, 177
Monitoring panel, 176, 178
monitoring server, 180–181
monitoring utilities, 95
monitors, 176–180
monthly and annual tasks, 568–570
motherboards, multiprocessor, 421–422
Movehere method, 660
MS (Message Store), 37, 39, 41
MS Mail
  accessing legacy e-mail data, 232
  address type, 403
  coexistence with and migration from, 392–404
  connecting Exchange 2000 to, 395–398
  data transfer with Exchange, 393–394
  Dirsync, 398
  GAL, 398
  migrating to Exchange 2000, 403–404
  resetting synchronization numbers, 403
MS Mail POs
  ascertaining the health of, 395
  configuring to recognize the Exchange Connector PO, 398
MS PC Mail, 11–13
  16-bit nature of, 12
  compared to Exchange, 14
MSF (Microsoft Foundation) model, 341
MSMail Connector as an installation component, 444
MTA service, 142, 580, 581
MTA transport protocols, 199–201
MTA values, overriding for a specific server, 203
MTAcheck, 18
  command switches allowed in, 581–582
  reviewing and repairing queues, 580–582
MTAcheck.log file, 580
MTADATA subdirectory, 598
MTAs (Message Transfer Agents), 19, 37, 139, 580, 709
  causing to take turns sending, 205
  in an MTS, 39, 41
MTAView tool, 586
MTS (Message Transfer System), 37, 39, 41
Multicast Address Dynamic Client Allocation Protocol (MADCAP), 465, 710
multicast connection to a conference, 461
multicast group IP address, 465
multicast videoconference, 465
multilingual capabilities of Exchange 2000, 28
multilingual support in OWA, 313
Multilink Protocol (MLP), 68
multiple administrative groups, 101, 102
Multiple Authentication Provider, 485
multiple databases in a storage group, 25
multiple message databases in Enterprise server, 87
multiple-server environment, OWA in, 304
multiprocessor motherboards, 421–422
Multipurpose Internet Mail Extensions. See MIME format
mutual authentication, using Kerberos, 487–488
MX records, 127

N
name resolution
   checking with ping, 291
   in DNS, 73
   as a source of problems, 619
name server in DNS, 73
Name Service Provider Interface (NSPI), 485
named URLs as a primary access method, 300
names
   verifying in SMTP, 49
   in X.500, 56
namespace, binding to, 654
Namespaces container, programming with ADSI, 662–663
naming conventions
   for Exchange 2000 entities, 440
   reviewing and updating, 515–517
   suggestions for establishing, 516–517
   in X.500, 56–59
native content store, 89, 710
native mode, 710
   converting to, 99
   versus mixed mode, 378–379
   running Exchange 2000 in, 23
   switching to, 379
NCSA Common Log File Format, 217
NDRs (Non-Delivery Reports), 618, 710
   enabling NNTP to send, 210
   specifying recipients of, 218
NDS provider, 655
needs-based training plan, 552
Net domain name, 73
Net Logon service, 484, 710
NET2STRM tool, 586
NetBIOS, reliance of Windows NT 4.0 on, 125
NetMeeting
   as a Web part, 691
   whisper feature, 464
Netscape Communicator, 289, 699, 700
Netscape Messenger
   as an Internet mail client for POP3 and IMAP4, 289
   location of configuration settings, 290
NetWare, OSI model and, 36
network access, controlling, 485
Network Access layer of the TCP/IP protocol stack, 534, 535
network auto-detection in Outlook Express, 275
Network Courier, acquisition of, 12
network interconnection, handled by the OSI model, 35
network interface cards (NICs), 426–427
network layer of the OSI model, 35, 36, 534, 535, 711
Network Load Balancing, 305
Network News Transfer Protocol. See NNTP (Network News Transfer Protocol)
network security, OWA and, 305–308
network shares, created during setup, 450
network traffic
   analyzing after installation, 362
   created by IMAP4, 76
   created by POP3, 75
network transports, 199–201
network utilization in Outlook Express, 277–278
New Object–User dialog box, 330
New SMTP Virtual Server Wizard, 216
Newnews command, 207
newsfeeds, 213, 539
newsgroup messages, 273
newsgroups, 266, 539, 710
   configuring in Outlook Express, 272–274
   creating on the master server, 208–212
creating on the slave servers, 212–213
with information about Exchange, 622
listing of, 703
subscribing to, in Outlook Express, 272–273

Newsgroups command, 207

NICs (network interface cards), 426–427

NNTP (Network News Transfer Protocol), 70, 74, 207–214, 537, 539, 710

NNTP client, 84

NNTP News Service servers, 630

NNTP service
  failing to load, 325
  selecting during Windows 2000 installation, 319
  nodes, 72
    in a cluster, 170
    in the DNS hierarchy, 126
non-authenticated state for an IMAP4 server, 76

Non-Delivery Reports. See NDRs (Non-Delivery Reports)

non-leaf objects in a DIT, 56

non-receipt notifications (NRs), 44, 50
normal backup, 604

Not present condition for building filter rules, 109

Notes. See Lotus Notes

Notes client software, 406

Notification object container, 563

notifications, 180–182

Novell file servers, configuring network connectivity to, 387

Novell GroupWise. See GroupWise

NRs (non-receipt notifications), 44, 50

NSPI (Name Service Provider Interface), 485

NT 4. See Windows NT

NT Challenge/response, 212

NT Challenge/Response authentication method, 293–294, 484, 487, 710

NTBackup.exe, versions of, 596

NTDSUtil, 18, 574

NTFS, 710

NTGateway group, 413

NTLM authentication method, 70, 293–294, 484, 487

NTLM protocol, 306

numeric address in X.400, 41

NWCOMPAT provider, 655

0

OAB (Offline Address Book), 96, 234, 710
downloading, 247
installing, 251
rebuilding of, 247
.oab extension, 106

object types for ADC replication, 377

Object Viewer dialog box in Active Directory Browser, 677

objectives
  for implementation, 354–355
  in an implementation plan, 341

ObjectRenderer object, 646

objects, retrieving and setting properties in ADSI, 656

ODBC Logging, 217

Office Assistant in Outlook 2000, 252

Offline Address Book. See OAB (Offline Address Book)

offline address lists, 106
offline folders
  enabling, 238, 246–248
  fixing corrupt, 242
  storing data in, 242
  synchronizing, 246–247

Offline Folder Settings dialog box, 248
offline messages, encryption and, 507
offline repair, free storage space required for, 436

Offline Storage Files (*.OST), 710

offline support in Outlook Express, 276–277

off-site support for the rollout, 360

Oltfwiz.exe, 681

On Error Resume Next statement, 663, 671

on-demand tasks, 564–565

one-step migration by the Migration Wizard, 382

one-to-many interaction, 8

one-way connection agreements, 372–373

ongoing support, documentation of, 551–552

ongoing support plan, 552

online conferencing applications, 458

Online Knowledge Base, 621–622

on-site support for the rollout, 360

Open Address Lists permission, 109

Open Shortest Path First (OSPF) method, 151

Open System Interconnect model. See OSI model

Open System Interconnect standard, 711
Open System Interface reference model, 34–36
operating software for PIMs, 257
operation mode, displaying, 99
optional attributes, 673
O/R addresses, 41
Org domain name, 73
organization, creating a new, 446
organization forms library, 689
organization name
  entering, 447
  selecting, 326, 327
  specifying to a server, 629
Organization Name dialog box, 327
organization object, 99–100, 439
Organization options, choosing during installation, 446
Organizational Units. See OUs (Organizational Units)
Originator/Recipient (O/R) addresses, 41
OS/2, limitations of, 13
OSI layers, TCP/IP stack and, 70–71
OSI model
  numbering of protocols in, 69
  X.25 and, 67
OSI protocol stack, 35
OSI reference model, 34–36
OSI (Open System Interconnect) standard, 711
.OST extension, 242
OST files, 598
  implementing usage of, 592
  repairing damaged, 571
OST Integrity check tool, 242
OUs (Organizational Units), 131
  creating in ADSI, 664–665
  nesting into layers, 131
Out of Office Assistant in Outlook 2000, 253
Outbound connections button, 219
outbound e-mail retry, 219
Outbound security button, 219
Outlook 2000, 229–230
  bandwidth profile, 254–255
  calendaring, 458, 466
  components of, 230–235
  configuring, 235–236
  configuring as an Exchange client, 236–251
  configuring for remote use, 245–251
developing collaborative solutions, 639–644
enabling users to schedule conference resources, 479
Import/Export Wizard, 391
installing, 334–336
as an Internet mail client for POP3 and IMAP4, 287
limitations of OWA compared to, 300–301
location of configuration settings, 290
network traffic generated by starting up, 254
network utilization compared to Outlook Express, 278
options for customizing, 242–245
processes utilizing server resources, 254–255
reserving conference resources, 460
security highlights, 510–511
Startup Wizard, 334
synchronizing with PIMs, 256, 257–258
utilities in, 251–254
workgroup installation of, 83
Outlook Address Book, 234–235
Outlook calendar, support of Conferencing Server, 466
Outlook Express, 261
  components and features of, 261–266
  configuring, 267–270
  features of, 262
  installing, 266, 336–337
  location of configuration settings, 290
  offline support and message synchronization, 276–277
  remote-friendly features of, 275–277
  as an SMTP client, 83
Outlook Form Designer, 639–640
Outlook profiles, 237–239
Outlook Today, running Digital Dashboard samples in, 695
Outlook views, 244–245
“over the network” installation, 437
OWA (Outlook Web Access), 74, 297, 545, 711
  accessing My Calendar on the Exchange server, 298, 299
  administering, 686
  authentication options for, 305–308
  business scenarios for, 301–302
  configuring, 309–310
Exchange 2000 directories pertinent to, 308
features in Exchange 2000, 299–300
limitations compared to Outlook 2000, 300–301
in a multiple-server environment, 304
network utilization compared to Outlook Express, 278
new features in, 646, 686
old and new versions of, 297–301
preserving Exchange 5.5 customization, 303
setting up, 308–310
in a single-server environment, 303
upgrading to Exchange 2000, 303
virtual IIS directories used by, 308

Performance Optimizer, suggestions on when to run, 565
“period of volatility,” 353, 562
permanent virtual circuits (PVC), 68
permission assignments, granularity of, 527
permissions
assigning to public folders, 114–116
for Conferencing Server roles, 471–472
displaying for objects, 99
effect on topology and design, 526–527
for Exchange 2000 administrator roles, 509–510
granting, 102
inheritance of, 120, 711
managing by creating administrative groups, 101
managing with administrative groups, 137–138
for public folders, 115–116
setting for compatibility with pre-Windows 2000 servers, 321
as a source of problems, 619
specifying for address lists, 109
stored by ADS, 22
types of, 119
Windows 2000 needed to restore Exchange, 607
persistent finds, network traffic generated by, 255

Personal Address Book. See PAB (Personal Address Book)
Personal Folder File (*.PST), 711
personal folders, 235
migrating data to, 382
moving historical information to, 245
offline networking with, 250–251
storing data in, 240–241
personal folders file in the Team Folders Templates, 681
Personal Information Managers (PIMs), 256–259
personal store, 235
physical and logical designs, separation of, 23
physical constraints, implementation scheduling and, 350
physical design of collaborative solutions, 639
physical hard drives, 418. See also hard drives
physical layer of the OSI model, 35, 36, 534, 535, 711
physical level, standardization at, 555–556
physical network topology, 23
physical placement of devices and services, 514
physical profile of an organization, 556
pilot environment, 434
pilot phase of implementation, 357–359
PIMs (Personal Information Managers), 256–259
Pine
   as an Internet mail client for POP3 and IMAP4, 288–289
location of configuration settings, 290
Ping, checking name resolution and network connectivity, 291
"plain text" password, 293
planning and design, documentation of, 550–551
planning, process of, 340–341
planning team, 345
plans, components of, 550–551
Play LAN
   installing, 317–337
   logical and physical view of, 318
PO (Post Office) in cc: Mail, 383
PocketMirror, 257
PocketMirror synchronization software, 258
PODIAg tools, 395
pointers in DNS, 126
policies
   applying, 104
   effect on topology and design, 526–527
   implementing behavior based on designs, 527
   for usage and administration, 531
policy property sheets, 104
politics, identifying barriers and obstacles to implementation, 350
pools of valid security identifiers, 132
POP3 (Post Office Protocol version 3), 74–75, 263, 264, 265, 536, 711
   administering, 222
   compared to IMAP4, 709
   configuring for Outlook Express, 267
   connectivity model, 281–282, 283
   connector, 75
   Exchange 2000 and, 284–289
   network profile, 296
   SMTP with, 263
   standard port numbers for, 285
   support provided by messaging stores, 232
port 80
   for HTTP, 74
   opening and saving files, 89
port 110, 75, 285
port 119, 74
port 143, 76, 285
port 389, 59
port 563, 209
port 993, 285
port 995, 285
port numbers, configuring in Outlook Express, 270
Post Office (PO), 383
Post Office characteristics of MS Mail, 395
Post Office tab of the cc: Mail Connector, 387–388
postal address in X.400, 41
posting information on newsgroups, 273
post-installation
   monitoring and feedback, 362–364
   tasks for Conferencing Server, 478–479
power supplies, redundant, 427
pre-defined Web parts, 690
Premier Support, 620–621
Present condition for building filter rules, 109
Presentation Access Point (PSAP) in TP4, 69
presentation layer of the OSI model, 35, 36, 534, 535, 711
previous logs, 600
primary agreement, 374
Primary Management Domain (PRMD), 39, 40
primary migration file, 381
private conferences, 459, 467
Private DMD (PRDMD) in X.500, 57–58
Private Information Store, 14
Private Information Store property page, 609–610
private intranets, 6
private messaging database (Priv.edb), 161
PRMD (Primary Management Domain), 39, 40
problems
   common causes of, 619–620
   dealing with unexpected during the rollout, 361
   getting a handle on, 614–616
   solving, 625–629
problem-solving method, 624
processors
  benefit of multiple, 421–422
  for an Exchange 2000 server, 421
product compatibility engineer, 343–344
product ID. See CD key
Progen.exe, 238
Profile dialog box, 238
Profile Generation tool, 586
Profile property page, 239
profiles for Outlook, 237–239
PROFS connector, upgrading, 371
programmatic platform components, 94–95
programmatic stubs, 225
project manager, 343
project objectives for implementation, 354–355
project overview document, 355
project support engineers, 345
“Prompt for a profile to be used” radio button, 239
properties
  getting and setting, 656–659
  VBScript for reading the value of, 658
Property dialog box of the Organization object, 99
property value, setting in ADSI, 659
proprietary e-mail systems, 3–4
proprietary extensions of standards, 34
protocol analyzers, 623
protocol event sinks, 94
protocol logging, 292–293
protocol stack for TCP/IP, 534–536
protocols
  administrating, 207–225
  in the TCP/IP stack, 536
Protocols snap-in, 93
providers, shipped with ADSI, 655
proxy address generator .dll file, 381
PST files, 235, 598
  Best encryption for, 256
  encrypted, 572
  implementing usage of, 592
  placement of, 571
  repairing damaged, 571
  storing data in, 240–241
pst19upgd.exe, 241, 572
public area access computers, OWA with, 302
public conferences, 459, 467
public folder access, 143
public folder hierarchy, 688
Public folder Properties page, 332
public folder referrals, 117, 202
public folders, 712
  address list, 106
  administrating, 112–117
  allowing access to, 74
  categories of permissions for, 114
  configuring with replicas on multiple servers, 167
  creating, 113, 332
  establishing standards for naming and organizing, 529
  features of, 112
  mail-enabling, 114
  migrating to Exchange 2000, 371
  permission available for, 115–116
  permissions assigned to, 114–116
  putting policies in place to deal with, 565
  replication of components, 116–117
  storage required for, 424
  synchronizing with offline folders, 246
Public Folder Server, 521
public folder shortcuts. See virtual directories
public folder stores, See virtual directories
public folder trees
  configuring, 113–114
  hosting multiple, 167
Public Information Store, 14
public messaging database (Pub.edb), 161
Public Service Switched Network, connecting DTE and DCE, 67
public store, configuring settings for, 114
Public Store Properties dialog box, 114
public switched networks, 68
public-with-password conference, 459
pull replication, 116
purpose-made accounts, 438
push replication, 116
Put method of IADs, 656
Q
quality environment, producing, 559
queue monitoring, setting up ongoing, 157–159
queues. See message queues
quick installation, 317–337
RAID 0/1, 425
RAID 1, 419–420, 425
RAID 5, 425
RAID disk storage, 425
RAS (Remote Access Service), 276
RAS X.400 connector, 206
RC2-40 encryption algorithm, 506
RC2-64 encryption algorithm, 506
RC2-128 encryption algorithm, 506
RCPT command in SMTP, 47, 51
RDNs (Relative Distinguished Names), 56, 57
real-time collaboration management, effect on design and topology, 526–527
real-time communication in Exchange 2000, 26
real-time monitors, 562, 614
real-time tasks, 562–564
Receipt Notifications (RNs), 44
receiver-SMTP, 47, 48
Recipient objects, 106, 110–111
  configuring, 112
  creating, 110–111
recipient policies, building with OUs, 131
Recipient Update Service (RUS), 109–110
recipients, 96, 100, 110–112
  listing of all, 106
  managing, 674
  selecting for notifications, 181
VBScript for creating, 674
Recipients container for an Organization object, 100
record keeping for tasks, 570
recovery
  calculating the time required for, 608
  of offline folders, 242
  speeding up with a dedicated system, 607–608
recursive queries, 127
Red Book for the X.400 standard, 34
redundant power supplies, 427
reference collaborative solutions, 636
referrals
  to public folders, 112
  in X.500, 58
regional hub topology, 149
Registered Private Operating Agencies (RPOAs), 39
registration parameters, associated with event sinks, 697
registry entry, enabling NDRs in NNTP, 210
registry keys, adding for connectors, 381
relational database storage, compared to Web Store, 25
Relative Distinguished Names (RDNs), 56, 57
relay restrictions on the Access property tab, 218
reliability of Web Store, 172
Remote Access Service (RAS), 276
Remote Bridgehead property page of the routing group connector, 146, 194
Remote Dirsync requestors
  adding MS Mail POs as, 402
  property pages of, 402
Remote Mail, 248–250
Remote Mail tab of the Exchange Server dialog box, 249
remote networking, configuring Outlook for, 245–251
Remote Procedure Calls. See RPCs (Remote Procedure Calls)
remote Routing Group Connectors, 192
remote routing group, creating a connector in, 148
remote server, specifying the name of, 194
remote users
  security and firewall considerations, 255–256
  wireless e-mail for, 594–595
remote-friendly features of Outlook Express, 275–277
remote-triggered delivery, queuing mail for, 196
RenderingApplication object, 646
RepAdmin, 584–585
replicas of public folders, 116
replication
  of public folder components, 116–117
  of public folders, 167–169
  verifying, 568
replication admin tool, 585
replication architecture, improvements in, 133
replication bridgehead server, 135
replication connections, viewing, monitoring, and troubleshooting, 584
replication monitor tool, 584–585
replication monitor utility, 618
replication object database, 639
Replication Properties dialog box, 167–168
replication Schedule dialog box, 168
replication schedule for a CA, 377–378
Index ♦ R

Replication Status dialog box, 168–169
replies in SMTP, 52
Replmon, 136, 584–585, 618
reports, logging for Conferencing Server, 480
"request-response" method of communication in HTTP, 74
res1.log, 600
res2.log, 600
reserve databases, 605
reserved logs, 600
resolver in the DNS resolver model, 71–72
Resource Kit, 585–586
resource kits, 703
resource model in Conferencing Server, 459
Resource Records (RR), 127
resources
assigning to the rollout, 361
monitoring, 177–178
scheduling for conferences, 460
responsibilities in the implementation project, 345–347
restores, verifying, 606
restoring, 604–612
retention rate for e-mail, 424
reverse lookup zone, 625
reverse path in SMTP, 50
revocation options for certificates, 501, 502
RFC 791, 199
RFC 793, 199
RFC 821, 47–52
RFC 822, 52, 536
RFC 850, 207
RFC 1035, 537
RFC 1730, 75
RFC 1823, 66
RFC 1869, 51
RFC 1945, 74
RFC 2065, 74
RGC (Routing Group Connector). See routing group connectors
RGM (Routing Group Master), 151–152
rich-text files, support for storing, 166
RID Pool Master, 132
risk assessment table, 352, 353
risk factors in messaging environments, 350
risk management during planning and implementation, 350–353
risks, minimizing, 352–353
RNs (Receipt Notifications), 44
roles. See also administrative roles
   administrator, 509–510
   in the implementation project, 342–345
rollback procedures in the pilot phase, 359
rollout
documenting each stage of, 361
phases of, 359–361
procedures, 357
root
in ADSI, 659
in a DIT, 56
in the DNS hierarchy, 126
root level stores, 298
root object, VBScript for binding to, 661
root servers, contacting in DNS, 73
Root Store, adding certificates to, 500
round-robin DNS, OWA and, 304
routes
   selecting lowest cost, 190
   in SMTP, 50
routing and tracking collaborative solutions, 636
routing architecture, changes in, 140
routing connectors, 538
Routing Engine service, 89, 712
Routing Group Connector General property page, 192–193
routing group connectors, 118, 189, 538, 712. See also connectors; enterprise connectors
building, 191–195
compared to X.400, 520
creating, 145–148
Routing Group Master (RGM), 151–152
routing groups, 91–92, 142–143, 537, 712
creating, 117
creating and administering, 143–144
delivering messages between different, 538
displaying, 99, 192
linking, 144–148
logically grouping servers into, 140
planning, 523–524
renaming, 144
routing to servers in other, 156–157
routing within, 154–155
topologies for, 149–150
routing in a WAN, 47
routing scenarios, 153–157
routing topology, reviewing for MS Mail, 394
roving users, supporting with OWA, 302
RPC encryption, turning on, 256
RPCPing utility
testing RPC communication, 578–580
versions of, 579
RPCs (Remote Procedure Calls), 225, 578
RPingc client-side utility, 579–580
RPings server-side utility, 578–579
RPOAs (Registered Private Operating Agencies), 39
RR (Resource Records), 127
RRAS
as a network transport, 199
using with VPN, 295
RRAS connectivity, separate machine for, 437
RTS values on the X.400 TCP Connector Override property page, 203–204
rules
applying to forms with VBScript code, 644
testing, 252
Rules Wizard in Outlook 2000, 251–252
RUS (Recipient Update Service), 109–110

S
SA (System Attendant), 89, 175, 712
background process activities of, 186
configuring to monitor local or remote Exchange servers, 175–176
SA mailbox, manually removing messages from, 584
SACLs (System Access Control List), 484
SAM (Security Accounts Manager), 123, 485, 712
SAM database, replicating, 132
SAM directory, 123
scalability
with front-end/back-end architecture, 428
in intermediate-size organizations, 588–589
in the smallest organizations, 588
of Web Store, 172
scandisk, 242
ScanPST
caveats about, 572, 573
repairing inboxes, 571–573
running, 572–573
scanspent.exe, 242
scanreg.exe utility, 242
SCD files, 598
Schedule dialog box for replication, 168
Schedule tab
for a CA, 377–378
of the Dirsync Server, 402
of the Exchange Dirsync Requestor, 400
scheduling
of the implementation project, 350
laws of, 343, 350
the rollout, 360–361
tasks, 565–570
schema, 712
changing by adding Exchange 2000 to Active Directory, 329
updating the Active Directory, 448, 449
Schema Admins group, 129
schema admin tool, 585
schema class object, retrieving a user’s properties, 668–669
Schema Master, 132
schema partition, 90
schema property of an ADSI domain object, 663
scope creep, managing against, 348
scope of the implementation project, 348
scratch disk space, required for backup verification, 568
screen capture software, using for documentation, 553
script notifications, 182
scripts, 598, 638
SCSI hard drives, 420
sealing messages, 503–504
seamless access to conferences, 459
Search command with filtering in LDAP, 62–63
search criteria for the find utility, 254
Search response syntax in LDAP, 64
secondary agreements, 374
secondary migration file, 381–382
Secure Communication button, 218
Secure Domain Controller security template, 486–487
Secure e-mail area of the Security property page, 500–501
Secure Password Authentication (SPA), 277, 278, 293
secure ports for Outlook Express services, 269
secure server policy for IP security, 491
Secure Sockets Layer. See SSL (Secure Sockets Layer)

security
- accounts required for installation, 438
- centralized method of defining, 493
- considerations for Outlook data, 256
- context identified by Windows 2000 Server, 94
- descriptors, 484, 485
- features in Exchange 2000, 494–507
- groups, 105, 331–332
- as a layered process, 483
- levels for conferences, 459
- levels in Conferencing Server, 29
- log, 616
- managing, 697–698
- of message information, 294–295
- in Outlook Express, 274–275
- provided by Conferencing Server, 467
- for PST files, 241
- SMTP and, 47
- types included in Exchange 2000, 93–94
- in Web Store, 173

Security Accounts Manager (SAM), 123, 485, 712

Security Configuration and Analysis snap-in, 492–493

Security Configuration Tool Set, 492–493

Security Log in Event Viewer, 486

Security page, enabling in Organization and Administrative Groups objects, 100

Security property page, 500–501

Security Service Provider Interface (SSPI), 488

Security Templates snap-in, 493

security-related events, auditing, 486–487

selected state for an IMAP4 server, 76

sender-SMTP, 47, 48

serial number for a Connector PO, 393

server activity, monitoring after installation, 362

Server Edition of Exchange 2000, 20–21

server membership for each routing group, 144

Server Monitors, 95, 175–176, 618

server policy for IP security, 491

server roles in Windows 2000 enterprise, 132–134

servers
- assigning roles to, 629–630
- for Conferencing Server, 469
- installing a second, 333
- logical grouping of, 140

moving between routing groups, 144

naming, 516

new roles for, 521–523

overestimating usage purposely, 518–519

physical placement of, 518–519

requirements for Conferencing Server, 470

routing messages to other routing groups, 156–157

routing within a single, 153–154

running out of disk space, 619

stopping and restarting, 181

Servers container for an Organization object, 100

server-side collaboration solutions, 644

service account
- creating, 438
- specifying for the ADC, 375
- specifying to a server, 629

service levels
- implementing reasonable, 352
- targeted in the ongoing support plan, 552

service packs
- installing Windows 2000, 437
- integrating Windows 2000, 453

Service Principal Names (SPNs), 488

services, added during installation, 450, 451

Session Access Point (SSAP) in TP4, 69

session encryption, 294–295

session layer of the OSI model, 35, 36, 534, 535, 711

session token. See access token

SetInfo method
- calling to commit property changes, 659
- of IADs, 656

Setspn utility, 488

Settings property tab for NNTP, 210, 211

Settings tab of the Exchange Dirsync Requestor, 400

setup log, 617, 712

Setup Manager utility, 453

setup options, 442–443

Setup Type screen in the Installation Wizard for Conferencing Server, 476

setup.exe, initiating an installation, 442

SHA-1 hash algorithm, 506

Shadow PO, 393

shared applications in data conferences, 463

shared clipboard in data conferences, 464

shared-key encryption, 506
SIDs (Security Identifiers), 484
signatures
  creating and editing in Outlook 2000, 243
  creating for newsgroups in Outlook
  Express, 274
  verifying, 505–506
Simple Mail Transfer Protocol. See SMTP (Simple
Mail Transfer Protocol)
single administrative group, 101, 102
"single instance storage," 165
single-server OWA environment, 303, 305
site boundaries, assigning for Exchange 2000,
527–528
Site connector in previous versions of
Exchange, 145
site links, 135–136
Site Replication Service (SRS), 372, 712
site services account in Exchange 5.5, 438
sites, 134–136
  in Exchange 5.5, 140, 516, 527
  first server in, 629–630
  in previous versions of Exchange, 17
  replication within, 135
  in Windows 2000/Exchange 2000, 527
Sites and Services MMC snap-in, 134
slave servers
  creating newsgroups on, 212–213
  distributing the user load across NNTP, 208
slow-speed links, 591–592
SLV files, 598
Small Computer System Interface hard drives, 420
smallest organizations, improving performance and
scalability, 588
Smart Host designation, 220
smart installation, performing, 431–456
S/MIME encryption, 506
S/MIME messaging, 500
SMS (Systems Management Server), 455
SMTP (Simple Mail Transfer Protocol), 19, 70,
536, 712
  administering, 214–220
  basics of, 46–52
  clients, 83
  commands, 47, 48, 50–51, 712
  enterprise connector, 189
  headers, 52
  increased role of, 140
  instances, 141–142
  mail transmission example, 48–49
  message format, 214–215
  minimum number of commands in a
dialogue, 52
  port, 152
  protocol, 534
  queue threshold monitor, 179
  replies, 47, 48
  security and, 47
  between servers within a routing group, 92
  service, 141, 264
  virtual servers, 215–220
  virus protector, 512
SMTP connector, 84, 118, 141, 414, 538
  creating routing groups, 148
  pointing to different, 264
  selecting to send a notification, 181
SNADS connector, upgrading, 371
snap-ins
  available with Exchange 2000, 92–93
  to MMC, 16
SNMP (Simplified Network Management
Protocol), 70
software RAID, 425
software requirements for Conferencing Server,
470–471
software, updating, 569
sorting in columns in Outlook, 254
Source Code folder in the CD-ROM with this
book, 699
SPA (Secure Password Authentication), 277,
278, 293
spaces in X.400 addresses, 44
special accounts for installation and
administration, 438
special domain account, 438
SPNs (Service Principal Names), 488
SQL Server Sample Digital Dashboard, 692–693
SRS (Site Replication Service), 372, 712
SRS database files, 598
SSL (Secure Sockets Layer), 484, 712
  configuring in Outlook Express, 270
  encrypting sessions, 294
  with OWA, 305, 307–308
SSL client authentication, 212
SSL port 442, 224
SSL port 993, 220
SSL port 995, 222
stability of ADS, 22
Stack property page for the X.400 TCP Connector, 203
standalone information manager, configuring Outlook as, 236
standard operating procedures, 557
standard permissions, 119
standard port numbers for POP3, 285
standard TCP port numbers for Outlook Express services, 269
standard views in Outlook 2000, 643
standards
  extending, 34
  for messaging, 6, 7
  supported by Exchange 2000, 26–28
standards and conventions, process for developing, 553–558
standards document, 552–553
standards-based environments, troubleshooting in, 560
star topology, 149–150
Start Key Management Service dialog box, 495
Starts with condition, 108
Startup Wizard for Outlook 2000, 334
states for IMAP4 servers, 76
static information in Web parts, 690
steady state, transition to a, 363–364
.stm file, 166, 167
STM files, 598
storage
  changes in Exchange 2000, 18–19
  planning, 435
storage groups, 88, 162, 712
  assigning to a cluster, 170
  backing up entire, 603
  building, 165
  creating, 162–163
  multiple, 25
  multiple databases and, 162–170
  performance implications of multiple, 435
  physical location of transaction logs, 425
  restoring, 605
storage requirements, calculating, 422–425
storage technology, improvements in, 161
store events, 695–696
store providers, 230–232
STORE.EXE, 162
Store.exe process, 435
stores, 87, 712
stream algorithm, 506
streaming data, storing, 89
streaming file (.stm), 87
streaming media files, support for storing, 166–167
stubs, programmatic, 225
sub-components, 445
subdomains in the DNS hierarchy, 126
subnets, adding to the network, 134
subscribing to newsgroups in Outlook Express, 272–273
subtree model in DNS, 126
subtrees in the DNS hierarchy, 126
support team, 347
symbol files, 617
symmetric algorithm, 506
synchronization options
  configuring for offline folders, 246, 248
  for PIMs, 257–258
synchronization software for PIMs, 257
synchronization types for PIMs, 258
synchronous collaborative solutions, 636
synchronous events, 696
Sysprep, 454
system and network resources, controlling for conferences, 459
System Attendant. See SA (System Attendant)
system events, 696
system log, 616
System Management Tools installation component, 445
system messages, disallowing, 194
system monitor, configuring to monitor queries, 157–158
system page file, 420
system policies, 103–104
System Policies container for an Organization object, 100
System Policy objects, 103–104
system requirements
  for the CD-ROM with this book, 703–704
  for Conferencing Server, 469
System State folder, 597
systems, building large, 427–428

T
T.120 MCU/H.323 Conference Bridge, 477, 713.
  See also MCUs
table view, 244
tape drives
  adding multiple, 426
  cleaning, 569
Task Manager, 563–564, 584
  analyzing processor utilization, 422
  showing available memory, 591
tasks, scheduling, 565–570
TCP (Transmission Control Protocol), 70–71, 536
TCP port 25, 151–152, 219, 269
TCP port 80, 224
TCP port 110, 222, 269
TCP port 119, 209, 269
TCP port 143, 220, 269
TCP port 389, 269
TCP port 563, 269
TCP port 636, 269
TCP port 691, 151
TCP port 993, 269
TCP port 995, 269
TCP port numbers for Outlook Express services, 269
TCP Transport Service Stack General property page, 200
TCP/IP, 70, 77, 713
  configuring as an MTA transport, 199–200 protocols, 70
TCP/IP filtering, 491–492
TCP/IP model for the Internet, 534, 535
TCP/IP protocol stack
  for the Internet, 536
  providing protection on, 490
TCP/IP service, monitoring, 179, 180
team folder, 713
  Team Folder home page, 685
team folders, 679–686
  creating, 680, 682–685
  naming, 683
  Team Folders Kit (TFK), 679–680
  Team Folders Templates, 680
  Team Folders Wizard, 682–685, 686
  Team Folders Wizard Setup, 681
  Team Folder Wizard, deploying, 680
TechNet, 621, 622
telephone access to voice mail messages, 27
Telnet, 70, 291
tempdfrg.edb, 574
templates of team folders, 682–683
temporary human resources, 349
terminal address in X.400, 41
terminal emulation software on PCs, 4
test environment
  creating, 434
  populating with a one-way connection agreement, 373
test mode of ISInteg, 576
testing and validation of designs, 356–357
text word-wrap, 202
text-based messaging, 4
TFK (Team Folders Kit), 679–680
third-party connectors, upgrading, 371
third-party vendors, services written for Outlook by, 235
time lines, building, 353
Time synchronization server, 132
TLH (Top Level Hierarchy), 167, 169
TLS encryption, 219
tokens
  entered by users, 499
  generating for KMS, 497–498
Toolbox in Outlook 2000, 641
Tools container for an Organization object, 100
tools for migrating from Lotus Notes to Exchange, 408–410
Top Level Hierarchy (TLH), 167, 169
top-level containers in the Organization object, 100
top-level objects in the CDO Rendering Library, 645
topologies for routing groups, 149–150
topology considerations, affected by legacy system migration, 518–520
topology designs, 356
Topology Diagrammer tool, 586
TP0, 69
TP0.X.25 protocol, configuring as an MTA transport, 199, 200–201
TP4, 69, 71
tracking.log shared folder, 450
traffic. See network traffic
training in preparation for implementation, 355
training lead, 345
training plan, 341
training team, 346–347
transaction log file path, 599–600
transaction log files, 598
separating from their databases, 435
for storage groups, 88
transaction logs, 574, 599–600
transactions, writing to a write-ahead transaction log, 172
transfer time-outs on the X.400 TCP Connector Override property page, 204
Transmission Control Protocol. See TCP (Transmission Control Protocol)
Transport Access Point (TSAP) in TP4, 69
transport event sinks, 94
transport layer
of the OSI model, 35, 36, 534, 535, 711, 713
of the TCP/IP protocol stack, 534, 535
Transport Layer Security (TLS) encryption, 196
transport protocols, 198
trees, 129
trouble ticket database, 552
troubleshooting
basic method, 624
Internet client connectivity, 289–293
process of, 613–629
in a standards-based environment, 560
truncation in legacy messaging systems, 517
trust for delegation setting, enabling for Conferencing Server, 474
two-step migration by the Migration Wizard, 382
two-way CAs, 373–374
.TXT format for saving event logs, 567
Typical installation mode, 443

U
UAs (User Agents), 37
unambiguous name in X.500, 56
“unattend” setup switch, 453
Unattend.ini file, 454
Unattend.txt file, 453
undeliverable mail message notification in SMTP, 50
underscores, eliminating from names, 516
unicast connection to a conference, 461
Uninterruptible Power Supply (UPS), 427, 569
unique names in X.500, 56
unique node names, 126
Universal groups
creating, 331
Infrastructure Master and, 133
Universal Resource Identifier (URI), 74
Unix e-mail client, Pine as a, 288
UNIX file system, compared to DNS database structure, 125–126
Unix host, forwarding unresolved e-mail to, 219
unverified domain name, 220
upgrading, component level look at, 371
UPS (Uninterruptible Power Supply), 427, 569
URI (Universal Resource Identifier), 74
URL addressing in the Web Storage System, 89
URLs, listing of, 701–702
usage patterns, predicting, 522
Usenet news system, 207, 539
user accounts
changes to, 564
VBScript for renaming, 668
verifying prior to Conferencing Server installation, 473
User Agents (UAs), 37
User creation summary screen, 330
user mailboxes. See also mailboxes
common naming conventions for, 517
user management, effect on design, 526
user messaging, tracking after installation, 362
users. See also mail-enabled users
adding during Exchange 2000 installation, 329–330
adding in Exchange 2000, 23
associating with groups in ADSI, 669–670
configuring OWA for, 310

Continued
users (continued)
  connecting to a virtual server in OWA, 311–312
  creating mailboxes for, 110–111
  disconnecting all from NNTP, 214
  enabling in KMS, 498
  programming with ADSI, 667–669
  searching for on the Internet, 271
  selecting for team folders, 684
  surveying after rollout, 363
  training, 345
  VBScript for removing, 669–670
  VBScript for retrieving the full name of, 657
user-side files, 597
utilities, 95
  major available, 571–586
  in Outlook 2000, 251–254
utilization on clustered servers, 28
UUEncode, specifying for POP3, 222

V
 .vbs extension, 656
VBScript code, applying roles to forms, 644
vCard, including in a signature, 243
vendors
  establishing business partnerships with key, 555
  of Exchange 2000 products, 700
verifying backups and restores, 606
versions of Exchange 2000, 20–21
video conferences
  development of, 8
  joining, 461
video conferencing clients
  hardware requirements for, 469
  software requirements for, 471
Video Conferencing Provider, 29, 464–465
view summary, 244
View/Current View menu in Outlook 2000, 244
views
  in collaborative solutions, 638
  in Outlook 2000, 244–245, 643
virtual access control, 508–509
virtual days, 429
virtual directories, 312
Virtual Directories tab in NNTP, 214
virtual drive, added during installation, 449
virtual memory, 420
Virtual Memory monitor, 178, 179
Virtual Private Network (VPN), 713
Virtual Server properties, 508
Virtual Server property tab for NNTP, 209
virtual servers, 81, 82, 713
  configuring, 217
  connecting users to in OWA, 311–312
  controlling connections to, 509
  creating in OWA, 310–311
  designing and configuring multiple, 216
  disabling in OWA, 312
  managed by the SMTP service, 215–216
  setting up, 216
Virtual Server security, 507–509
virus protection in Exchange messaging systems, 511–512
virus protection software, checking for updates, 567–568
volatility, period of, 353, 562
VPIM (Internet Voice Messaging Protocol) format, 27
VPN (Virtual Private Network), 713
VPN router, encrypting sessions, 295
VRFY command in SMTP, 49, 51

W
W3C extended log format, 217
WAN bandwidth in intermediate-size organizations, 588–589
WAN/LAN backup, separate backup network for, 426
WAP (Wireless Access Protocol), 259
Web browser. See also browsers
  accessing e-mail from the Internet, 545
  using for messaging, 301
Web content, integrating in a Digital Dashboard, 691
Web forms, 95
Web Monitor tool, 586
Web pages in collaborative solutions, 638
Web Part Builder, 691
Web parts
  adding to a digital dashboard, 694
  defining, 690
Web Server, 521
Web sites, supporting multiple authors for, 89
Web Storage System, 84, 88–89, 647, 688–689
Web Store, 18, 24, 25, 171–173, 713
  administration of, 172–173
  indexing in, 171
  Internet protocols supported by, 171–172
  logging in, 172
Web Store Forms
  mapping table in, 689
  speed of, 689
Web-Based Enterprise Management (WBEM), 585
WebDAV (Web Distributed Authoring and Versioning), 539–540, 647, 713
  with OWA, 299
  in the Web Storage System, 89
  in Web Store, 171
weekly tasks, 567–568
White Book for the X.400 standard, 34
whiteboard in data conferences, 464
Windows 2000, 122
  additional sources of information on, 121
  core directory services provided by, 122
  domains in, 130
  editions of, 21
  installing, 317–324
  integrating service packs, 453
  integration with Exchange 2000, 89–94
  preparing the Exchange 2000 server, 437
  security subsystem of, 483
  server operating systems of, 21
  server roles, 521–522
  support Web page, 437
  unattended setup of, 453
Windows 2000 Active Directory. See Active Directory
Windows 2000 Advanced Server, clustering in, 170
Windows 2000 backup utility, backing up Exchange databases, 603–604
Windows 2000 bridgehead server, specifying for a CA, 377
Windows 2000 File System Sample Dashboard, 691–692
Windows 2000 Microsoft Management Console (MMC), 16, 22–23
Windows 2000 Server Event Viewer logs, 566–567
Windows 2000 services, monitoring, 179, 180
Windows 2000 sites. See sites
Windows Clustering, monitoring services on all nodes, 86–87
Windows DNA, 29–30
Windows Management Instrumentation (WMI), 585
Windows NT
  features of, 13
  OSI model and, 35, 36
  planning the migration to Windows 2000, 129
  reliance on NetBIOS, 125
  shortcomings of, 121–122
Windows NT Challenge/Response authentication method. See NTLM authentication method
Windows NT domain model, shortcomings of, 128
Windows Security provider Interface, 212
WindowsCE, 257, 258
WinNT provider, 655
Winroute tool, 586
WINS (Windows Internet Name Service), 125
WinZip on the CD-ROM with this book, 699
Wireless Access Protocol (WAP), 259
wireless e-mail, 594–595
wireless technology, 8
wizards in Outlook 2000, 251–253
WMI (Windows Management Instrumentation), 585
workflow capabilities of Exchange 2000, 25–26
workflow designer, 648–649
workflow engine, 25, 647–648
workflow states, delineating, 648
write-ahead transaction log, 172
written plans, 550
X
X.25, 67, 714
  configuring as an MTA transport, 199, 200–201
  OSI model and, 67
  terminology associated with, 67–68
  two-layer protocol for carrying packets, 68
X.400 addresses, 42–43, 44
X.400 conformance, setting, 206
X.400 connectors, 118, 142, 414, 538–539
  building, 198–206
  compared to site connectors, 519–520
  connecting routing groups, 148
X.400 enterprise connectors, 189
X.400 fields, 43–44
X.400 IPM, 40–42
X.400 messaging route, 118
X.400 queue growth monitor, 180
X.400 specification, 142
X.400 standard, 34, 714
  body part number assignments, 46
  components of, 37–38
  creating valid addresses, 42–43
  P designations in, 38
  properties significant to, 40–41
  relationship to X.500, 38–39
X.400 systems, 43–44
X.400 TCP Connector, 201–206
X.400-based MTA service, 142
X.400/RPC combination, compared to SMTP, 141
X.500 model, 53
X.500 standard, 52–56, 714
  limitations of, 27
  naming conventions, 56–59
  relationship to X.400 and ADS, 38–39
X.75, 68
XML (Extensible Markup Language), 172, 540, 708
XML Style Sheets, 172
XSL standard, 172
Z
zone databases, 126
zones
  compared to domains, 126–127
  in DNS, 73, 126–127
Hungry Minds, Inc.
End-User License Agreement

READ THIS. You should carefully read these terms and conditions before opening the software packet(s) included with this book (“Book”). This is a license agreement (“Agreement”) between you and Hungry Minds, Inc. (“HMI”). By opening the accompanying software packet(s), you acknowledge that you have read and accept the following terms and conditions. If you do not agree and do not want to be bound by such terms and conditions, promptly return the Book and the unopened software packet(s) to the place you obtained them for a full refund.

1. License Grant. HMI grants to you (either an individual or entity) a nonexclusive license to use one copy of the enclosed software program(s) (collectively, the “Software”) solely for your own personal or business purposes on a single computer (whether a standard computer or a workstation component of a multi-user network). The Software is in use on a computer when it is loaded into temporary memory (RAM) or installed into permanent memory (hard disk, CD-ROM, or other storage device). HMI reserves all rights not expressly granted herein.

2. Ownership. HMI is the owner of all right, title, and interest, including copyright, in and to the compilation of the Software recorded on the disk(s) or CD-ROM (“Software Media”). Copyright to the individual programs recorded on the Software Media is owned by the author or other authorized copyright owner of each program. Ownership of the Software and all proprietary rights relating thereto remain with HMI and its licensors.

3. Restrictions On Use and Transfer.

(a) You may only (i) make one copy of the Software for backup or archival purposes, or (ii) transfer the Software to a single hard disk, provided that you keep the original for backup or archival purposes. You may not (i) rent or lease the Software, (ii) copy or reproduce the Software through a LAN or other network system or through any computer subscriber system or bulletin-board system, or (iii) modify, adapt, or create derivative works based on the Software. (b) You may not reverse engineer, decompile, or disassemble the Software. You may transfer the Software and user documentation on a permanent basis, provided that the transferee agrees to accept the terms and conditions of this Agreement and you retain no copies. If the Software is an update or has been updated, any transfer must include the most recent update and all prior versions.
4. **Restrictions on Use of Individual Programs.** You must follow the individual requirements and restrictions detailed for each individual program in the *About the CD-ROM* appendix of this Book. These limitations are also contained in the individual license agreements recorded on the Software Media. These limitations may include a requirement that after using the program for a specified period of time, the user must pay a registration fee or discontinue use. By opening the Software packet(s), you will be agreeing to abide by the licenses and restrictions for these individual programs that are detailed in the *About the CD-ROM* appendix and on the Software Media. None of the material on this Software Media or listed in this Book may ever be redistributed, in original or modified form, for commercial purposes.

5. **Limited Warranty.**

   (a) HMI warrants that the Software and Software Media are free from defects in materials and workmanship under normal use for a period of sixty (60) days from the date of purchase of this Book. If HMI receives notification within the warranty period of defects in materials or workmanship, HMI will replace the defective Software Media.

   (b) **HMI AND THE AUTHOR OF THE BOOK DISCLAIM ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE SOFTWARE, THE PROGRAMS, THE SOURCE CODE CONTAINED THEREIN, AND/OR THE TECHNIQUES DESCRIBED IN THIS BOOK. HMI DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE SOFTWARE WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE SOFTWARE WILL BE ERROR FREE.**

   (c) This limited warranty gives you specific legal rights, and you may have other rights that vary from jurisdiction to jurisdiction.

6. **Remedies.**

   (a) HMI’s entire liability and your exclusive remedy for defects in materials and workmanship shall be limited to replacement of the Software Media, which may be returned to HMI with a copy of your receipt at the following address: Software Media Fulfillment Department, Attn.: *Exchange 2000 Server Administrator’s Bible*, Hungry Minds, Inc., 10475 Crosspoint Blvd., Indianapolis, IN 46256, or call 1-800-762-2974. Please allow four to six weeks for delivery. This Limited Warranty is void if failure of the Software Media has resulted from accident, abuse, or misapplication. Any replacement Software Media will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer.
(b) In no event shall HMI or the author be liable for any damages whatsoever (including without limitation damages for loss of business profits, business interruption, loss of business information, or any other pecuniary loss) arising from the use of or inability to use the Book or the Software, even if HMI has been advised of the possibility of such damages.

(c) Because some jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation or exclusion may not apply to you.

7. U.S. Government Restricted Rights. Use, duplication, or disclosure of the Software for or on behalf of the United States of America, its agencies and/or instrumentalities (the “U.S. Government”) is subject to restrictions as stated in paragraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause of DFARS 252.227-7013, or subparagraphs (c) (1) and (2) of the Commercial Computer Software - Restricted Rights clause at FAR 52.227-19, and in similar clauses in the NASA FAR supplement, as applicable.

8. General. This Agreement constitutes the entire understanding of the parties and revokes and supersedes all prior agreements, oral or written, between them and may not be modified or amended except in a writing signed by both parties hereto that specifically refers to this Agreement. This Agreement shall take precedence over any other documents that may be in conflict herewith. If any one or more provisions contained in this Agreement are held by any court or tribunal to be invalid, illegal, or otherwise unenforceable, each and every other provision shall remain in full force and effect.
CD-ROM Installation Instructions

Each software item on the *Exchange 2000 Server Administrator’s Bible* CD-ROM is located in its own folder. To install a particular piece of software, open its folder with My Computer or Internet Explorer. What you do next depends on what you find in the software’s folder:

1. **First, look for a ReadMe.txt file or a .doc or .htm document.** If this is present, it should contain installation instructions and other useful information.

2. **If the folder contains an executable (.exe) file,** this is usually an installation program. Often it will be called Setup.exe or Install.exe, but in some cases the filename reflects an abbreviated version of the software’s name and version number. Run the .exe file to start the installation process.

The ReadMe.txt file in the CD-ROM’s root directory may contain additional installation information, so be sure to check it.

For a listing of the software on the CD-ROM, see the *About the CD-ROM* appendix.