Administrator's Guide

Netscape Messaging Server

Version 4.1


Recycled and Recyclable Paper

Version 4.1

151-09694-00

Printed in the United States of America. 00 99 98 5 4 3 2 1

Netscape Communications Corporation, 501 East Middlefield Road, Mountain View, CA 94043

Contents

About This Guide 15
What You Need to Know 15
What's in This Guide 15
Document Conventions 17
Where to Find This Guide Online 18
How to Use Online Help 18
Where to Find Related Information 18

Chapter 1 Getting Started with Netscape Messaging Server 21
Messaging Server Features 22
Highly Scalable with Standard Protocols 23
Flexible Configuration and Monitoring 23
Powerful Security and Access Control 24
Convenient Management Interface 24
Where to Go for More Information 24
Deployment and Installation 25
Deployment Considerations 25
Installation Configurations 33
The Installation Process 38
Post-Installation Directory and File Organization 40
Using Netscape Console 44
Getting to a Messaging Server 46
Performing Typical Tasks 47
Performing All Configuration and Administration Tasks 49

Using the Command Line 52
Configuring General Messaging Capabilities 54
Viewing Basic Server Information 54
SNMP Setup 54
Configuring End-User Information 55
Configuring Default Languages 56
Specifying a Site Language 58
Specifying a Domain Language 59
Starting and Stopping Services 61
Customizing Directory Lookups 63
Encryption Settings 66
Where to Go from Here 67

Chapter 2 Configuring POP, IMAP, and HTTP Services 69
General Configuration 70
Enabling and Disabling Services 70
Specifying Port Numbers 70
Ports for Encrypted Communications 71
Service Banner 72
Login Requirements 72
Anonymous Login 72
Password-Based Login 73
Certificate-Based Login 74
Performance Parameters 74
Number of Processes 74
Number of Connections per Process 75
Number of Threads per Process 76
Dropping Idle Connections 77
Logging Out HTTP Clients 77

Client Access Controls 78
Configuring POP Services 78
Configuring IMAP Services 80
Configuring HTTP Services 82
Customizing HTTP Services 86

Chapter 3 Configuring SMTP Services 89
About SMTP 90
Viewing and Configuring Domain Information 90
Specifying an Address Completion Domain 91
Specifying the Domains Local to Your Server 92
Specifying Delivery Options 93
Delivering Mail to Unix Mail Folders 93
Delivering Mail to a Program 94
Deferring Delivery 96
Verifying Recipient Addresses 97
Performing Reverse IP Address Lookups 98
Specifying the Number of MTA Hops 99
Reserving Free Disk Space for the Message Queue 100
Enabling Optional SMTP Features 100
Verifying User Names (VRFY) 101
Verifying a Mailing List (EXPN) 102
Enabling Requests for Deferred Queue Processing (ETRN) 103
Limiting Message Size (SIZE) 103
Specifying Automatic Reply Information 104
Specifying Error Handling 106
Specifying Routing and Addressing Information 107
Specifying Envelope Rewrite Methods 108
Specifying From Address Rewrite Style 109
Specifying Alternate Search Methods 110
Editing SMTP Routing Table Entries 111
Controlling Access to SMTP Services 112
Enabling Authenticated SMTP 112
Specifying Access Control Filters 114

Filtering Unsolicited Bulk Email 114
Working with SMTP Plugins 114
Message Queue Concepts 114
Logical Queue 115
Physical Queue 115
Specifying Actions on Deferred Queues 116
Specifying Message Handling for Deferred Queues 117
Specifying Alternate Paths for Physical Queues 118

Chapter 4 Managing Mail Users and Mailing Lists 121
About Users and Groups for Messaging 121
Users and Mail Accounts 122
Groups and Mailing Lists 122
Mail-Administration Features 123
Managing Mail Users 124
Accessing Mail Users 124
Specifying User Email Addresses 128
Configuring Delivery Options 130
Specifying Forwarding Addresses 132
Configuring Auto-Reply Settings 133
Managing Mailing Lists 135
Accessing Mailing Lists 135
Specifying Mailing List Settings 139
Specifying List Members 141
Defining Message-Posting Restrictions 145
Defining Message-Rejection Actions 146

Chapter 5 Managing the Message Store 149
Overview 149
Message Store Directory Layout 150
How Messages Are Erased from the Store 154
Specifying Administrator Access to the Store 154
Adding an Administrator 155
Modifying an Administrator Entry 155
Deleting an Administrator Entry 156

Configuring User Disk Quotas 156
Specifying a Default User Disk Quota 157
Specifying a Quota Threshold 158
Defining a Quota Warning Message 159
Setting a Grace Period 159
Configuring Message Store Partitions 160
Specifying Aging Policies 162
Performing Maintenance and Recovery Procedures 165
Using the stored Utility 165
Managing Mailboxes 166
Repairing Mailboxes and the Mailboxes Database 166
Monitoring Disk Space 167
Monitoring Disk Quota Usage 167
Backing Up and Restoring the Message Store 167

Chapter 6 Security and Access Control 169
About Server Security 170
About HTTP Security 171
User Password Login 172
IMAP, POP, and HTTP Password Login 172
SMTP Password Login 173
Configuring SSL Encryption and Authentication 175
Obtaining Certificates 177
Enabling SSL 181
Setting Up Certificate-Based Login 184
Configuring Administrator Access to Messaging Server 185
Hierarchy of Delegated Administration 186
Providing Access to the Server as a Whole 187
Restricting Access to Specific Tasks 188
Configuring Client Access to TCP Services 189
How Client Access Filters Work 189
Filter Syntax 191
Filter Examples 196
Creating Access Filters for Services 198

Creating Access Filters for HTTP Proxy Authentication 200

Chapter 7 Working with SMTP Plug-Ins 203
About SMTP Plug-Ins 203
Managing SMTP Plug-Ins with Netscape Console 206
Installing Plug-Ins 206
Deleting (Uninstalling) Plug-Ins 207
Activating and Deactivating Plug-Ins 208
Configuring Plug-Ins 209
Managing SMTP Plug-Ins Manually 210
Installing and Configuring Plug-Ins 210
Deleting Plug-Ins 212
Installing and Configuring Protocol-Level Plug-Ins 212

Chapter 8 Filtering Unsolicited Bulk Email 215
About UBE 215
About the UBE Plug-In 216
UBE Filters and the UBE Plug-In 216
How UBE Filters Work 217
UBE Filter Format 218
Label 218
Message Field 218
Match Criteria 219
Action 220
Argument 220
Available Actions for UBE Filters 221
Regular Expressions for Match Criterion 223
Envelope Fields and Header Fields 225
Special Message-Field Names 227
Negation Modifier 230
Managing Filters with Netscape Console 230
Activating the UBE Plug-In 231
Creating a New Filter 231
Editing an Existing Filter 233
Activating and Deactivating Filters 234

Changing the Order of Filters 234
Parsing Header Fields 235
Creating Filters Manually 235
Plug-In File and Configuration Files 236
Editing the Filter Configuration File 237
Omitting Parts of a Filter 237
Entering Comments 238
Examples 239
Extending the UBE Plug-In 242
Using the RUN Action 242
Using an Extension Library 243
Anti-Relaying Defenses 244
Creating an Anti-Relay Filter 247
Using the Anti-Relay Plug-In 248

Chapter 9 Message Routing 255
Overview 255
Sending a Message 257
Routing a Message 257
Retrieving a Message 258
How Messaging Server Routes Messages 258
Step 1: Qualifying the Address 261
Step 2: Searching for matching LDAP Entries 262
Step 3: Checking if Domain is Local or Remote 263
Step 4: Checking Routing Attributes 263
Step 5: Routing to Remote Server 264
About Alternate Search Methods 266
About Recipient Address Rewrites 268
About Mailing List Expansion and Delivery 270
About the Domain Name System (DNS) 273

Chapter 10 Monitoring and Maintaining Your Server 277
Overview 278
Performing Daily Tasks 279
Checking postmaster Mail 279

Monitoring and Maintaining the Log Files 280
Setting Up the stored Utility 280
Starting and Stopping Services 281
Monitoring and Controlling Disk Usage 281
Monitoring Disk Usage 282
Controlling Disk Usage 282
Monitoring Server Response Time 284
Performing Recovery Tasks 285
Factors Affecting Messaging Server Performance 286
Number of Users per Disk 287
Configuration of POP, IMAP, and HTTP Services 287
Configuration of SMTP Services 288
Configuration of Logging Services 290
Size of Mailboxes 290
Distribution of the Store and Queue Directories 291
MTA Thread Settings 292
Applications Co-Resident with Messaging Server 293
Activity of Administration Server 293
Activity of Directory Server 294
Location of Messaging Server and Directory Server 294
Number of Address Lookups per Message 294
Ratio of Local Delivery to Outbound Sends 295
Use of RAID Technology 295
Memory, Disk, and CPU Requirements 295
System Monitoring Tools 296
Using SNMP on Unix Platforms 299
Communication Between the NMS and the Managed Device 300
The Messaging Server Subagent 301
Configuring SNMP 302
Configuring the Subagent 302
Enabling Statistics Collection 304
Starting and Stopping the Subagent 304
Verifying SNMP Configuration Changes 305

Chapter 11 Logging and Log Analysis 307
Log Characteristics 308
Services That Are Logged 308
Levels of Logging 309
Facilities as Categories of Logged Events 310
Filename Conventions for Log Files 311
Content Format for Log Files 312
Log-File Directories 313
Defining and Setting Logging Options 313
Flexible Logging Architecture 314
Planning the Options You Want 314
Setting Logging Options 315
Searching and Viewing Logs 318
Search Parameters 318
Specifying a Search and Viewing Results 319
Analyzing Logs with Third-Party Tools 320
Selected Event-Message Formats 321
SMTP-Accept Log Format 321
SMTP-Deliver Log Format 322
Mailbox-Deliver Log Format 323

Chapter 12 Program Delivery 325
About Program Delivery 325
Program Delivery and Mailbox Delivery 327
Program Delivery Failures 327
Security Considerations 327
Trusted Programs and Directory 328
Trusted Directory and Operating Modes 329
Guarding the Trusted Directory 329
Scripts and Batch Files 330
Enabling the Program Delivery Module 330
Using Program Delivery to Handle Incoming Mail 331
Administrators 332
Users and Account Owners 334

Program Delivery in Unix Environments 335
Program Delivery and Unix 336
How Program Delivery Works (Unix) 337
Secure and Non-secure Modes (Unix) 338
Running Programs as root 340
Setting Up Program Delivery (Unix) 340
Suspending Program Delivery (Unix) 342
Disabling Program Delivery (Unix) 343
Program Delivery in NT Environments 343
How Program Delivery Works (NT) 343
Setting Up Program Delivery (NT) 344
Suspending Program Delivery (NT) 345
Disabling Program Delivery (NT) 346

Chapter 13 Messaging Multiplexor 347
About Messaging Multiplexor 347
Multiplexor Benefits 348
How Multiplexor Works 350
Encryption (SSL) Option 351
Certificate-Based Client Authentication 352
User Pre-Authentication 353
Virtual Domains 354
Multiple Multiplexor Instances 356
Multiplexor Configuration 357
Multiplexor Configuration Parameters 359
Command-Line Configuration Options 366
Using the install Option 371
Installing and Configuring Multiplexor (Unix) 372
Before You Install (Unix) 372
Multiplexor Files (Unix) 373
Multiplexor Installation (Unix) 374
Creating a Multiplexor Instance (Unix) 375
Creating Additional Instances (Unix) 377
Modifying an Instance (Unix) 377

Multiplexor Configuration (Unix) 378
Installing and Configuring Multiplexor (NT) 386
Before You Install (NT) 386
Multiplexor Files (NT) 387
Multiplexor Installation (NT) 387
Creating a Multiplexor Instance (NT) 388
Creating Additional Instances (NT) 389
Modifying an Instance (NT) 390
Multiplexor Configuration (NT) 390
Running Multiplexor 395
Running Multiplexor (Unix) 395
Running Multiplexor (NT) 397
Uninstalling Multiplexor 398
Removing Multiplexor (Unix) 398
Removing Multiplexor (NT) 399

Appendix A Command-line Utilities 401
Overview of Command-Line Utilities 402
Command-Line Utilities—General Information 403
Messaging Server File Locations 404
Location of Configuration Data 404
Usage Requirements 405
Messaging Server Utilities—Descriptions 405
configutil 406
counterutil 410
deliver 410
hashdir 412
imscripter 413
mailq 415
mboxutil 417
MoveUser 419
NscpMsg 423
processq 424
qconvert 425

quota 427
readership 427
reconstruct 428
stored 433
upgrade 435
Alarm Attributes 437

Appendix B sendmail Migration and Compatibility 441
Moving Users to Messaging Server 441
Running the unix2ldif Utility 442
Running the ldifsplit Utility 451
Running the chkuniq Utility 453
Updating the LDAP Directory 454
Moving sendmail Messages to Messaging Server 456
Running the MigrateUnixSpool Utility 457
Compatibility with Unix sendmail 458
Command-line Compatibility 458
Functional Compatibility 460
sendmail Emulator Options and Aliases 462

Appendix C SNMP MIB 467
About the Messaging Server MIB 467
How the MIB Is Activated 468
Format of MIB Entries 469
Description of the MIB File 471
MIB Imports List 471
Module Definition 472
MIB Variables 472
MIB Traps 474
The Messaging Server MIB 476

Glossary 485

Index 501

About This Guide

The *Netscape Messaging Server Administrator's Guide* explains how to get started with, configure, and administer Netscape Messaging Server 4.1.

This preface contains the following sections:

- What You Need to Know
- What's in This Guide
- Document Conventions
- Where to Find This Guide Online
- How to Use Online Help
- Where to Find Related Information

What You Need to Know

The guide assumes that you are a server administrator with a general understanding of the following:

- The Internet and the World Wide Web
- Netscape Administration Server
- Netscape Directory Server and LDAP
- The Netscape Console user interface, as explained in *Managing Servers with Netscape Console,* the administrator's guide to using Netscape Console

What's in This Guide

The *Netscape Messaging Server Administrator's Guide* covers the information you need to understand, set up, and administer all aspects of Netscape Messaging Server. Check the following table for a quick summary of what each chapter covers.

If you want to do this:	See this chapter:
Learn more about the product and how to get started with all aspects of using it	Chapter 1, "Getting Started with Netscape Messaging Server"
Configure your server to support POP, IMAP, and HTTP services	Chapter 2, "Configuring POP, IMAP, and HTTP Services"
Configure your server's SMTP services	Chapter 3, "Configuring SMTP Services"
Learn about mail accounts and mailing lists and how to create and manage them	Chapter 4, "Managing Mail Users and Mailing Lists"
Learn about the message store and how to set up aging policies, quota policies, configure partitions, and generally manage the message store	Chapter 5, "Managing the Message Store"
Learn about server security in general and about how to configure security features of the server	Chapter 6, "Security and Access Control"
Install and configure SMTP plug-ins (dynamic libraries that extend the capabilities of Messaging Server)	Chapter 7, "Working with SMTP Plug-Ins"
Screen out unsolicited email	Chapter 8, "Filtering Unsolicited Bulk Email"
Learn about how Messaging Server receives and delivers messages	Chapter 9, "Message Routing"
Find out which monitoring and maintenance tasks you need to do and which are handled automatically	Chapter 10, "Monitoring and Maintaining Your Server"
Customize, manage, and view log files to help you monitor server operation	Chapter 11, "Logging and Log Analysis"
Set up Messaging Server to deliver incoming messages to external programs	Chapter 12, "Program Delivery"
Create a single point of connection to multiple Messaging Servers	Chapter 13, "Messaging Multiplexor"

If you want to do this:	See this chapter:
Identify utilities to help you with Messaging Server installation, migration, starting, stopping, administration, message management, problem recovery, monitoring, and reporting	Appendix A, "Command-line Utilities"
Move user mail accounts and mail messages from a Unix `sendmail` system to Messaging Server	Appendix B, "sendmail Migration and Compatibility"
Learn about how the Netscape Management Information Base (MIB) for Messaging Server stores the server information that administrators manage through the Simple Network Management Protocol (SNMP)	Appendix C, "SNMP MIB"
Look up terms you run across in using Messaging Server or its documentation	Glossary
Find information in the guide	Index

Document Conventions

This guide uses the following conventions:

- The `monospace` font is used for sample code and code listings, API and language elements (such as method names and property names), filenames, path names, directory names, HTML tags, and any text that must be typed on the screen.
- The *`monospace italic`* font is used in code to represent placeholder parameters that should be replaced with an actual value.
- Standard *Italic* type is used for book titles, emphasis, letters as letters (for example, "Spell it with a *T."*), and words as words (for example, "The word *server* is in every product name.*")*.
- Brackets ([]) are used to enclose parameters that are optional.
- A slash (/) is used to separate directories in a path. (Windows NT supports both the slash and the backslash.)

Where to Find This Guide Online

You can find the *Netscape Messaging Server Administrator's Guide* online in PDF and HTML formats. To find these files, use this URL:

`http://home.netscape.com/eng/server/messaging/4.1`

The information in this guide is also part of the help system for Messaging Server. See "How to Use Online Help."

How to Use Online Help

When you are using Netscape Messaging Server, you can click Help in any Messaging Server window to open a web browser window showing explanations of all of the window's user interface parts. You can then navigate to other parts of the help document, many of which are the same as those in the printed *Netscape Messaging Server Administrator's Guide*.

To move to the top of whatever major part of online help you're in and gain access to navigational buttons and to a drop-down Contents list, Index, and "Bookshelf" of resources, click the arrow button to the left of a heading. (If you don't see an arrow button, scroll until you do.)

Note: To view the full drop-down Contents list, you might need to make your browser window larger.

Because you view help in a browser window, you can use the navigational tools of your browser as well as the navigational tools within help. For example, you can use browser commands to print, find information, and copy and paste.

Where to Find Related Information

In addition to this guide, Messaging Server 4.1 comes with supplementary information for administrators as well as documentation for end users and developers. Use the following URL to see all the Messaging documentation:

`http://home.netscape.com/eng/server/messaging/4.1`

These are the additional documents that are available:

- *Messaging Server 4.1 Installation Guide*
- Release Notes
- Mailstone Utility
- *Messaging Access SDK Guide - Java Version*
- *Messaging Access SDK Guide - C Version*
- *Messaging Server Plug-in API Guide*

You can view documentation of schema for all Netscape products at this site:
`http://home.netscape.com/eng/server/directory/schema`

Chapter

1

Getting Started with Netscape Messaging Server

Welcome to Netscape Messaging Server 4.1. Messaging Server provides a powerful and flexible cross-platform solution to the email needs of enterprises and messaging hosts. It uses an open, Internet-standard approach to messaging while providing lightning-fast processing of messages that is scalable to many thousands of simultaneous users.

This chapter describes how to get started administering Messaging Server, from installation through basic configuration of general messaging capabilities. It concludes with references to other chapters that contain configuration instructions and procedures for managing your server and your community of mail users.

This chapter has the following sections:

- Messaging Server Features
- Deployment and Installation
- Using Netscape Console
- Using the Command Line
- Configuring General Messaging Capabilities
- Viewing Basic Server Information
- SNMP Setup
- Configuring End-User Information
- Configuring Default Languages
- Starting and Stopping Services

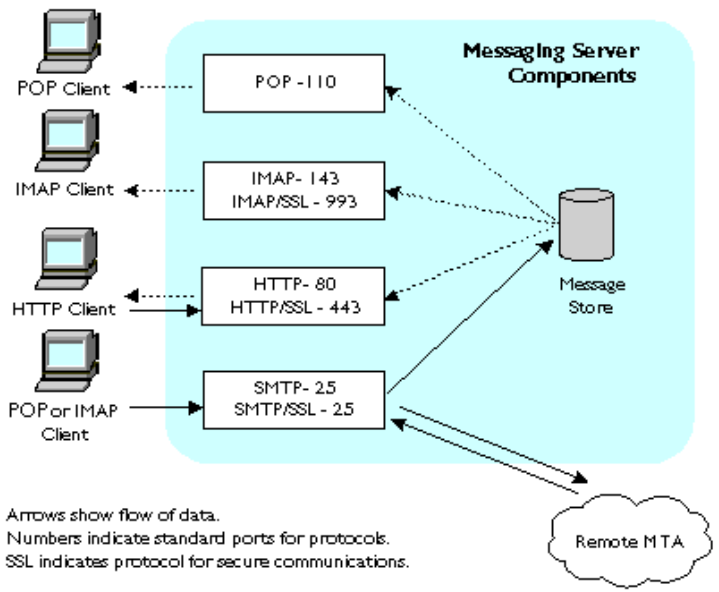
- Customizing Directory Lookups
- Encryption Settings
- Where to Go from Here

Messaging Server Features

Netscape Messaging Server 4.1 is the fourth generation of a powerful, standards-based Internet messaging server. Messaging Server is designed for high-capacity, reliable handling of the messaging needs of both enterprises and service providers of all sizes, from small to extremely large.

The server consists of several modular, independently configurable components with many powerful features. Figure 1.1 shows the core server components that support the standard messaging protocols.

Figure 1.1 Messaging Server components



Highly Scalable with Standard Protocols

- Fast and reliable, multithreaded message transfer agent (MTA) that uses the Internet-standard Simple Mail Transfer Protocol (SMTP) to handle both internal and Internet mail messages.
- Extremely efficient, high-capacity, and fast multithreaded Internet Mail Access Protocol (IMAP4) service for mailbox retrieval, supporting thousands of simultaneous users.
- Shared IMAP folders. End users can set access permissions on the IMAP folders in their mailbox.
- Full-featured, fast, multithreaded Post Office Protocol (POP3) service for mailbox retrieval, providing complete support for the most widely used Internet mailbox protocol.
- Specialized Hypertext Transfer Protocol (HTTP) service for web-based email; HTTP clients send mail to a specialized HTTP service which transfers requests to the MTA.
- Centralized database for server configuration, based on the Lightweight Directory Access Protocol (LDAP) (not shown in Figure 1-1).
- Centralized LDAP database for mail-user account storage, user authentication, and mail-routing control (not shown in Figure 1-1).
- High scalability of services to support greatly increased usage over time.

Flexible Configuration and Monitoring

- Flexible message store with fast indexed access, configurable partitions, quotas, and aging rules
- Highly configurable logging features with automated rollover
- Monitoring capabilities through Simple Network Management Protocol (SNMP)
- Multiplexor service for providing mail users a single point of connection to many POP and IMAP mailbox servers (not shown in Figure 1-1)

- Plug-in architecture for developing extensions to server capabilities
- Mailstone stress-testing utility for capacity planning and testing

Powerful Security and Access Control

- Support for password login (to POP, IMAP, HTTP, or SMTP) and certificate-based login
- Delegated administration through access-control instructions (ACIs)
- Client access filters (to POP, IMAP, HTTP, or SMTP)
- Filtering of unsolicited bulk email (UBE)

Convenient Management Interface

- Graphical Netscape Console interface for managing multiple servers from a single location
- End-user self-management of account information through HTML forms

Where to Go for More Information

Procedures for configuring these and other features are described in this book. Depending on the current status of your messaging deployment, you may want to start with one of the following sections in this chapter:

- If you have already installed Netscape Messaging Server and are ready to start configuring it, see "Configuring General Messaging Capabilities" on page 54.
- If you want information about using the Netscape Console graphical interface or the command line to configure and run Netscape Messaging Server, see "Using Netscape Console" on page 44 or "Using the Command Line" on page 52.

- If you are interested in an overview of the process of deploying and installing Netscape Messaging Server, see the next section, "Deployment and Installation."

Deployment and Installation

Before you can administer a server, its place in a deployment scheme must be determined and it must be installed. This section gives an overview of the issues involved in designing and installing a messaging solution with Netscape Messaging Server. It outlines some important deployment concepts and installation configurations to be considered, and then summarizes the installation process for a single server.

For complete documentation on Messaging Server installation, see *Messaging Server Installation Guide*. For more in-depth information on the deployment and installation-configuration topics presented here, see *Managing Servers with Netscape Console,* and Chapter 9, "Message Routing."

Deployment Considerations

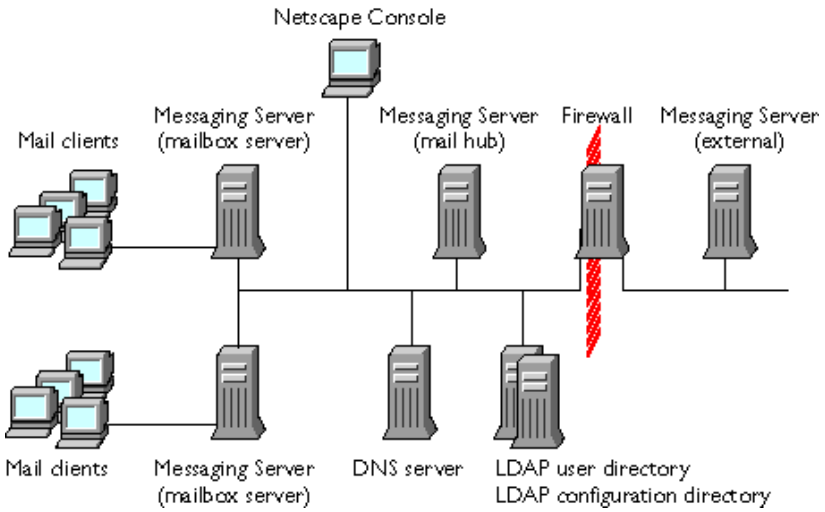
A successful messaging installation requires careful planning and execution. This section discusses some of the most basic topics to be considered in implementing a messaging solution with Messaging Server, including the following:

- Calculation of network topologies and server sizes
- Resolution of host names for routing messages among servers
- Resolution of user names for routing messages to mailboxes
- Deployment of additional servers for specialization and redundancy
- Integration of messaging with firewall security
- Creation and migration of mail accounts

This is not an exhaustive list of topics, and the discussion here won't by itself allow you to design and deploy a messaging solution; it provides only a context for subsequent server-specific discussions. For more in-depth information, consult the references listed with each topic.

Each installed Messaging Server is one component of the messaging solution implemented for your enterprise. Figure 1.2 is a simplified diagram of the principal components that might be found in an enterprise messaging solution. (Service providers may have additional components, as discussed in "Enterprise vs. ISP Topologies" on page 32.) How your Messaging Server needs to interact with clients, with other Messaging Servers, and with the other components shown in Figure 1.2 will affect how you install, configure, and maintain the server.

Figure 1.2 Potential components of an enterprise messaging solution



Sizing and Topology

Messaging installations that use Netscape Messaging Server are highly scalable. One or more servers can be organized into a messaging infrastructure that supports anywhere from a few users up to potentially millions of users.

Designing the network topology for a messaging solution, and calculating the numbers and sizes of host machines and server instances required (both today and in the foreseeable future), is a basic deployment task. It is also, typically, an iterative process.

One way to start is by relating your total user base to basic server capacity information as follows:

1. Start by assuming your total anticipated number of users.

2. Estimate your peak load: how many of your users need simultaneous access to their POP, IMAP, or HTTP mailboxes? Compare that to benchmark results of the maximum number of simultaneous connections possible with Messaging Server 4.1 on a given hardware configuration. Given those figures, estimate how many servers you need to handle your users.

3. Estimate your message traffic: how many total messages need to be sent through your messaging system per day? Compare that to benchmark results of the maximum message-transfer rate possible with Messaging Server 4.1 on a given hardware configuration. Given those figures, estimate how many servers you need to handle the message flow.

Benchmark studies and field deployments have shown that a single Messaging Server, installed on a moderately powerful, single-processor, dedicated server host machine with sufficient memory and storage, can, under optimum conditions, support several thousand users and deliver tens to hundreds of thousands of messages per day. Furthermore, these figures scale to much higher numbers as you add more processors to the host machine.

Initial estimates you make in this way are just the start of a sizing effort. Messaging Server and the other components it relies on function in a complex network of interactions, and requirements for specialization and redundancy can add further complexity. Multiple stages of recalculation, including actual field testing, are required as additional components and refinements are brought into the design.

Your Netscape representative can also help you address sizing questions, both for a new installation and for scaling existing installations to meet added demand. Consultants from Netscape's Worldwide Professional Services are also available to help design and implement installations of any size or complexity.

Role of DNS

The Domain Name Service (DNS) is an integral part of Internet communication; it converts names to machine addresses. DNS is a requirement for routing mail in a Netscape messaging installation. Unix and Windows NT operating-system

vendors make DNS available with their operating systems. For complete information on setting up and using DNS, see *DNS and BIND, 2nd ed.,* by Paul Albitz and Cricket Liu (O'Reilly).

Your enterprise must have at least one DNS server (the primary server) that has authoritative information for the names in your domain. You can have other DNS servers as well, on several host machines in several locations. Your DNS servers may be on machines dedicated to DNS or on machines with other responsibilities as well. Firewall machines are commonly used also as DNS servers.

Fundamentally, DNS translates host names and domain names to IP addresses, and vice versa. DNS uses Address (A) records for this purpose. Therefore, you need to make sure that your DNS server has A records for all Messaging Server hosts in your enterprise.

Secondarily, DNS can also translate domain names and host names to other host names. DNS uses Mail Exchange (MX) records for this purpose. This feature allows you to create private domains and to use domain-based email addresses (such as `sandee@airius.com`) instead of host-specific email addresses (such as `sandee@mail1.airius.com`).

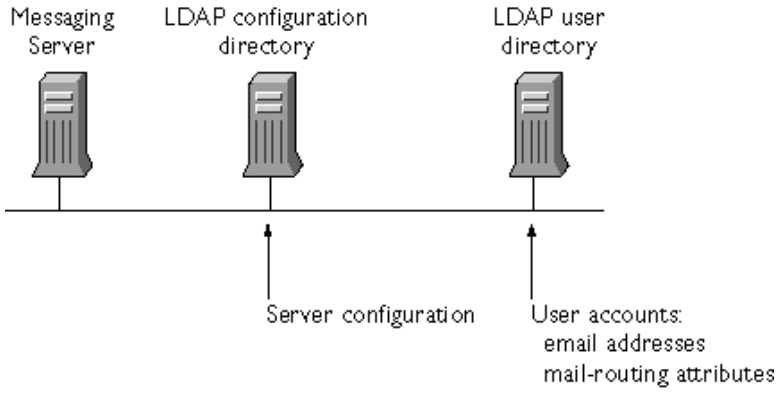
The way you set up DNS affects which of your servers first handle incoming messages, which ones pass outgoing messages to external recipients, and how messages within the enterprise get to the right mailbox server. For details on setting up A records and MX records, see Chapter 9, "Message Routing."

SMTP routing table. Each Messaging Server instance keeps a local SMTP routing table that, in addition to DNS, can determine the proper destination server for a message based on the recipient's address or domain. Entries in the routing table are optional, but they provide a method for directly transferring messages from one server to another. Routing-table entries are commonly used, for example, to directly transfer all outside messages to a firewall server. For more information, see Chapter 9, "Message Routing."

The Role of the LDAP Directory

Messaging Server 4.1 requires the use of an LDAP directory, such as Netscape Directory Server, for storing both server-configuration settings and mail-account information (Figure 1.3). A Directory Server must already be installed somewhere on your network before you can install Messaging Server.

Figure 1.3 User directory and configuration directory



The LDAP user directory in which your Messaging Server stores account information is typically on a separate host machine. A single Directory Server can manage the user directory for a very large organization, although for performance reasons all or parts of the directory are often replicated to one or more other machines. Setting up a directory is covered in detail in *Directory Server Deployment Guide* and *Directory Server Administrator's Guide*.

The entry for each user's account in the user directory includes mail-addressing and mail-routing attributes for that account. Whenever Messaging Server receives a message, it checks the user directory to make sure that the recipient's mail address (such as `dimitria@airius.com`) exists in the directory; if it does, Messaging Server routes the mail to the recipient's host server, also indicated in the directory entry. Routing the message may involve rewriting the mail address.

The process that Messaging Server uses to match a user in the directory with an email address can be complex. You can specify at least the following attributes for each user's directory entry: primary mail address, alternate mail addresses, mail host, and mail-routing address. For detailed information on how Messaging Server uses these mail-related attributes, see Chapter 9, "Message Routing."

Separation of Services

For increased performance and security, large enterprises may want to separate their messaging services by placing them on different host machines. As noted in Figure 1.2, for example, mailbox services might be separated from

centralized message-transfer services at a mail hub. Furthermore, different mailbox servers might be specialized for only POP or only IMAP. Other enterprises might in addition separate outgoing messages from incoming messages, channeling them through different SMTP mail hubs.

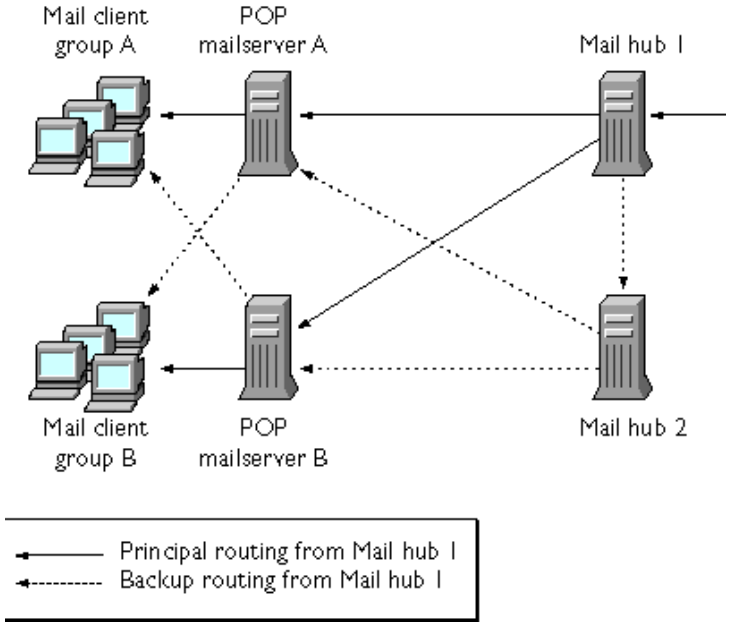
Such specializations increase the total number of servers and hosts in the enterprise and can greatly increase the complexity of routing configurations. As a result, directory services, DNS records, and SMTP routing tables need careful setup.

Redundancy Requirements

Server software is not perfect, nor are the host machines and network hardware it relies on. Almost any enterprise needs to plan for backup and for failover in case any of its important servers go down.

Therefore, in designing a messaging installation, be sure to consider the consequences of a failure of each individual Messaging Server and its host machine. Usually this means providing extra, redundant machines that can automatically take over a given server's tasks if it should fail. In installations in which messaging is distributed among specialized machines, servers already used to implement distributed functionality and replication can also function as failover servers (see Figure 1.4.)

Figure 1.4 Redundancy in a portion of a messaging installation



Designing your messaging topology for redundancy and setting up automatic failover capability can add greater complexity to an already complex configuration in a large installation.

Firewalls and Messaging

Most enterprises connected to the Internet maintain some form of firewall, a hardware or software barrier intended to prevent unauthorized external users from accessing the enterprise's servers and host machines. You can increase security by locating Messaging Servers behind the firewall, and channeling all mail access to the enterprise through one or more mail hubs, as shown in Figure 1.2. Channeling all outgoing mail through another hub provides additional control and security, allowing you to rewrite addresses or otherwise control information that leaves your enterprise.

For enterprises that receive a large volume of external mail, it might be optimal to place one Messaging Server, containing only publicly accessible accounts, outside the firewall. That server in turn would have limited access to internal servers, across the firewall, for forwarding messages to internal accounts.

Using a setup with mail hubs communicating across a firewall requires careful setup of firewall routing configurations, DNS services, and possibly SMTP routing tables to handle the complex routing possibilities. If you place a messaging server outside the firewall, you might need to use a separate, external, directory server as well.

Creation and Migration of Mail Accounts

Installing Messaging Server does not by itself create any user or group accounts or migrate existing proprietary mail accounts to the user directory. Messaging Server provides the Netscape Console graphical interface for entering user and group information for individual accounts; it provides command-line utilities for batch migration of large numbers of users to Netscape messaging from existing mail systems.

For instructions on how to enter and modify mail-related attributes in the user directory, see Chapter 4, "Managing Mail Users and Mailing Lists." For instructions on migrating `sendmail` user accounts to the LDAP user directory, see Appendix B, "sendmail Migration and Compatibility."

Enterprise vs. ISP Topologies

Enterprises with messaging intranets for employees are similar to Internet service providers (ISPs) with messaging hosting for subscribers, in that both can be required to support many thousands of accounts and a high volume of daily traffic. Typical network topologies and server configurations may differ, however.

For example, an enterprise might have many internal, directly connected mail users, with client machines and mail hosts located mostly inside the company firewall. Domain names may relate directly to host IP addresses. Client connections to mail servers may be frequent and heavy during the day, but drop off sharply after hours. Clients may stay connected for long periods.

An ISP, on the other hand, may have many servers but very few onsite client machines. Its customers typically retrieve their mail through dial-up connections. The ISP may offer custom domain services and thus may have multiple server instances per physical host machine. At the same time, ISPs may want to isolate users from specific mail hosts and thus are more likely to use a solution like Messaging Multiplexor. ISPs may have a larger proportion of mailbox servers to hubs than do most enterprises. Redundancy for 100% reliability may be even more important to an ISP than to many enterprises.

Client connections to the mail servers may be less frequent and shorter in duration, but they also may be spread out over more hours during the day, especially during the evening. ISPs, even more than enterprises, may be concerned with denying access to unauthorized users and filtering out unsolicited bulk email (UBE) to keep it from filling their customers' mailboxes.

Differences like these all have effects on the implementation of mail-routing strategies, access-filtering techniques, server-performance tuning, and server-installation configuration. For more information on access filtering and UBE filtering, see Chapter 6, "Security and Access Control," and Chapter 8, "Filtering Unsolicited Bulk Email."

Installation Configurations

To deploy a messaging solution that meets your needs and addresses the issues raised in the previous section, you may need to install Netscape Messaging Server on different host machines in different installation configurations. Depending on the size and purpose of your enterprise and the nature of your network and system hardware, your messaging deployment can consist of one or many instances of Messaging Server, on one or many host machines, with identical or different messaging capabilities. Required supporting software—such as Netscape Console, Administration Server, Directory Server, and the DNS service—may also be concentrated or distributed across your network.

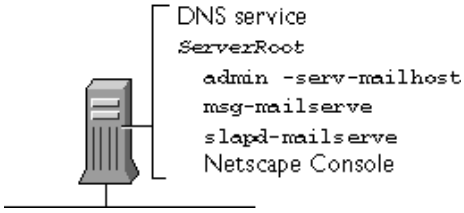
This section summarizes the common Messaging Server installation configurations. For more detailed information on installation configuration and on the interaction between Netscape Messaging Server and other services, see *Managing Servers with Netscape Console*. For additional information on LDAP directories and the Netscape Directory Server, see the Directory Server documentation.

All Services on One Host

A one-host configuration (shown in Figure 1.5) can be practical for smaller installations. It economizes on server hardware at the expense of performance and capacity. (It also provides no backup, should the one server fail.) Nevertheless, it is possible to use a single host machine to house everything. Note that, in this configuration, the single *server root* (the directory into which all Netscape servers are installed) contains the three required Netscape servers—Messaging Server, Directory Server, and Administration Server—as a

single *server group* (the set of servers managed by a single Administration Server). The single Directory Server in this case manages both the *user directory* (which contains mail-account information) and the *configuration directory* (which contains server-configuration information). The DNS service and Netscape Console are also on the same host machine.

Figure 1.5 All messaging-related services on a single host

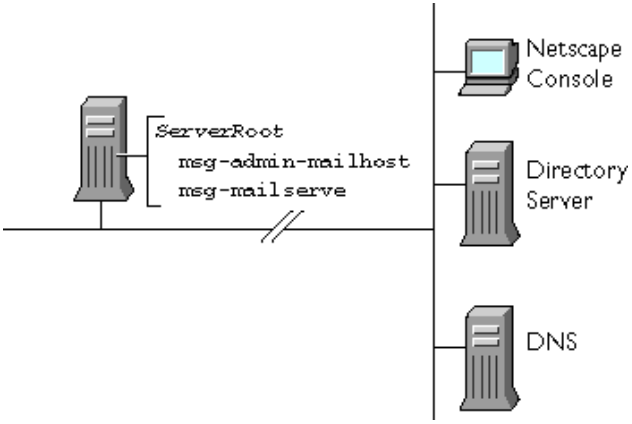


In this configuration the one host machine must have sufficient performance capacity to handle all services without undue strain. It must also have sufficient storage capacity to hold all messages and all directory information for the enterprise.

One Messaging Server per Dedicated Host

A common deployment configuration is to have a dedicated host machine for each Messaging Server instance. As Figure 1.6 shows, the LDAP directory (or directories, if user and configuration directories are separated), the DNS service, and possibly Netscape Console are on separate hosts from the installed Messaging Server. There may be one or several messaging host machines, but each contains a single server root in which a single Messaging Server and its Administration Server make up the server group.

Figure 1.6 Single Messaging Server on a single host

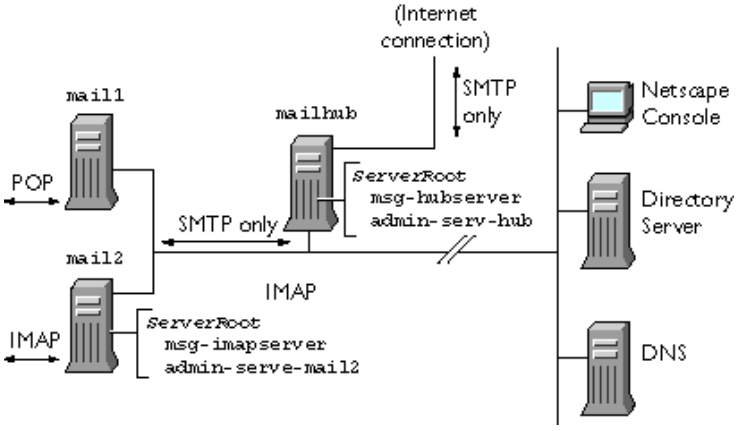


This configuration allows for optimizing each server host machine for strictly messaging tasks. Different divisions or offices of the enterprise may each have their own Messaging Server in a configuration like this one, perhaps with all servers accessing a single user directory on a dedicated host machine.

Specialized Messaging Services on Each Host

Another common deployment configuration, especially in larger installations, is to implement only certain messaging services on each host machine. As shown in Figure 1.7, for example, a centralized mail hub server, using only SMTP, connects to individual mailbox servers that use only POP or only IMAP to send mail to their users.

Figure 1.7 Mail hub and mailbox servers on separate hosts

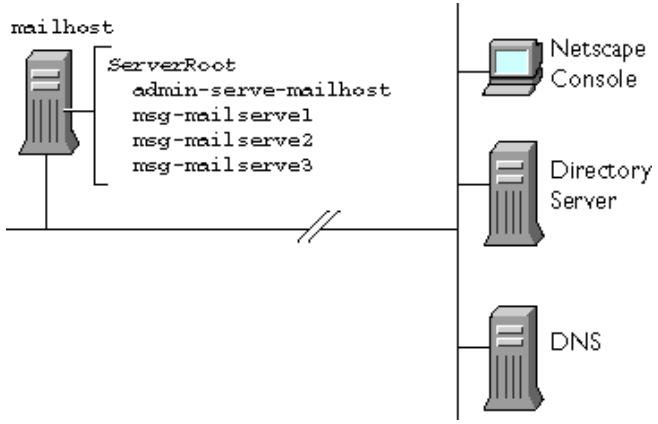


This configuration can increase security (because outsiders can connect only at one point, the mail hub), and it allows for optimizing each server machine for the specific service (SMTP, POP, IMAP) that it supports.

Multiple Server Instances per Host

If appropriate for your needs, you can install multiple server instances on a single host machine. As the example in Figure 1.8 shows, a single server root contains a server group consisting of one Administration Server and multiple instances of Messaging Server. All Messaging Server instances run from a single installed set of executable programs and libraries.

Figure 1.8 Multiple instances of Messaging Server on a single host

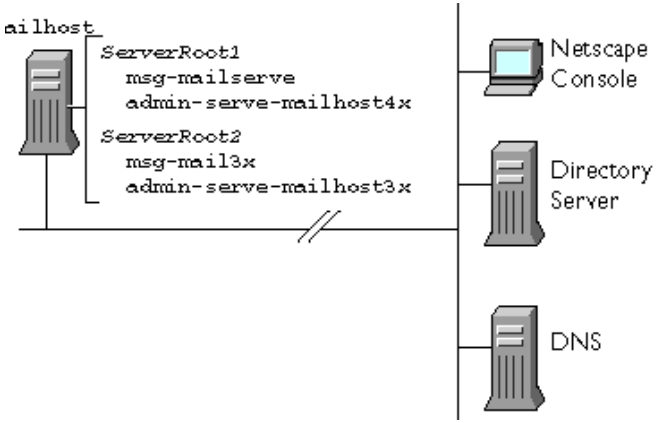


This configuration allows multiple custom domain names to be created for a single machine that has a single IP address. A host machine in this configuration must have sufficient capacity to execute and store messages from all the server instances.

Multiple Server Roots per Host

If a single host machine includes Netscape servers that have different version numbers, it may be necessary to create separate server groups, and thus separate server root directories, on the machine. Figure 1.9 shows an example in which some employees in an enterprise are using Netscape Messaging Server 3.0, while others have upgraded to Netscape Messaging Server 4.1. Both servers are running on the same host machine.

Figure 1.9 Two versions of Messaging Server on a single host



This configuration may be necessary because different versions of Messaging Server may require different directory structures or different versions of the Administration Server. The Netscape Server Setup program facilitates this configuration, letting you create a separate server root when you install new servers and leaving an existing server root undisturbed.

This configuration is commonly used for pilot deployment of new server versions, for creating a temporary setup until all users migrate to the newer version, or even for failover protection, with different server instances stored on different physical storage devices.

The Installation Process

All Netscape servers, and also the Netscape Console application that you use to manage them, are installed by running the Netscape Server Setup program. The program is provided with every Netscape server product.

This section only summarizes the installation process. For detailed instructions on installing Netscape Messaging Server, see the *Messaging Server Installation Guide*. For additional general information on the Netscape Server Setup Program, see *Managing Servers with Netscape Console*.

Before you install Messaging Server, your Netscape Directory Server must be installed and your DNS service set up. Then you can install Messaging Server onto its host machine.

In summary, take these steps:

1. Obtain the Messaging Server installation package and unpack the files.

   Whether you have obtained the package from a CD-ROM or through a network download, copy the package into a temporary directory and unpack the files into that directory.

2. Configure your LDAP Directory Server appropriately for messaging, using the tools provided.

   The configuration tools add Messaging Server schema extensions to the configuration directory and prepare it for holding server-configuration information for this server instance.

3. Run the Netscape Server Setup Program (`setup`).

   Read the Welcome message and the Licensing Agreement, select the products to be installed (servers, components, or Netscape Console), choose a level of installation (Express, Typical, or Custom), and answer the prompts.

   If this is the first installation of Netscape servers on this host machine, the setup program also installs an instance of Netscape Administration Server. For information on how the Administration Server works and how to install and configure it, see *Managing Servers with Netscape Console*.

4. At the last prompt, confirm the correctness of the information you have entered.

   At this point, the installer extracts the appropriate files, configures the Administration Server (if it is being installed) and the Messaging Server, and starts the servers.

Installation is complete. You can now use Netscape Console (see "Using Netscape Console" on page 44) to continue configuring the server (see "Configuring General Messaging Capabilities" on page 54 and "Where to Go from Here" on page 67).

Silent Install. You can use the Netscape Server Setup program, along with a special configuration file, to install Messaging Server in a non-interactive mode that does not require your continued presence at the machine on which the installation occurs. If you have many similar server configurations to set up, you can place the configuration file plus the server installation package on each machine. You execute the setup program on each machine; it then extracts all information it needs from the configuration file as it performs the installation.

Whenever you perform a manual installation, the setup program creates a log file that you can use as the configuration file for subsequent silent installs. For more information, see the *Messaging Server Installation Guide*.

Console-only installation. You can use the Netscape Server Setup program to install the Netscape Console alone, so that you can use it from a client machine for remote administration. The setup program can also install Messaging Server patches and updates. For details, see the installation instructions.

Post-Installation Directory and File Organization

Once you have installed Netscape Messaging Server, its directories and files are arranged in the organization depicted in Table 1.1. The table is not exhaustive; it shows only those directories and files of most interest for typical server administration tasks.

Note: Where pathnames for Windows NT and Unix installations are identical except for separator symbols, only the Unix version is shown. Where they differ materially, both are shown. Metavariables (replaceable text strings) in pathnames are shown in italics.

Table 1.1 Important Messaging Server directories and files

Directory or file	Default or required location	Explanation
server root directory (*serverRoot*)	**Unix:** `usr/netscape/server4/` (default location) **NT:** `Netscape\Server4\` (default location)	The directory into which all servers of a given server group (that is, all servers managed by a given Administration Server) are installed. This may include other Netscape servers in addition to Messaging Server.
installation directory (*installDirectory*)	*serverRoot*`/bin/msg/` (required location)	The directory containing the binary (executable) files of the installed Messaging Server.

Table 1.1 Important Messaging Server directories and files (Continued)

Directory or file	Default or required location	Explanation
instance directory (*instanceDirectory*)	*serverRoot*/msg-*instanceName*/ (required location) where *instanceName* is the name of this instance of Messaging Server, as specified at installation. (Default = host name of server machine)	The directory containing the configuration files that define a given instance of Messaging Server. Multiple instances of Messaging Server, all using the same binary files, may exist on a given host machine.
message queue directory	*instanceDirectory*/queue/ (default location)	The directory that holds the message queues, the temporary holding areas for received messages. For more details, see "Message Queue Concepts" on page 114.
message store directory	*instanceDirectory*/store/ (required location)	The directory that holds the user mailboxes. For more details, see "Managing the Message Store" on page 149.
user mailbox	*instanceDirectory*/store/partition/primary/=user/userID/*subMailbox*/ where *userID* is the mail ID of the user, and *subMailbox* is the POP or IMAP folder (such as INBOX) (default location)	The location within the message store of a given mailbox directory. For more details, see "Managing the Message Store" on page 149.
administrative command-line utilities	*installDirectory*/admin/bin/ (required location)	The directory containing command-line utilities that handle most aspects of server configuration and management. For more details, see "Command-line Utilities" on page 401.

Table 1.1 Important Messaging Server directories and files (Continued)

Directory or file	Default or required location	Explanation
storage-related command-line utilities	*installDirectory*`admin/bin/` (required location)	The directory containing command-line utilities that handle mail delivery and storage-database management. For more details, see "Command-line Utilities" on page 401.
start-stop utility	**Unix:** `/etc/NscpMsg` (required location)	A Unix-only utility that starts and stops Messaging services. For more details, see "Command-line Utilities" on page 401.
	Windows NT: `Control Panel->Services->Start` or `Stop` (required location)	The Windows NT Services Control Manager.
local configuration file	*instanceDirectory*`/config/configdb` (required location for Unix, default location for NT)	A file containing locally stored Messaging Server configuration information; includes the location of the main server-configuration information, stored on an LDAP directory server. For more details, see "configutil" on page 406.
SMTP routing table	*instanceDirectory*`/config/configdb`	A portion of the file `configdb` consisting of routing instructions for forwarding messages from this server to other servers. For more details, see "Editing SMTP Routing Table Entries" on page 111.
trusted directory	*instanceDirectory*`/smtp-bin/delivery` (required location)	The directory that holds programs that work with program delivery. For more details, see Chapter 12, "Program Delivery."

Table 1.1 Important Messaging Server directories and files (Continued)

Directory or file	Default or required location	Explanation
Mailstone utility	`/mailstone/` (default location after separate Mailstone installation)	The directory that holds the executable and configuration files for the Mailstone stress-testing utility. See *Netscape Mailstone Utility* for more details.
Messaging Multiplexor	*`serverRoot`*`/mmp/` (default location after separate Multiplexor installation)	The directory that holds the executable and configuration files for Messaging Multiplexor. For more information, see Chapter 13, "Messaging Multiplexor."
log files	*`instanceDirectory`*`/log/`*`service`* (default location) where *`service`* is the name of the service (such as IMAP) being logged	The directories containing sets of log files for each of the services provided by Messaging Server. For more details, see Chapter 11, "Logging and Log Analysis."
SMTP plug-ins configuration file	*`instanceDirectory`*`/smtp-bin/plugins/plugins.cfg` (required location)	The file that specifies which SMTP plug-ins have been installed and what their configurations are. For more details, see Chapter 7, "Working with SMTP Plug-Ins."
UBE filter configuration file	*`instanceDirectory`*`/smtp-bin/plugins/UBEfilter.cfg` (default location)	The file that contains the mail filtering rules for the Unsolicited Bulk Email (UBE) plug-in. For more details, see Chapter 8, "Filtering Unsolicited Bulk Email."
End-user interface HTML pages	*`serverRoot`*`/bin/user/admin/` (default location)	Customizable HTML pages and associated CGIs that provide end-user access to account information. For more details, see "Configuring End-User Information" on page 55.

Using Netscape Console

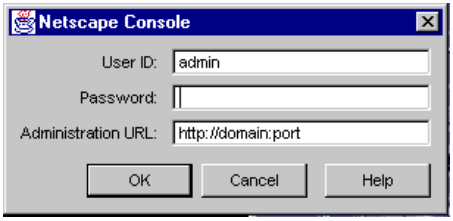
Netscape Console is a Java application that provides server administrators with a graphical interface for managing all Netscape servers. From any installed instance of Netscape Console, you can see and access all the Netscape servers on your enterprise's network to which you have been granted access rights. (For information on how administrator access to servers is configured, see "Configuring Administrator Access to Messaging Server" on page 185.) For complete documentation on Netscape Console, see *Managing Servers with Netscape Console*.

If you need to create a new instance of Netscape Console for managing Messaging Server, use the Netscape Server Setup program (see "The Installation Process" on page 38) to install Netscape Console onto the machine from which you intend to administer your Messaging Servers. You can install Netscape Console onto the same host as a Messaging Server, or onto any other machine on the network.

Note: For any given instance of Netscape Console, the limits of the network it can administer are defined by the set of resources whose configuration information is stored in the same configuration directory. That is the maximum set of hosts and servers that can appear in the Console window. For a given administrator using Netscape Console, the actual number of visible serves and hosts may be fewer, depending on the access permissions that administrator has.

When you launch Netscape Console, it first displays a login window (Figure 1.10). You enter your administrator's ID, your password, and the URL (including port number) of the Administration Server representing a server group to which you have access. You cannot use Netscape Console without having login access to at least one server group on your network.

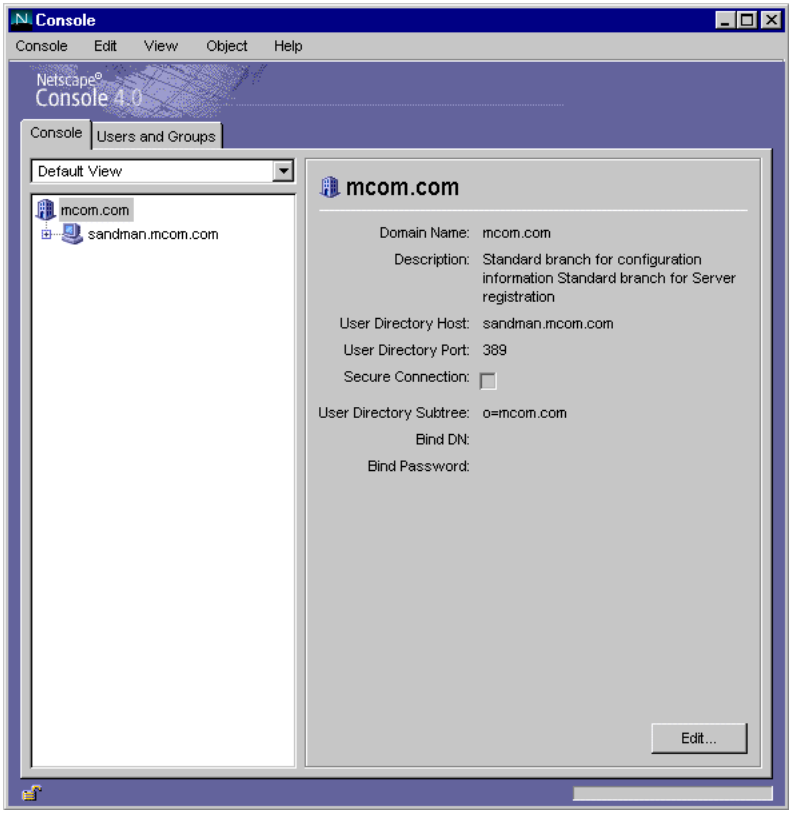
Figure 1.10 Netscape Console login window



If the information you enter into the login window is acceptable, Netscape Console displays a graphical representation of all the hosts and servers on your network that you have access to.

In the example shown in Figure 1.11, the left pane of the Console window shows that the entire network to which the administrator has access consists of a single host machine and all the servers on it. (See *Managing Servers with Netscape Console* for an explanation of the administrative-domain information displayed in the right pane of Figure 1.11.)

Figure 1.11 Netscape Console window (with Domain Information form)



Getting to a Messaging Server

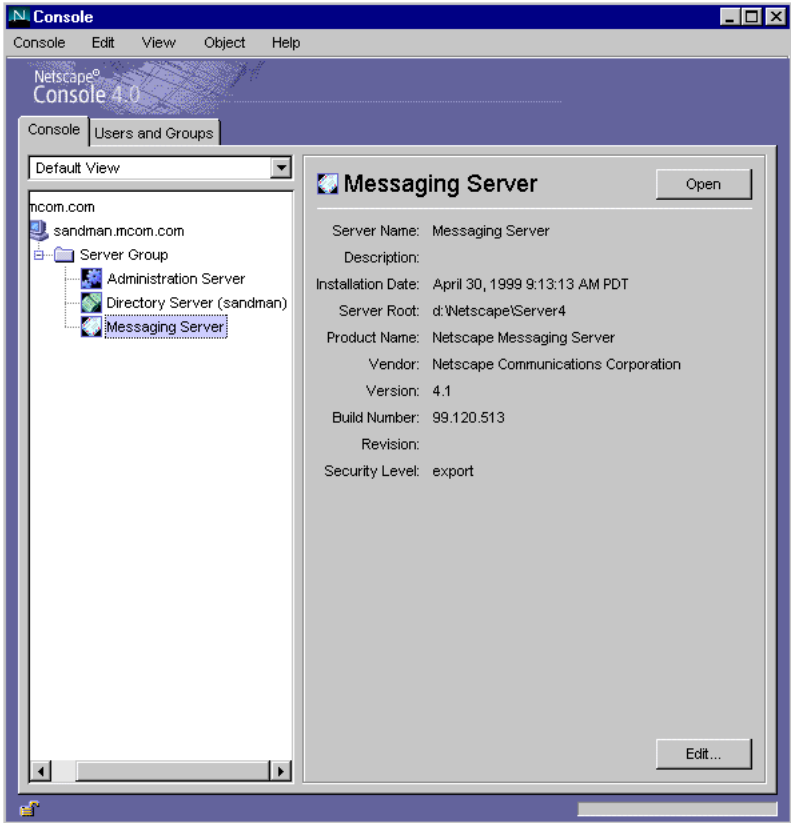
After you have launched Netscape Console, take these steps to access the instance of Messaging Server you want to manage:

1. In the Netscape Console window, click the Console tab if it is not already frontmost.
2. Open the icon of the host machine containing the server.
3. Open the folder icon representing the server group that contains the server.
4. Select the icon of the server itself.

   The Server Information form for the selected server appears, as shown in Figure 1.12.
5. Open the selected Messaging Server.

   Either click the Open Server button in the Server Information form or double-click the selected server icon below the Console tab. The Messaging Server Tasks form, described next, appears.

Figure 1.12 Netscape Console window (with Server Information form)



Performing Typical Tasks

When you open Messaging Server from Netscape console, the first item displayed is the Tasks form (Figure 1.13). The Tasks form contains a list of common Messaging Server administration tasks; clicking the button beside a task opens windows through which you can perform the task.

Figure 1.13 Messaging Server Tasks form

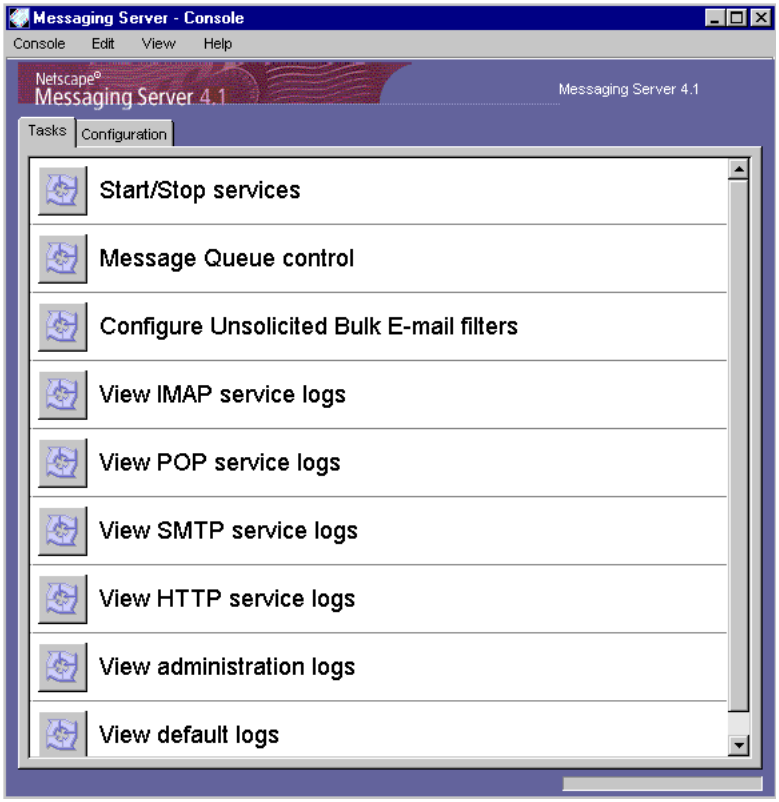


Figure 1.13 shows the full list of available tasks. When you open a Messaging Server, you may see fewer tasks, depending on your access rights to the server. For more information on administrator access to server tasks, see "Configuring Administrator Access to Messaging Server" on page 185.

Table 1.2 directs you to the part of this book that describes procedures for performing each task listed in Figure 1.13.

Table 1.2 Documentation for tasks listed in the Tasks form

Task	Where described
Start/Stop services	“Starting and Stopping Services” on page 61
Message Queue control	“Message Queue Concepts” on page 114
Configure Unsolicited Bulk Email filters	“Filtering Unsolicited Bulk Email” on page 215
View IMAP Service logs	“Searching and Viewing Logs” on page 318
View POP Service logs	“Searching and Viewing Logs” on page 318
View SMTP Service logs	“Searching and Viewing Logs” on page 318
View administration logs	“Searching and Viewing Logs” on page 318
View default logs	“Searching and Viewing Logs” on page 318

Using the Task form is not the only way to access server tasks. If you have the required access rights to the server, you can perform all the tasks shown in Figure 1.13—and many other tasks as well—through the Configuration Tab (described next).

Performing All Configuration and Administration Tasks

You can use the Configuration Tab to access all task and configuration forms available through Netscape Console. Access through the Configuration tab is more complete, though not always as direct, as through the Tasks form. Take these steps to access a task through the Configuration tab:

1. In Netscape Console, open the Messaging Server that you want to configure. (If you need instructions, see “Getting to a Messaging Server” on page 46.)

2. Click the Configuration tab.

   The left pane of the window displays a hierarchical set of icons that represent the services and features of Messaging Server. The Messaging Server icon itself is at the top; directly below it are icons for Services,

Message Store, and Log files. These icons can be individually selected, and some can also be opened to reveal other icons that can themselves be selected or opened.

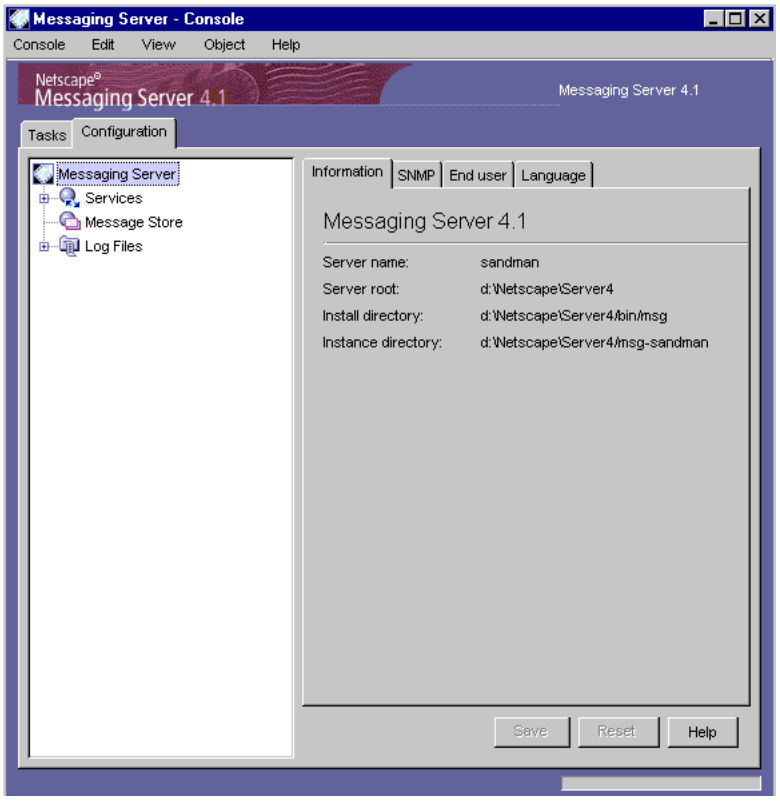
3. Select an icon, or open an icon and select one of the icons that appear below it.

   The right pane displays a form, possibly including tabs for accessing additional forms. For forms that include tabs, clicking a tab displays another form related specifically to that tab. The form or forms are the interface to a configuration or administration task represented by the selected icon.

4. View or enter information into the forms, as appropriate, to complete the task.

For example, if you select the Messaging Server icon itself in the left pane, the right pane displays the three tabs shown in Figure 1.14.

Figure 1.14 Messaging Server Configuration tab with Messaging Server icon selected



The tasks that you can perform through these Netscape Console forms are described throughout the rest of this book.

Note: The set of tasks available in this manner is a superset of the tasks available through the Tasks form described in the previous section. Also note that most server tasks can be performed from the command line; see "Using the Command Line" (next).

Using the Command Line

Netscape Messaging Server provides a set of command-line utilities as an alternative to using the Netscape Console interface for performing certain configuration and administration tasks. In the case of massive or repetitive operations, such as batch processing of user accounts, it can be far more efficient to use the command line than to manually enter information at the console.

Table 1.3 lists the command-line utilities available with Netscape Messaging Server. For reference information on these utilities, see Appendix A, "Command-line Utilities," and Appendix B, "sendmail Migration and Compatibility."

Table 1.3 Command-line utilities

Command-line utility	Description
Management	
`configutil`	Lets you view and make changes to server configuration settings (both local settings and settings stored in the configuration directory).
`imscripter`	Executes an IMAP command or sequence of commands.
`mboxutil`	Lists, creates, renames, or moves mailboxes.
`hashdir`	Identifies the directory that contains the message store for a particular user.
`processq`	Manually delivers messages from the mail queue.
`deliver`	Delivers mail to a user mailbox.
`stored`	Performs background and daily tasks on the message store; erases expunged messages.
Monitoring and reporting	
`counterutil`	Monitors a counter object and displays all counters in it.
`mailq`	Checks the mail queue and reports the number of messages in it.
`quota`	Reports mailbox quota usage.
`readership`	Collects readership information on shared mailboxes.

Table 1.3 Command-line utilities (Continued)

Command-line utility	Description
Recovery	
`NscpMsg`	Starts and stops the server and resets configuration variables (Unix only).
`reconstruct`	Reconstructs mailboxes that have been damaged or corrupted.
Migration from another messaging server	
`MoveUser`	Moves contents of user mailboxes from one Messaging to another.
`qconvert`	Converts a Messaging 3.x message queue to Messaging 4.x format.
`upgrade`	Converts Messaging 3.x mailboxes to 4.x format and moves them to the 4.x server.
Migration from `sendmail`	
`unix2ldif`	Converts Unix `sendmail` user-account information to LDAP Directory Interchange Format (LDIF).
`ldifsplit`	Analyzes the results of the `ldifsplit` utility, separating the LDIF data into entries that are already in the user directory from those that are not.
`chkuniq`	Checks the output of `unix2ldif` and `ldifsplit` for duplicate entries.
`ldapmodify`	Updates an LDAP directory with the LDIF output of the `sendmail` utilities.
`MigrateUnixSpool`	Moves user messages from `sendmail` spool files to Messaging Server mailboxes.

In this book, the description of each server task you can perform includes a discussion of the command-line utilities, if any, that you can use to accomplish the task.

Configuring General Messaging Capabilities

The following sections describe the general Messaging Server tasks—such as starting and stopping services, configuring directory access, and configuring end-user access—that you can perform with Netscape Console or with command-line utilities. Tasks specific to individual Messaging Server services—such as POP, IMAP, and SMTP—are described in subsequent chapters.

Viewing Basic Server Information

You can review some of the basic information about an installed Messaging Server by viewing its Information form in Netscape Console.

To display the Information form:

1. In Netscape Console, open the Messaging Server whose information you want to view.
2. Select the server's icon in the left pane.
3. Click the Information tab in the right pane, if it is not already frontmost.

   The Information form appears. It displays the server name, server root directory, installation directory, and instance directory. (For an explanation of these terms, see Table 1.1 on page 40.)

SNMP Setup

You can use Netscape Console to set up and enable the Simple Network Management Protocol (SNMP) subagent for your Messaging Server. By using SNMP, an administrator can monitor multiple servers remotely through an SNMP network management station.

The Messaging Server subagent collects information and generates statistics relating to the server's functioning, and it transfers the information to the SNMP master agent.

Although this task is a general configuration task, it is described in Chapter 10, "Monitoring and Maintaining Your Server." For a description of the Messaging Server SNMP management information base, see Appendix C, "SNMP MIB." For information on setting up your network's SNMP master agent, see *Managing Servers with Netscape Console*.

Configuring End-User Information

Netscape Messaging Server provides end users with limited server access, through which they can manage certain aspects of their own mail accounts and also create or subscribe to mailing lists. The server employs HTML forms that users fill out to make these changes.

Netscape Messaging Server includes a set of HTML forms (and associated CGI scripts) for this purpose. As server administrator, you can control which forms, if any, users can access, and where those forms are located. You specify the URLs to those forms, so that client software that connects to your server can access the forms. The following forms are provided with Netscape Messaging Server:

- **Personal Account Manager.** This form allows users to perform tasks such as changing their passwords or modifying their personal information (home phone number, vacation message, and so on).
- **Mail Account Manager.** This form allows users to manage parts of their mail-account configuration, including changing the access permissions for their own mail folders so that they can share folders with other users.

You can use the provided forms unchanged, or you can customize them for your enterprise. Note that the forms are complex; making more than minor cosmetic changes (especially to the Mail Account Manager form) can be a difficult process, requiring sophisticated manipulation of HTML and JavaScript. Whether you customize them or not, you should leave the forms in their default locations (`serverRoot/bin/user/admin/html/`) on the server.

On the other hand, if you have already implemented end-user access to directory information with HTML forms of your own design, you can provide client access to those forms by using Netscape Console to specify their URLs.

In addition to controlling access to end-user forms, Messaging Server also allows you to create a greeting message to be sent to each new user.

To configure end-user access or create a new-user greeting:

1. Generate the HTML forms you need, or modify and use the forms provided with Messaging Server.
2. Store the forms in an appropriate location. The default location for the forms provided with Messaging Server is *serverRoot*/bin/user/admin/html/.
3. In Netscape Console, open the Messaging Server whose end-user access you want to configure.
4. Click the Configuration tab. If the server's icon in the left pane is not already highlighted, select it.
5. Click the "End user" tab in the right pane. The End User Configuration form appears.
6. Make changes to the form URLs as needed. The default URLs are consistent with the default locations of the HTML files.
7. Create a new-user greeting or make changes, as needed.

   You must format the greeting as an email message, with a header (containing at least a subject line), then a blank line, then the message body.

   When you create a message, specify its language with the drop-down list above the message field. You can create several messages in several languages, if desired. If you do, the locale of each new user is compared to the language of the message, and the server sends the correct message to the new user.
8. Click Save.

Configuring Default Languages

Users can create messages for the server to automatically send under certain specified conditions. For example, an "I am on vacation" message as an automatic reply to all incoming mail. When users create messages of this kind, they can specify that the message is written in a particular language. This

allows users to create different, language-specific versions of messages that the server is to send. (If no language is specified, the server assumes that the message is in English.)

Users can also choose a preferred language. In cases where there are language-specific versions of messages, notices, or prompts that the server automatically sends out, users will receive those messages in the language they have chosen as their preferred language if a version for their language exists on the server. For example, if someone has created multiple versions of an "I am on vacation" notice in English, Japanese, Spanish, and French, users who have chosen Spanish as their preferred language will receive the vacation notice in that language.

For notices and messages sent by the server, the server selects the language-specific version to send according to the following rules:

1. If the user to whom the message is being sent has chosen a preferred language, and a language-specific version of that message exists, that is the version that is sent. For example, if the user has chosen Japanese, and there is a Japanese version, the Japanese version is sent.

2. If the user has not chosen a preferred language, or has chosen a preferred language but there is no version of the message in that language, the version that matches the default domain language is sent. For example, if the default domain language is Spanish and the user has chosen French but there is no French version, the Spanish version is sent.

3. If there is no version of the message that matches either the user's preferred language or the default domain language, but there is an English-language version, the English version is sent. For example, if the default domain language is Spanish and the user has chosen German but there are only French and English versions of the message, the English version is sent.

A user's preferred language is stored in the domain's Directory Server. When the server sends messages to users outside of the server's domain it does not know what their preferred language is unless it is responding to an incoming message with a preferred language specified in the incoming message's header (`X-Accept-Language`):

- If there is a preferred language in an incoming message's header, the server accepts the preferred language specification from the message header.

- If there is no preferred language in the message header, the server treats that user as someone who has not chosen a preferred language.

All domains have a default preferred language which is determined as follows:

- If the domain name ends in a country code, the default domain language is the standard default language for that country. For example, by default all domains ending in `.jp` have Japanese as their preferred language.
- All domains with names ending in something other than a country code use English as their default preferred language. For example, the default language for domains with names ending in `.com`, `.org`, `.gov`, .`edu`, and `.mil` all have English as their preferred language.

As administrator, you can specify a different default preferred language for your domain or for a subdomain. When you specify a domain language, that language becomes the preferred language for all users in that domain except those that have individually chosen a different preferred language. When users choose a different preferred language, their choice overrides the default domain language.

Similarly, you can specify a default site language for all the domains and subdomains within your site. You can specify a language for the site as a whole, and different languages for individual domains and subdomains. A domain language overrides a site language, and a user's individually chosen preferred language overrides both site language a domain language. For example, if your default site language is French but you specify that a particular subdomain's default language is Spanish, users in that subdomain will receive language-specific notices in Spanish unless they individually specify some other preferred language.

Specifying a Site Language

To specify a site language:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab.
3. In the right pane, click the Language tab.

4. From the site language drop-down list, choose the language you wish to use.

5. Click Save.

Command Line

You can also specify a site language at the command line as follows:

```
configutil -o gen.sitelanguage -v value
```

where *value* is one of the local supported languages:

af	Afrikaans
ca	Catalan
da	Danish
de	German
en	English
es	Spanish
fi	Finnish
fr	French
ga	Irish
gl	Galician
is	Icelandic
it	Italian
ja	Japanese
nl	Dutch
no	Norwegian
pt	Portuguese
sv	Swedish

Specifying a Domain Language

To specify a domain language:

1. In Netscape Console, open the Messaging Server that you want to configure.

2. Click the Configuration tab.

3. In the right pane, click the Language tab.

4. Click the Add button.

   The Domain/Language Mapping window appears.

5. In the Domain/Subdomain field, enter the name of the domain.

   You must enter a fully qualified domain name. For example, `sales.airius.com`.

6. From the Preferred Language drop-down list, choose the language you wish to use for this domain.

7. Click OK to return to the Language tab.

   Your changes appear in the Domain/Language Mapping table.

8. Click Save to save your changes.

Your existing domain language settings are displayed in the Domain/Language Mapping Table beneath the Site Language pull-down menu. The left element of each entry is the domain name, the right element is the two-letter abbreviation of that domain's default language.

Keep in mind that the number of entries in this table has some effect on performance.

To change a domain or subdomain's default language:

1. Highlight the domain.

2. Click Edit.

   The Domain/Language Mapping window for that domain is displayed.

3. From the Preferred Language drop-down list, select a new language for this domain.

4. Click OK to return to the Language tab and view your changes in the Domain/Language Mapping table.

5. Click Save to save your changes.

Command Line

You can also specify a domain language at the command line as follows:

```
configutil -o service.smtp.domainlangtable -v string
```

where *string* maps domain names to languages. If *string* contains more than one domain, the string must be enclosed in quotes and each domain must be separated by the dollar sign character ($).

In the following example, the default language for the domain `airius.com` is English; the default language for the domain `jp.airius.com` is Japanese; the default language for the domain `fr.airius.com` is French:

```
configutil -o service.smtp.domainlangtable
   -v "airius.com en$jp.airius.com ja$fr.airius.com fr"
```

Starting and Stopping Services

Netscape Console provides a form that allows you to start and stop individual services and view status information about each service.

For each service--IMAP, POP, SMTP, and HTTP--the form displays the service's current state (on or off). If the service is running, the form shows the time at which the service was last started up, and it can also display other status information.

Note: You must first enable the POP, IMAP, and HTTP services before starting or stopping them. For more information, see "Enabling and Disabling Services" on page 70.

You only need to run the services that your server actually uses. For example, if you are temporarily using a particular instance of Messaging Server solely as a message transfer agent (MTA), you can turn on SMTP alone. Or, if maintenance, repair, or security needs require shutting down the server, you may be able to turn off just the affected service. (If you never intend to run a particular service, you should disable it instead of just turning it off.)

To start up, shut down, or view the status of any messaging services:

1. From Netscape Console, open the Messaging Server whose services you want to start or stop.
2. Get to the Services General Configuration form in either of these two ways:
   - Click the Tasks tab, then click "Start/Stop Services".
   - Click the Configuration tab and select the Services folder in the left pane. Then click the General tab in the right pane.
3. The Services General Configuration form appears.

   The left column of the Process Control field lists the services supported by the server; the right column gives the basic status of each of the services (ON or OFF, plus—if it is ON—the time it was last started).
4. To view status information about a service that is currently on, select the service in the Process Control field.

   The Service Status field displays status information about the service.

   For POP, IMAP, and HTTP the field shows the last connection time, the total number of connections, the current number of connections, the number of failed connections since the service last started, and the number of failed logins since the service last started.

   For SMTP, the field shows the current number of queued messages, the total number of messages sent and received since startup, and the current numbers of messages waiting for both external and internal delivery.

   The information in this field helps you to understand the load on the server and the reliability of its service, and it can help spotlight attacks against the server's security.
5. To turn a service on, select it in the Process Control field and click Start.
6. To turn a service off, select it in the Process Control field and click Stop.
7. To turn all enabled services on or off simultaneously, click the Start All or Stop All button.

On Windows NT, you can also use the Services Control Manager to start and stop services. On Unix platforms, you can use the `NscpMsg` utility, as described below.

Important: If a server process crashes, other processes will hang as they wait for locks held by the server process that crashed. Therefore, if any server process crashes, you should stop all processes, then restart all processes. This includes the POP, IMAP, HTTP, and SMTP processes, as well as the `stored` (message store) process, and any utilities that modify the message store, such as `mboxutil`, `quota`, `deliver`, `reconstruct`, `readership`, or `upgrade`.

Command Line

On Unix platforms you can use the `NscpMsg` utility to start or stop any of the messaging services, as shown in the following example.

```
/etc/NscpMsg start imap
/etc/NscpMsg stop pop
```

For more information, see "NscpMsg" on page 423.

Customizing Directory Lookups

Netscape Messaging Server cannot function without an LDAP-based directory system such as the Netscape Directory Server. Messaging Server and Netscape Console require directory access for three purposes:

- When you first install a Messaging Server, you enter configuration settings for the server. These settings are stored in a central *configuration directory*. Part of the installation process includes configuring the connection to that directory.
- When you create or update account information for mail users or mail groups, you enter that information through the Users and Groups interface of Netscape Console. The information is stored in a directory called the *user directory*. Your server group's Administration Server is configured at installation so that when you access Users and Groups, Netscape Console connects by default to the user directory that defines your *administrative domain*—the set of Netscape servers that all share the same configuration directory and user directory.
- When routing messages and delivering mail to mailboxes, Messaging Server looks up information about the sender or recipient(s) in the user directory. By default, Messaging Server looks in the same user directory that its Administration Server has been configured to use.

You can modify each of these directory-configuration settings in the following ways:

- The Administration Server interface of Netscape Console lets you change the connection settings for the configuration directory. (For more information, see the Administration Server chapter of *Managing Servers with Netscape Console*.)
- The Users and Groups interface of Netscape Console lets you temporarily connect to a different user directory from the default when making changes to user and group information. (For more information, see the Users and Groups chapter of *Managing Servers with Netscape Console*.)
- The Messaging Server interface of Netscape Console lets you configure your Messaging Server to connect to a different user directory from the default defined by the Administration Server. This is the configuration task discussed in this section.

Reconfiguring your Messaging Server to connect to a different user directory for user and group lookups is strictly optional. In most cases, the user directory that defines your server's administrative domain is the one used by all servers in the domain.

Note: If you specify a custom user directory for your Messaging Server lookups, you must also specify that same directory whenever you access the Users and Groups interface of Netscape Console to make changes to the directory's user or group information. For more information, see Chapter 4, "Managing Mail Users and Mailing Lists."

To modify the Messaging Server LDAP user-lookup settings:

1. From Netscape Console, open the Messaging Server whose LDAP connection you want to customize.
2. Click the Configuration tab.
3. Select the Services folder in the left pane.
4. Select the LDAP tab in the right pane. The LDAP form appears.

The LDAP form displays the configuration settings for both the configuration directory and the user directory. The configuration-directory settings, however, are read-only in this form. See the Administration Server chapter of *Managing Servers with Netscape Console* if you need to change them.

5. To change the user-directory connection settings, click the box labeled "Use messaging server specific directory settings".

6. Update the LDAP configuration by entering or modifying any of the following information (for explanations of directory concepts, including definitions of terms such as *distinguished name,* see the *Directory Server Administrator's Guide*):

   Host name: The name of the host machine on which the directory containing your installation's user information resides. This is typically not the same as the Messaging Server host, although for very small installations it might be.

   Port number: The port number on the directory host that Messaging Server must use to access the directory for user lookup. This number is defined by the directory administrator, and may not necessarily be the default port number (389).

   Bind DN: The distinguished name that your Messaging Server uses to represent itself when it connects to the directory server for lookups. The bind DN must be the distinguished name of an entry in the user directory itself that has been given search privileges to the user portion of the directory. If the directory allows anonymous search access, you can leave this entry blank.

   Base DN: The search base—the distinguished name of a directory entry that represents the starting point for user lookups. To speed the lookup process, the search base should be as close as possible in the directory tree to the information being sought. If your installation's directory tree has a "people" or "users" branch, that is a reasonable starting point.

7. To change the password used, in conjunction with the Bind DN, to authenticate this Messaging Server to the LDAP directory for user lookups, click the Change password button. A Password-Entry window opens, into which you can enter the updated password.

Your own security policies should determine what password you use in this situation. Initially, the password is set to no password. The password is not used if you have specified anonymous access by leaving the Bind DN field blank.

To return to using the default user directory, uncheck the "Use messaging server specific directory settings" box.

Command Line

You can also set values for the user-directory connection settings at the command line as follows:

To specify whether to use messaging server specific directory settings:

```
configutil -o local.ugldapuselocal -v [ yes | no ]
```

To specify the LDAP host name for user lookup:

```
configutil -o local.ugldaphost -v name
```

To specify the LDAP port number for user lookup:

```
configutil -o local.ugldapport -v number
```

To specify the LDAP base DN for user lookup:

```
configutil -o local.ugldapbasedn -v basedn
```

To specify the LDAP bind DN for user lookup:

```
configutil -o local.ugldapbinddn -v binddn
```

Encryption Settings

You can use Netscape Console to enable Secure Sockets layer (SSL) encryption and authentication for Messaging Server and to select the specific encryption ciphers that the server will support across all of its services.

Although this task is a general configuration task, it is described in "Enabling SSL" on page 181. That section is part of Chapter 6, "Security and Access Control," which also contains background information on all security and access-control topics for Messaging Server.

Where to Go from Here

This chapter has provided background information on messaging deployment and Messaging Server installation, and it has described how to make general configuration settings to Messaging Server. Subsequent chapters in this book describe the bulk of the administrative tasks, from configuring services through setting up users and groups to monitoring and maintaining the server.

To perform the following tasks, go to the chapters or appendixes indicated.

To Configure Services

- Chapter 2, "Configuring POP, IMAP, and HTTP Services"
- Chapter 3, "Configuring SMTP Services"
- Chapter 5, "Managing the Message Store"
- Chapter 6, "Security and Access Control"
- Chapter 9, "Message Routing"
- Chapter 12, "Program Delivery"
- Chapter 13, "Messaging Multiplexor"
- Appendix A, "Command-line Utilities"

To Set Up Mail Users and Mailing Lists

- Chapter 4, "Managing Mail Users and Mailing Lists"
- Appendix B, "sendmail Migration and Compatibility"

To Work with Plug-ins and UBE Filters

- Chapter 7, "Working with SMTP Plug-Ins"
- Chapter 8, "Filtering Unsolicited Bulk Email"

To Monitor and Maintain the Server

- Chapter 10, "Monitoring and Maintaining Your Server"
- Chapter 11, "Logging and Log Analysis"
- Appendix C, "SNMP MIB"
- *Netscape Mailstone Utility*

Chapter

2

Configuring POP, IMAP, and HTTP Services

Netscape Messaging Server supports the Post Office Protocol 3 (POP3), the Internet Mail Access Protocol 4 (IMAP4), and the HyperText Transfer Protocol (HTTP) for client access to mailboxes. IMAP and POP are both Internet-standard mailbox protocols. Messenger Express, a web-enabled electronic mail program, lets end users access their mailboxes using a browser running on an Internet-connected computer system using HTTP.

This chapter describes how to use Netscape Console to configure your server to support one or more of these services. For information on configuring Simple Mail Transfer Protocol (SMTP) services, see Chapter 3, "Configuring SMTP Services."

You can also perform many POP, IMAP, and HTTP configuration tasks through the command-line utility `configutil`. For instructions on how to use `configutil`, see Appendix A, "Command-line Utilities."

This chapter has the following sections:

- General Configuration
- Login Requirements
- Performance Parameters
- Client Access Controls
- Configuring POP Services
- Configuring IMAP Services

- Configuring HTTP Services
- Customizing HTTP Services

General Configuration

Configuring the general features of the Messaging Server POP, IMAP, and HTTP services includes enabling or disabling the services, assigning port numbers, and optionally modifying service banners sent to connecting clients. This section provides background information; for the steps you follow to make these settings, see "Configuring POP Services" on page 78, "Configuring IMAP Services" on page 80, and "Configuring HTTP Services" on page 82.

Enabling and Disabling Services

You can control whether any particular instance of Messaging Server makes its POP, IMAP, or HTTP service available for use. This is not the same as starting and stopping services (see "Starting and Stopping Services" on page 61); to function, POP, IMAP, or HTTP must be both enabled and started.

Enabling a service is a more "global" process than starting or stopping a service. For example, the Enable setting persists across system reboots, whereas you must restart a previously "stopped" service after a reboot.

There is no need to enable services that you do not plan to use. For example, if a Messaging Server instance is used only as a message transfer agent (MTA), you should disable POP, IMAP, and HTTP. If it is used only for POP services, you should disable IMAP and HTTP. If it used only for web-based email, you should disable both POP and IMAP.

Specifying Port Numbers

For each service, you can specify the port number that the server is to use for service connections:

- If you enable the POP service, you can specify the port number that the server is to use for POP connections. The default is 110.

- If you enable the IMAP service, you can specify the port number that the server is to use for IMAP connections. The default is 143.
- If you enable the HTTP service, you can specify the port number that the server is to use for HTTP connections. The default is 80.

You might need to specify a port number other than the default if you have, for example, two or more IMAP server instances on a single host machine, or if you are using the same host machine as both an IMAP server and a Messaging Multiplexor server. (For information about the Multiplexor, see Chapter 13, "Messaging Multiplexor.")

Keep the following in mind when you specify a port:

- Port numbers can be any number from 1 to 65535.
- Make sure the port you choose isn't already in use or reserved for another service.

Ports for Encrypted Communications

Messaging Server supports encrypted communications with IMAP and HTTP clients by using the Secure Sockets Layer (SSL) protocol. For general information on support for SSL in Messaging Server, see "Configuring SSL Encryption and Authentication" on page 175.

IMAP Over SSL

You can accept the default IMAP over SSL port number (993) or you can specify a separate port for IMAP over SSL.

Messaging Server provides the option of using separate ports for IMAP and IMAP over SSL because most current IMAP clients require separate ports for them. Same-port communication with both IMAP and IMAP over SSL is an emerging standard; as long as your Messaging Server has an installed SSL certificate (see "Obtaining Certificates" on page 177), it can support same-port IMAP over SSL.

HTTP Over SSL

You can accept the default HTTP over SSL port number (443) or you can specify a separate port for HTTP.

Service Banner

When a client first connects to the Messaging Server POP or IMAP port, the server sends an identifying text string to the client. This service banner (not normally displayed to the client's user) identifies the server as Netscape Messaging Server, and gives the server's version number. The banner is most typically used for client debugging or problem-isolation purposes.

You can replace the default banner for the POP or IMAP service if you want a different message sent to connecting clients.

You can use Netscape Console or the `configutil` utility to set service banners. For information about `configutil`, see "configutil" on page 406.

Login Requirements

You can control how users are permitted to log in to the POP, IMAP, or HTTP service to retrieve mail. You can allow anonymous login (for IMAP or HTTP services), password-based login (for all services), and certificate-based login (for IMAP or HTTP services). This section provides background information; for the steps you follow to make these settings, see "Configuring POP Services" on page 78, "Configuring IMAP Services" on page 80, or "Configuring HTTP Services" on page 82.

Anonymous Login

Anonymous login refers to a user logging in under the special user name `anonymous`, which requires no password. (By convention analogous to that of FTP, users enter their email addresses as passwords, so that their accesses are logged.) One reason for permitting anonymous login might be to provide read-only access to, for example, archived messages of a mailing list or to shared IMAP or HTTP folders.

By default, anonymous login for IMAP and HTTP is disabled. Anonymous login is not available for the POP service.

Password-Based Login

In typical messaging installations, users access their POP, IMAP, or HTTP mailboxes by entering a password into their mail client. The client sends the password to the server, which uses it to authenticate the user. If the user is authenticated, the server decides, based on access-control rules, whether or not to grant the user access to certain mailboxes stored on that server.

If you allow password login, users can access POP, IMAP, or HTTP by entering a password. (Password-based login is the only authentication method for POP services.) Passwords are stored in an LDAP directory. Directory policies determine what password policies, such as minimum length, are in effect.

If you disallow password login for IMAP or HTTP services, password-based authentication is not permitted. Users are then required to use certificate-based login, as described in the next section.

To increase the security of password transmission for IMAP and HTTP services, you can require that passwords be encrypted before they are sent to your server. You do this by selecting a minimum cipher-length requirement for login.

- If you choose 0, you do not require encryption. Passwords are sent in the clear or they are encrypted, depending on client policy.
- If you choose a nonzero value, the client must establish an SSL session with the server—using a cipher whose key length is at least the value you specify—thus encrypting any IMAP or HTTP user passwords the client sends.

If the client is configured to require encryption with key lengths greater than the maximum your server supports, or if your server is configured to require encryption with key lengths greater than what the client supports, password-based login cannot occur. For information on setting up your server to support various ciphers and key lengths, see “Enabling SSL” on page 181.

Certificate-Based Login

In addition to password-based authentication, Netscape servers support the authentication of users through examination of their digital certificates. Instead of presenting a password, the client presents the user's certificate when it establishes an SSL session with the server. If the certificate is validated, the user is considered authenticated.

For instructions on setting up Messaging Server to accept certificate-based user login to the IMAP or HTTP service, see "Setting Up Certificate-Based Login" on page 184.

You don't need to uncheck the "Allow password login" box in the IMAP or HTTP System form to enable certificate-based login. If the box is checked (its default state), and if you have performed the tasks required to set up certificate-based login, both password-based and certificate-based login are supported. Then, if the client establishes an SSL session and supplies a certificate, certificate-based login is used. If the client does not use SSL or does not present a client certificate, it will send a password instead.

Performance Parameters

You can set some of the basic performance parameters for the POP, IMAP, and HTTP services of Messaging Server. Based on your hardware capacity and your user base, you can adjust these parameters for maximum efficiency of service. This section provides background information; for the steps you follow to make these settings, see "Configuring POP Services" on page 78, "Configuring IMAP Services" on page 80, or "Configuring HTTP Services" on page 82.

Number of Processes

Messaging Server can divide its work among several executing processes, which in some cases can increase efficiency. This capability is especially useful with multiprocessor server machines, in which adjusting the number of server processes can allow more efficient distribution of multiple tasks among the hardware processors.

There is a performance overhead, however, in allocating tasks among multiple processes and in switching from one process to another. The advantage of having multiple processes diminishes with each new one added. A simple rule of thumb for most configurations is to have one process per hardware processor on your server machine, up to a maximum of perhaps 4 processes. Your optimum configuration may be different; this rule of thumb is meant only as a starting point for your own analyses.

Note: On some platforms you might also want to increase the number of processes to get around certain per-process limits (such as the maximum number of file descriptors), specific to that platform, that may affect performance.

The default number of processes is 1 each for the POP, IMAP, or HTTP service.

Number of Connections per Process

The more simultaneous client connections your POP, IMAP, or HTTP service can maintain, the better it is for clients. If clients are denied service because no connections are available, they must then wait until another client disconnects.

On the other hand, each open connection consumes memory resources and makes demands on the I/O subsystem of your server machine, so there is a practical limit to the number of simultaneous sessions you can expect the server to support. (You might be able to increase that limit by increasing server memory or I/O capacity.)

IMAP, HTTP, and POP have different needs in this regard:

- IMAP connections are generally long-lived compared to POP and HTTP connections. When a user connects to IMAP to download messages, the connection is usually maintained until the user quits or the connection times out. In contrast, a POP or HTTP connection is usually closed as soon as the POP or HTTP request has been serviced.

- IMAP and HTTP connections are generally very efficient compared to POP connections. Each POP reconnection requires re-authentication of the user. In contrast, an IMAP connection requires only a single authentication because the connection remains open for the duration of the IMAP session (login to logout). An HTTP connection is short, but the user need not reauthenticate for each connection because multiple connections are allowed for each HTTP session (login to logout). POP connections,

therefore, involve much greater performance overhead than IMAP or HTTP connections. Netscape Messaging Server, in particular, has been designed to require very low overhead by open but idle IMAP connections and by multiple HTTP connections.

Note: For more information about HTTP session security, see "About HTTP Security" on page 171.

Thus, at a given moment for a given user demand, Messaging Server may be able to support many more open IMAP or HTTP connections than POP connections.

The default value for IMAP is 4000; the default value for HTTP is 6000 connections per process; the default value for POP is 600. These values represent roughly equivalent demands that can be handled by a typically configured server machine. Your optimum configuration may be different; these defaults are meant only as general guidelines.

Number of Threads per Process

Besides supporting multiple processes, Messaging Server further improves performance by subdividing its work among multiple threads. The server's use of threads greatly increases execution efficiency, because commands in progress are not holding up the execution of other commands. Threads are created and destroyed, as needed during execution, up to the maximum number you have set.

Having more simultaneously executing threads means that more client requests can be handled without delay, so that a greater number of clients can be serviced quickly. However, there is a performance overhead to dispatching among threads, so there is a practical limit to the number of threads the server can make use of.

For POP, IMAP, and HTTP, the default maximum value is 250 threads per process. The numbers are equal despite the fact that the default number of connections for IMAP and HTTP is greater than for POP. It is assumed that the more numerous IMAP and HTTP connections can be handled efficiently with the same maximum number of threads as the fewer, but busier, POP connections. Your optimum configuration may be different, but these defaults are high enough that it is unlikely you would ever need to increase them; the defaults should provide reasonable performance for most installations.

Dropping Idle Connections

To reclaim system resources used by connections from unresponsive clients, the IMAP4, POP3, and HTTP protocols permit the server to unilaterally drop connections that have been idle for a certain amount of time.

The respective protocol specifications require the server to keep an idle connection open for a minimum amount of time. The default times are 10 minutes for POP, 30 minutes for IMAP, 3 minutes for HTTP. You can increase the idle times beyond the default values, but you cannot make them less.

If a POP or IMAP connection is dropped, the user must reauthenticate to establish a new connection. In contrast, if an HTTP connection is dropped, the user need not reauthenticate because the HTTP session remains open. For more information about HTTP session security, see “About HTTP Security” on page 171.

Idle POP connections are usually caused by some problem (such as a crash or hang) that makes the client unresponsive. Idle IMAP connections, on the other hand, are a normal occurrence. To keep IMAP users from being disconnected unilaterally, IMAP clients typically send a command to the IMAP server at some regular interval that is less than 30 minutes.

Logging Out HTTP Clients

An HTTP session can persist across multiple connections. HTTP clients are not logged out when a connection is dropped. However, if an HTTP session remains idle for a specified time period, the server will automatically drop the HTTP session and the client is logged out (the default time period is 2 hours). When the session is dropped, the client’s session ID becomes invalid and the client must reauthenticate to establish another session. For more information about HTTP security and session ID’s, see “About HTTP Security” on page 171.

Client Access Controls

Netscape Messaging Server includes access-control features that allow you to determine which clients can gain access to its POP, IMAP, or HTTP messaging services (and SMTP as well). You can create flexible access filters that allow or deny access to clients based on a variety of criteria.

Client access control is an important security feature of Netscape Messaging Server. For information on creating client access-control filters and examples of their use, see "Configuring Client Access to TCP Services" on page 189.

Configuring POP Services

You can perform basic configuration of the Messaging Server POP service through Netscape Console. To configure your POP service:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Select POP.
4. Click the System tab in the right pane.
5. To enable the service, check the box labeled "Enable POP service at port" and assign a port number.

   For more information, see "Enabling and Disabling Services" on page 70 and "Specifying Port Numbers" on page 70.
6. Specify connection settings as follows:

   - Set the maximum number of network connections per process. For more information, see "Number of Connections per Process" on page 75.
   - Set the maximum idle time for connections. For more information, see "Dropping Idle Connections" on page 77.

7. Specify process settings as follows:
   - Set the maximum number of threads per process. For more information, see "Number of Threads per Process" on page 76.
   - Set the maximum number of processes. For more information, see "Number of Processes" on page 74.
8. If desired, in the POP service banner field, specify a service banner.
9. Click Save.

Note: For the POP service, password-based login is automatically enabled.

Command Line

You can also set values for these attributes at the command line as follows:

To enable or disable the POP service:

```
configutil -o service.pop.enable -v [ yes | no ]
```

To specify the port number:

```
configutil -o service.pop.port -v number
```

To set the maximum number of network connections per process:

```
configutil -o service.pop.maxsessions -v number
```

To set the maximum idle time for connections:

```
configutil -o service.pop.idletimeout -v number
```

To set the maximum number of threads per process:

```
configutil -o service.pop.maxthreads -v number
```

To set the maximum number of processes:

```
configutil -o service.pop.numprocesses -v number
```

To specify a protocol welcome banner:

```
configutil -o service.pop.banner -v banner
```

Configuring IMAP Services

You can perform basic configuration of the Messaging Server IMAP service through Netscape Console. To configure your IMAP service:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Select IMAP.
4. Click the System tab in the right pane.
5. To enable the service, check the box labeled "Enable IMAP service at port" and assign a port number.

   For more information, see "Enabling and Disabling Services" on page 70 and "Specifying Port Numbers" on page 70.
6. If desired, enable anonymous login by checking the "Allow anonymous login" checkbox.

   For more information, see "Anonymous Login" on page 72.
7. If desired, enable password-based login.

   For more information, see "Password-Based Login" on page 73.
8. Specify connection settings as follows:

   - Set the maximum number of network connections per process. For more information, see "Number of Connections per Process" on page 75.
   - Set the maximum idle time for connections. For more information, see "Dropping Idle Connections" on page 77.
9. Specify process settings as follows:

   - Set the maximum number of threads per process. For more information, see "Number of Threads per Process" on page 76.
   - Set the maximum number of processes. For more information, see "Number of Processes" on page 74.

10. If desired, in the IMAP service banner field, specify a service banner.

11. Click Save.

Command Line

You can also set values for the IMAP attributes at the command line as follows:

To enable or disable the IMAP service:

```
configutil -o service.imap.enable -v [ yes | no ]
```

To specify the port number:

```
configutil -o service.imap.port -v number
```

To enable a separate port for IMAP over SSL:

```
configutil -o service.imap.enablesslport -v [ yes | no ]
```

To specify a port number for IMAP over SSL:

```
configutil -o service.imap.sslport -v number
```

To allow anonymous login to the IMAP service:

```
configutil -o service.imap.allowanonymouslogin -v [ yes | no ]
```

To enable or disable password login to the IMAP service:

```
configutil -o service.http.plaintextmincipher -v value
```

where *value* is one of the following:

```
 -1 - Disables password login
  0 - Enables password login without encryption
 40 - Enables password login and specifies an encryption strength
128 - Enables password login and specifies an encryption strength
```

To set the maximum number of network connections per process:

```
configutil -o service.imap.maxsessions -v number
```

To set the maximum idle time for connections:

```
configutil -o service.imap.idletimeout -v number
```

To set the maximum number of threads per process:

```
configutil -o service.imap.maxthreads -v number
```

To set the maximum number of processes:

```
configutil -o service.imap.numprocesses -v number
```

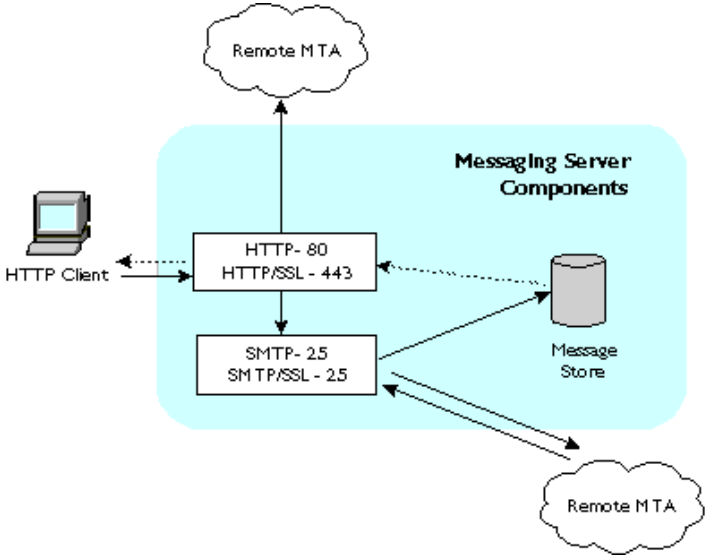
To specify a protocol welcome banner:

```
configutil -o service.imap.banner -v banner
```

Configuring HTTP Services

POP and IMAP clients send mail directly to the Netscape Messaging Server MTA for routing or delivery. In contrast, HTTP clients send mail to a specialized web server that is part of Netscape Messaging Server. The HTTP service then sends the message to the local MTA or to a remote MTA for routing or delivery, as shown in Figure 2.1 .

Figure 2.1 HTTP Service Components



Many of the HTTP configuration parameters are similar to the parameters available for the POP and IMAP services. These include parameters for anonymous login, connection settings, and process settings. Some parameters are specific to the HTTP service; these include parameters for message settings and MTA settings.

Message Settings. When an HTTP client constructs a message with attachments, the attachments are uploaded to the server and stored in a file. The HTTP service retrieves the attachments and constructs the message before sending the message to an MTA for routing or delivery. You can accept the default attachment spool directory or specify an alternate directory. You can also specify a maximum size allowed for attachments.

MTA Settings. By default, the HTTP service sends outgoing web mail to the local MTA for routing or delivery. You might want to configure the HTTP service to send mail to a remote MTA, for example, if your site is a hosting service and most recipients are not in the same domain as the local host machine. To send web mail to a remote MTA, you need to specify the remote host name and the SMTP port number for the remote host.

To configure your HTTP service:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Select HTTP.
4. Click the System tab in the right pane.
5. To enable the service, check the box labeled "Enable HTTP service at port" and assign a port number.

   For more information, see "Enabling and Disabling Services" on page 70 and "Specifying Port Numbers" on page 70.
6. If desired, enable anonymous login by checking the "Allow anonymous login" checkbox.

   For more information, see "Anonymous Login" on page 72.
7. If desired, enable password-based login.

   For more information, see "Password-Based Login" on page 73
8. Specify connection settings as follows:

   - Set the maximum number of network connections per process. For more information, see "Number of Connections per Process" on page 75.

- Set the maximum idle time for connections. For more information, see "Dropping Idle Connections" on page 77.
- Set the maximum idle time for client sessions. For more information, see "Logging Out HTTP Clients" on page 77.

9. Specify process settings as follows:

- Set the maximum number of threads per process. For more information, see "Number of Threads per Process" on page 76.
- Set the maximum number of processes. For more information, see "Number of Processes" on page 74.

10. Specify Message settings as follows:

- If desired, specify the attachment spool directory.
- If desired, specify the maximum attachment size.

For more information, see Message Settings.

11. Specify MTA settings as follows:

- If desired, specify an alternate MTA host name.
- If required, specify an alternate MTA port.

For more information, see MTA Settings.

12. Click Save.

Command Line

You can also set values for the HTTP attributes at the command line as follows:

To enable or disable the HTTP service:

```
configutil -o service.http.enable -v [ yes | no ]
```

To specify the port number:

```
configutil -o service.http.port -v number
```

To enable a separate port for HTTP over SSL:

```
configutil -o service.http.enablesslport -v [ yes | no ]
```

To specify a port number for HTTP over SSL:

```
configutil -o service.http.sslport -v number
```

To enable anonymous login:

```
configutil -o service.http.allowanonymouslogin -v [ yes | no ]
```

To enable or disable password login:

```
configutil -o service.http.plaintextmincipher -v value
```

where *value* is one of the following:

```
 -1 - Disables password login
  0 - Enables password login without encryption
 40 - Enables password login and specifies an encryption strength
128 - Enables password login and specifies an encryption strength
```

To set the maximum number of network connections per process:

```
configutil -o service.http.maxsessions -v number
```

To set the maximum idle time for connections:

```
configutil -o service.http.idletimeout -v number
```

To set the maximum idle time for client sessions:

```
configutil -o service.http.sessiontimeout -v number
```

To set the maximum number of threads per process:

```
configutil -o service.http.maxthreads -v number
```

To set the maximum number of processes:

```
configutil -o service.http.numprocesses -v number
```

To specify the attachment spool directory for client outgoing mail:

```
configutil -o service.http.spooldir -v dirpath
```

To specify the maximum attachment size:

```
configutil -o service.http.maxmessagesize -v size
```

where *size* is a number in bytes.

To specify an alternate MTA host name:

```
configutil -o service.http.smtphost -v hostname
```

To specify the port number for the alternate MTA host name:

```
configutil -o service.http.smtpport -v portnum
```

Customizing HTTP Services

You can customize HTTP services for Messenger Express. Customizing Messenger Express features requires knowledge of JavaScript and HTML.

To customize the look and feel of the features, modify the following HTML documents located in *server-root*/msg-*instance*/html/.

Table 2.1 How to customize the Messenger Express interface

Feature to Customize	HTML Document(s) to Modify
Attachments	attach_fs.html
Collect mail from another server	collect_fs.html
Message Composition	comp_fs.html
Folder Tab	fldr_fs.html
Address Lookup	ldap_fs.html
Mailbox Tab	mbox_fs.html
Message Tab	msg_fs.html
Options Tab	opts_fs.html
Return Receipt	receipt_fs.html

Messenger Express lets you work with the above .html documents as well as three .js files. The three Javascript files are: main.js, util.js. and i18n.js. For example, to change colors you create a new color_sets array in main.js. To set up common variables, use the util.js file. To change user interface (UI) prompts, labels, and other text, modify the i18n.js file. For example, if you want to change the expunge label from "Expunge" to "Shred", you would do so in the i18n.js file.

While you can certainly alter all of the files, you might find that by simply changing color, company logo, and localization you can achieve the look and feel you want with minimal effort.

The following describes how to change the color, banner, and localization features:

Color. To customize the color scheme, create a `color_sets` array file in `main.js`. Another way to change the color scheme is to modify one of the two existing arrays to include the desired colors.

Company Banner (Logo). To add a logo, modify the `brand` variable in `main.js` to include the desired logo.

Localization. To localize the user interface, copy `mail-en.html` to a new file named `mail-`*`xx`*`.html`, where *`xx`* is the two letter abbreviation for a specific language. Translate all the string values associated with the i18n array elements. To localize the online help, modify the `help.html`.

To ensure security and convenience, each page of the Messenger Express application must include the following lines:

```
<script src="util.js"></script>

<script>

init('parent')

</script>
```

The parameter to `init()` is the path to the main application frame, which contains `mail.html`.

Chapter

3

Configuring SMTP Services

This chapter describes how to configure SMTP services for your server using Netscape Console. For information on how to configure the POP, IMAP, or HTTP services, see Chapter 2, "Configuring POP, IMAP, and HTTP Services."

You can also perform many SMTP configuration tasks through the command-line utility `configutil`. For instructions on how to use `configutil`, see Appendix A, "Command-line Utilities."

This chapter contains the following sections:

- About SMTP
- Viewing and Configuring Domain Information
- Specifying Delivery Options
- Verifying Recipient Addresses
- Performing Reverse IP Address Lookups
- Specifying the Number of MTA Hops
- Reserving Free Disk Space for the Message Queue
- Enabling Optional SMTP Features
- Specifying Automatic Reply Information
- Specifying Error Handling
- Specifying Routing and Addressing Information
- Controlling Access to SMTP Services
- Working with SMTP Plugins

- Message Queue Concepts
- Specifying Actions on Deferred Queues
- Specifying Message Handling for Deferred Queues
- Specifying Alternate Paths for Physical Queues

About SMTP

Netscape Messaging Server supports the Internet-standard Simple Mail Transfer Protocol (SMTP). SMTP is the protocol most commonly used by the Internet to define how email is transferred between computers.

User Agents (UAs), such as Netscape Communicator, use SMTP to send mail to a Message Transfer Agent (MTA). MTAs use SMTP to route messages to other MTAs within a network.

Netscape Messaging Server listens for incoming mail on port 25 by default, the standard port for SMTP services. Incoming mail can arrive from a local mail client (UA) or from a remote MTA. For detailed concepts about how Netscape Messaging Server receives and routes messages, see Chapter 9, "Message Routing."

Viewing and Configuring Domain Information

A domain identifies a site on the Internet. Messaging servers use the domain name in an email address to route messages throughout the Internet. Every email message must contain a domain name in its address.

Each Messaging Server is responsible for a particular domain or domains. These domains are considered local to Messaging Server. If a server receives a message without a specified domain name, the server will complete the address by adding a domain name to the address. If Messaging Server receives mail for a remote domain, it attempts to route the message to a remote MTA.

For more information about domains, the Domain Name System (DNS), and how messages are routed, see Chapter 9, "Message Routing."

To view and configure information about domains:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Select SMTP.
4. Click the System tab in the right pane.

From this window, you can perform the following tasks:

- Specifying an Address Completion Domain
- Specifying the Domains Local to Your Server

Specifying an Address Completion Domain

If Messaging Server receives a message that does not contain a domain name in the recipient address, it will add the domain name to the address to complete the address. You can specify the domain name to be used for address completion. If you do not specify a domain, the fully-qualified host name of the machine on which Messaging Server resides is used to complete the address.

To specify an address completion domain:

1. Go to the SMTP System tab.
2. In the "Address completion domain" field, type the name of the DNS domain that will be used to complete a recipient address if the address does not contain a domain name.
3. Click Save.

Command Line

You can also specify an address completion domain at the command line as follows:

```
configutil -o service.smtp.defaultdomain -v domainname
```

Specifying the Domains Local to Your Server

A domain is local to your server if Messaging Server knows the recipient addresses in the domain. Messaging Server identifies a recipient address as local if the domain part of the address matches one of the following:

- The name of the host on which Messaging Server resides
- A local domain setting

If a message is sent to a local domain, but the recipient cannot be found in the directory, Messaging Server will bounce the message. Otherwise, the server will either deliver the message to a local mailbox or route the message to another server.

The server also checks the local domain configuration before it uses the "user ID" search method when that search method is enabled (see "Specifying Alternate Search Methods" on page 110). The server checks to see if the domain in the address is configured as a local domain; if the domain is local, the server will use the "user ID" search method if configured to do so.

To specify the domains local to your server:

1. Go to the SMTP System tab.
2. Click the Add button beside the "Local domain" field.
3. Type the domain you want to add.
4. Click OK to add the domain to the list of local domains in the SMTP System window.

   Mail sent to an unknown recipient at any of these domains is bounced.
5. When you finished adding domain information, click Save in the SMTP System window.

   Note that changes are not saved until you click Save in the SMTP System window.

Command Line

You can also specify the domains local to your server at the command line as follows:

```
configutil -o service.smtp.smtp-router.localmaildomains -v domainnames
```

where *domainnames* is a space-separated list of domain names.

Specifying Delivery Options

You can specify the following delivery options for messages sent to your server:

- Delivering Mail to Unix Mail Folders
- Delivering Mail to a Program
- Deferring Delivery

Delivering Mail to Unix Mail Folders

For user's who have a Unix system account on the Messaging Server host machine, Messaging Server can deliver mail to the user's local Unix mail folder. You specify the Unix mail delivery program to which Messaging Server should deliver mail.

For users to use this feature, you must enable this feature for the user account (see Chapter 4, "Managing Mail Users and Mailing Lists") and the user must turn on this option for their accounts (specified in the end user account management form).

Unix delivery is available only to users with a system account on the Messaging Server host (in addition to the Messaging Server account).

To specify a Unix mail delivery program:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Select SMTP.

4. Click the System tab in the right pane.

5. In the "Local mail delivery program" field, type the path of the Unix mail delivery program to which Messaging Server should deliver mail for accounts with the Unix-delivery option enabled.

   For example: `/usr/bin/mail`

6. Click Save.

Command Line

You can also set the Unix mail delivery program at the command line as follows:

```
configutil -o service.smtp.unix-deliver.maildeliveryprogram
   -v progpath
```

Delivering Mail to a Program

By default, messages are delivered to an account inbox. Program delivery allows messages to be delivered to external programs, such as filtering programs, file server programs, and so on.

When you or a user specifies program delivery as an account option, one or more programs are run whenever mail addressed to that account is received. Messaging Server starts the program and delivers mail to the program.

On Unix platforms, for security reasons, Messaging Server never runs any program as "root." To enable program delivery for the root account on Unix, you must specify a safe ID for root. If a root user enables the program delivery option in the server account management forms, mail sent to root will be handled by one or more programs running under the safe ID for root. If you do not specify a safe ID, program delivery for the root account will fail and the server will bounce messages sent to programs set up for the root account.

On Windows NT, programs are run under the server account specified at installation time (the System account by default). If you wish, you can specify that programs run under another account.

For more information about setting up and enabling program delivery, see Chapter 12, "Program Delivery."

To specify a safe ID for root (Unix platforms) or to specify the Windows NT account under which the program will run:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Select SMTP.
4. Click the System tab in the right pane.
5. Specify information for the following fields:

   For Unix platforms:

   Safe user ID for running programs. In this field, type the safe Unix user ID for running programs set up for the root account.

   Safe group ID for running programs. In this field, type the safe Unix group ID for running programs set up for the root account. The safe Unix user ID should be a member of the safe group ID.

   For Windows NT:

   NT account to run program as. In this field, type the user ID under which programs will run.

   Password for this account. In this field, type the password for the NT account.
6. Click Save.

Command Line

You can also set these values at the command line as follows:

Unix

```
configutil -o service.smtp.prog-deliver.defaultuid -v uid
configutil -o service.smtp.prog-deliver.defaultgid -v groupid
```

NT

```
configutil -o service.smtp.prog-deliver.ntaccount -v uid
configutil -o service.smtp.prog-deliver.ntpassword -v password
```

Deferring Delivery

By default, Messaging Server attempts to deliver messages immediately; the server queues mail only if there is a problem. You can specify that Messaging Server queue all outgoing mail and attempt delivery only when it processes the message queue. The server processes the message queue on intervals you indicate. For more information, see "Message Queue Concepts" on page 114.

This option is most useful for businesses that do not maintain a continuous connection to the Internet, but use dial-up connections instead. For example, Messaging Server can dial out to a remote host and then process the mail queue for the remote host.

To specify deferred delivery, go to the SMTP Accept tab:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Select SMTP.
4. Click the Accept tab in the right pane.
5. Check the "Defer delivery to remote hosts" box.
6. Click Save.

If you are specifying deferred delivery, you might also want to turn on the SMTP command, ETRN, to enable requests for deferred queue processing. With deferred queue processing, when a client (in this case, another MTA) connects to the server to send a message, it can also initiate processing of the deferred queue for the client domain. For more information, see "Enabling Requests for Deferred Queue Processing (ETRN)" on page 103.

Command Line

You can also set this value at the command line as follows:

```
configutil -o service.smtp.smtp-deliver.alwaysqueue -v [ yes | no ]
```

Verifying Recipient Addresses

You can specify that Messaging Server verify recipient addresses for messages it accepts from clients.

By enabling this option, the server can detect bad recipient names in the envelope address and return an error to the client before the client sends the body of the message. The client can fix the name before sending the message text.

Specifying this option has slight performance impact because the server must perform an LDAP lookup for each recipient while connected to the client. The benefit, however, is that bad recipients can be rejected immediately, allowing the sender to fix the address before sending (instead of getting a bounce message later).

To specify verification of recipient addresses, go to the SMTP Accept tab:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Select SMTP.
4. Click the Accept tab in the right pane.
5. Check the "Verify each recipient's address" box.
6. Click Save.

Command Line

You can also set this value at the command line as follows:

```
configutil -o service.smtp.smtp-accept.verifyrcpts -v [ yes| no ]
```

Performing Reverse IP Address Lookups

You can specify that Netscape Messaging Server perform reverse IP address lookups for client connections.

Using the client's IP address, Messaging Server will use DNS to find the associated host name. Messaging Server will subsequently refer to client machines by host name instead of IP address. For example, host names will be used in the process table, the log file, and "Received" lines in message headers.

Note: If you handle a large volume of messages, be aware that selecting this option impacts performance adversely.

To specify that the server should perform reverse IP address lookups, go to the SMTP Accept tab.

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Select SMTP.
4. Click the Accept tab in the right pane.
5. Check the "Lookup client machine names" box.
6. Click Save.

Command Line

You can also set this value at the command line as follows:

```
configutil -o service.smtp.doclientdnslookup -v [ yes| no ]
```

Specifying the Number of MTA Hops

Each MTA adds a `Received:` header to each incoming message. By counting the number of Received lines in the message header, the MTA can determine how many MTAs have already handled this message. The act of routing a message from one MTA to another is called a hop or an MTA hop. Each time an MTA handles a message, the message has taken another hop.

To deliver a message might require many hops. You might want to limit the number of hops for various reasons; for example, to prevent infinite mail loops. If the number of hops exceeds the maximum you specify, the message is bounced with an error message.

To specify the maximum number of MTA hops, go to the SMTP Accept tab:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Select SMTP.
4. Click the Accept tab in the right pane.
5. In the "Maximum number of MTA hops" field, specify a number.

   The recommended range for this parameter is 30 or more. The default number is 30.
6. Click Save.

Command Line

You can also set this value at the command line as follows:

```
configutil -o service.smtp.smtp-router.maxmtahops -v number
```

Reserving Free Disk Space for the Message Queue

You can specify a minimum amount of disk space that will remain unused for the message queue. If the minimum threshold is reached, the server will temporarily refuse to accept messages until disk space is freed. The server returns an error notifying the client of a temporary disk space shortage and asks the sending client or MTA to resend the message at a later time.

The server can also reject messages based on an administrative message size limit. For more information about specifying a maximum message size, see "Limiting Message Size (SIZE)" on page 103.

To reserve free disk space for the message queue, go to the SMTP Accept tab:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Select SMTP.
4. Click the Accept tab in the right pane.
5. In the "Minimum free disk space" field, specify a number.
6. From the drop-down list beside the field, specify Kbytes or Mbytes.
7. Click Save.

Command Line

You can also set this value at the command line as follows:

```
configutil -o service.smtp.smtp-accept.minfreediskspace -v number
```

Enabling Optional SMTP Features

Netscape Messaging Server supports several SMTP commands for enabling extra functionality in the dialog between an SMTP client (either a UA or another server) and Messaging Server.

To enable these commands, go to the SMTP Accept tab:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Select SMTP.
4. Click the Accept tab in the right pane.

From this tab, you can enable SMTP commands for the following:

- Verifying User Names (VRFY)
- Verifying a Mailing List (EXPN)
- Enabling Requests for Deferred Queue Processing (ETRN)
- Limiting Message Size (SIZE)

Verifying User Names (VRFY)

The VRFY command enables clients to send a request to your server to verify that mail for a specific user name resides on the server.

The server sends a response indicating whether the user is local or not, whether mail will be forwarded, and so on. A response of 250 indicates that the user name is local; a response of 251 indicates that the user name is not local, but the server can forward the message. The server response includes the mailbox name. The VRFY command is defined in RFC 821.

To enable verification of user names:

1. Click the SMTP Accept tab.
2. Check the “Allow SMTP command VRFY” box to enable the SMTP command for verifying a user name.
3. Click Save.

Caution: Because the server response might include user IDs, do not enable this option unless you are willing to reveal user IDs to clients accessing your server.

Command Line

You can also set this value at the command line as follows:

```
configutil -o service.smtp.smtp-accept.allowvrfy -v [ yes | no ]
```

Verifying a Mailing List (EXPN)

If both the client and the server support the SMTP EXPN command, clients can make requests to your server to verify that a particular mailing list resides on the server. The EXPN command is defined in RFC 821.

To enable verification of mailing lists on your server:

1. Click the SMTP Accept tab.
2. Check the "Allow SMTP command EXPN" box to enable the SMTP command for verifying a user name.
3. Click Save.

Caution: Do not enable this option unless you are willing to acknowledge mailing lists to clients accessing your server.

Command Line

You can also set this value at the command line as follows:

```
configutil -o service.smtp.smtp-accept.allowexpn -v [ yes | no ]
```

Enabling Requests for Deferred Queue Processing (ETRN)

If both the client (in this case another MTA) and the server support the ETRN command—when the client connects to the server to send a message, it can initiate processing of the deferred queue for the client domain. If there are any messages awaiting delivery to the domain given in the ETRN command, the server attempts to send the messages using one or more new SMTP connections. The ETRN command is defined in RFC 1985.

This feature is useful for sites that are set up as secondary mail exchange (MX) hosts for other sites that only have a dial-up connection to the Internet. By enabling this command, you permit dial-up servers to request delivery of their mail. (For more information about setting up MX hosts, see "Using MX Records" on page 275.)

To enable requests for deferred queue processing:

1. Click the SMTP Accept tab.
2. Check the "Allow SMTP command ETRN" box to enable the SMTP command for enabling requests for deferred queue processing.
3. Click Save.

Command Line

You can also set this value at the command line as follows:

```
configutil -o service.smtp.smtp-accept.allowetrn -v [ yes | no ]
```

Limiting Message Size (SIZE)

If both client and server support the SIZE command, clients can declare the size of a particular message to the server, and the server can accept or reject the message based on its size. Any attempts to send a message larger than the specified size will automatically fail and the server will return an error message indicating that the message size exceeds the maximum allowed. The SIZE command is defined in RFC 1870.

The server can also reject a message temporarily if it is running low on disk space. For more information, see "Reserving Free Disk Space for the Message Queue" on page 100.

To limit the size of messages your server accepts:

1. Click the SMTP Accept tab.
2. Check the "Allow SMTP command SIZE" box to enable the SMTP SIZE command.
3. Indicate the maximum size message the server will accept by typing a number in the field beside the checkbox; from the associated drop-down list, select MBytes or KBytes.
4. Click Save.

Command Line

You can also set these values at the command line as follows:

```
configutil -o service.smtp.smtp-accept.allowsize -v [ yes | no ]
configutil -o service.smtp.smtp-accept.maxmessagesize -v value
```

Specifying Automatic Reply Information

You can specify default reply messages for several situations. For example, you can specify a default vacation reply message for users who do not write a personalized message or you can specify a default reply for messages sent to a particular address.

To specify automatic reply information, go to the SMTP Autoreply window:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Select SMTP.
4. Click the Autoreply tab in the right pane.

5. From the drop-down list for each field, select the language of your choice.

6. Type the default messages for each of the reply fields:

   Default vacation-mode reply message. Type an automatic reply for users who do not write a personalized vacation message.

   Anyone who sends messages to a user's account while the vacation setting is activated will receive one notice about the user's absence. Any subsequent messages that person sends are ignored.

   In most cases, you should not replace a user's current delivery with the vacation setting when they set up the AutoReply handler for that user's account. If you do this, the user will return from vacation only to find that all of his or her email has been thrown away. Rather, you should use the vacation setting in addition to the normal delivery method, so mail is held for the user to retrieve upon his or her return. (Users are prevented from making this mistake because Messaging Server doesn't accept account management forms with a delivery of "Vacation" only.)

   Default echo-mode reply message. Type an automatic reply for the server's echo feature.

   The echo feature generates a message to anyone who sends a message to the account. In addition, it returns the mail (as a MIME attachment) that was sent to the account, so that the sender gets back the original message as well as the message that you entered.

   The echo feature, like the vacation feature, is intended to inform people about the status of the account they have contacted. A common use of the echo feature is to return mail addressed to people who have moved on and left no forwarding address.

   Default reply-mode reply message. Type an automatic reply for the server's default reply mode.

   The default reply feature is useful for special accounts that are created to disseminate information of one kind or another. You can create a place where people can get files, analogous to a File Transfer Protocol (FTP) site on the Internet.

7. Click Save.

Command Line

You can also set values for these attributes at the command line as follows:

To set the value for the default vacation reply:

```
configutil -o service.smtp.autoreply-handler.defaultvacation -v text
```

To set the value for the default echo reply:

```
configutil -o service.smtp.autoreply-handler.defaultecho -v text
```

To set the value for the default reply mode:

```
configutil -o service.smtp.autoreply-handler.defaultreply -v text
```

You can use the `;lang` option to specify the default language. For more information, see "configutil" on page 406.

Specifying Error Handling

There are various situations in which an MTA cannot deliver or route a message. For example, when an address refers to an unknown local account, when the maximum number of MTA hops is exceeded, or when disk quota is exceeded.

To specify error handling instructions, go to the SMTP Error tab:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Select SMTP.
4. Click the Error tab in the right pane.
5. For each error situation, choose one or more of the following error handling methods:
   - Return message to sender
   - Notify the postmaster via email
   - Log the error in the log file
6. Click Save.

Command Line

You can also set these values at the command line as follows:

To specify an error action for when an address refers to an unknown local account:

```
configutil -o service.smtp.error-handler.unknownacctsactions -v value
```

To specify an error action for when disk quota is exceeded:

```
configutil -o service.smtp.error-handler.quotaexceededactions -v value
```

To specify an error action for when the maximum number of MTA hops is exceeded:

```
configutil -o service.smtp.error-handler.hopcountexceedactions -v value
```

value is one of the following:

1 - To indicate return message to sender
4 - To indicate notify the postmaster via email
8 - To log the error in the log file

or a combination of values; for example, specifying a value of 5 indicates return a message to the sender and notify the postmaster via email.

Specifying Routing and Addressing Information

For detailed conceptual information about routing and addressing, including information about envelope rewrite methods, alternate search methods, and the SMTP routing table, see Chapter 9, "Message Routing."

To specify routing and addressing information, go to the Address tab:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Select SMTP.
4. Click the Address tab in the right pane.

From this tab, you can perform the following tasks:

- Specifying Envelope Rewrite Methods
- Specifying From Address Rewrite Style
- Specifying Alternate Search Methods
- Editing SMTP Routing Table Entries

Note: You should shut down your server before specifying configuration changes on this tab.

Specifying Envelope Rewrite Methods

You can specify whether and how the server rewrites the envelope recipient address before routing a message to a remote MTA. To specify envelope rewrite methods:

1. Click the SMTP Address tab.

2. Check one or more of the following boxes:

   Use the mailRoutingAddress attribute. This rewrite method uses the `mailRoutingAddress` attribute, which specifies a specific mail routing address.

   If you enable this rewrite method, you must modify the user's LDAP entry to include the `mailRoutingAddress` attribute. You can set this attribute only by using LDAP tools such as `ldapmodify`. For more information, see "mailRoutingAddress Attribute" on page 268.

   Combine the uid with the mailHost attribute. This rewrite method combines the `uid` attribute and the `mailHost` attribute found in the LDAP directory.

   This method is most likely to work properly if the "uid" search method is employed on the next server. For more information, see "Combine uid and mailHost Attributes" on page 269.

   Combine the local part of the address with the mailHost attribute. This rewrite method combines the local part of the original address with the `mailHost` attribute value to create the new address.

   This method is useful to support entities, such as mail groups, that do not have a uid. For more information, see "Combine Local Part and mailHost Attribute" on page 269.

3. Click Save.

The default method is to use the original address unmodified.

Command Line

You can also set this value at the command line as follows:

```
configutil -o service.smtp.smtp-router.enveloperewritemethod -v value
```

value is one of the following:

1 - Use the mailRoutingAddress attribute
2 - Combine the uid with the mailHost attribute
4 - Combine the local part of the address with the mailHost attribute

or a summary of values; for example, you can specify 7 to enable all rewrite methods.

Specifying From Address Rewrite Style

Rewriting the From: address increases the odds that replies to outgoing messages are processed correctly. For example, often the address that a mail client inserts in the From: line isn't the best choice. To specify how the server should rewrite the From: address:

1. Click the SMTP Address tab.

2. From the "From address rewrite style" drop-down list, choose one of the following rewrite styles:

   "john doe"<jdoe@company.com>. Choose this option to rewrite the address in the style indicated.

   jdoe@company.com (John Doe). Choose this option to rewrite the address in the style indicated.

   jdoe@company.com. Choose this option if you want the server to try to complete an incomplete address.

   never rewrite addresses. Choose this option if you do not want the server to rewrite any part of the From: address.

You might want to choose this option, for example, if you have a plug-in program that performs address rewrites. Or, for another example, in a multilingual environment where you trust the sender to use the appropriate alphabet and do not want to modify the address.

3. Click Save.

Note: To determine how to rewrite the address, the server looks at the specified site language for the domain. If the user information for the sender includes language information that matches the site language, the server will rewrite the address using the specified site language.

Command Line

You can also set this value at the command line as follows:

```
configutil -o service.smtp.smtp-router.smtprewritestyle
  -v [ quoted | comment | qualify | never ]
```

Specifying Alternate Search Methods

You can expand the list of possible recipient matches by specifying one or more of the following search methods. If all search methods are specified, the server tries each method in the order listed until a match is found. The default setting is search on user ID only.

For detailed information on these search methods, see "About Alternate Search Methods" on page 266.

To specify alternate search methods:

1. Click the SMTP Address tab.

2. Check one or more of the following boxes:

   Search for custom domain. Check this box if you want the server to use the "custom domain" search method.

   Search using truncated domain. Check this box if you want the server to use the "truncated domain" search method.

   Search by user ID. Check this box if you want the server to use the "user ID" search method.

Note that the "search by user ID" feature is for compatibility with earlier versions of Netscape Messaging Server. If your installation is new, Netscape recommends that you disable this feature.

3. Click Save.

Note: Specifying alternate search methods has a slight impact on performance.

Command Line

You can also set this value at the command line as follows:

```
configutil -o service.smtp.smtp-router.fallbacksearchmethod -v value
```

where *value* is one of the following:

1 - Search for custom domain
2 - Search using truncated domain
4 - Search by user ID

or a summary of values; for example, you can specify 7 to enable all search methods.

Editing SMTP Routing Table Entries

If Messaging Server assumes another messaging server is responsible for this recipient, Messaging Server checks its mail routing table to see if mail for the recipient's domain should be routed to a specific messaging server host.

Entries in the mail routing table are processed in order. You should keep this in mind when creating entries. For example, if you have an entry that sends all non-local mail to a firewall messaging server, you want this entry to be the last entry in the routing table.

For more information on the SMTP routing table and for example entries, see "Checking the SMTP Routing Table" on page 264.

To edit SMTP routing table entries:

1. Click the SMTP Address tab.

2. Click the Add button by the SMTP routing table field.

3. Type a routing table entry.
4. Click OK to return to the SMTP Address tab.
5. Click Save.

Command Line

You can also specify routing table entries at the command line as follows:

```
configutil -o service.smtp.smtp-router.hostrewrites -v entry
```

Controlling Access to SMTP Services

Netscape Messaging Server provides several features that enable you to control access to your SMTP services. These features include:

- Specifying authenticated SMTP
- Specifying access control filters
- Filtering unsolicited bulk email (UBE)

Netscape Messaging Server also supports the Secure Sockets Layer (SSL) protocol for transferring private data over TCP/IP networks. For details about determining the access control and security requirements for your server, see Chapter 6, "Security and Access Control."

Enabling Authenticated SMTP

Authenticated SMTP provides for greater security in sending messages using the SMTP protocol. To use authenticated SMTP, you do not need to deploy a certificate-based infrastructure. However, authenticated SMTP does not provide the same level of security features as a certificate-based infrastructure.

With authenticated SMTP, the client (either a user agent or another server that supports authenticated SMTP) can indicate an authentication mechanism to the server, perform an authentication protocol exchange, and optionally negotiate a security layer for subsequent protocol interactions. For example, when supported by the user's mail client, authenticated SMTP can require users to enter a password before they are allowed to send messages.

For more information about authenticated SMTP, and when and how to use it in your security and access scheme, see Chapter 6, "Security and Access Control."

To specify authenticated SMTP, go to the SMTP System tab:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Select SMTP.
4. Click the System tab in the right pane.
5. Check the "Allow password login" box.
6. Specify a minimum cipher length for password encryption.

   A cipher is the algorithm used to encrypt and decrypt data in the encryption process. A cipher operates on data by applying a key—a long number—to the data. Generally, a longer key represents a more secure encryption process.

   Caution: If you specify 0, the server permits unencrypted passwords. Do not specify 0 if you are concerned about sending passwords in clear text. Choose 40 or 128 to ensure that passwords are sent over secure channels.
7. Click Save.

Command Line

You can also set this value at the command line as follows:

```
configutil -o service.smtp.plaintextmincipher -v value
```

where *value* is one of the following:

```
 -1    - To disable authenticated SMTP
  0    - To enable with unencrypted passwords
 40    - To enable with 40-bit cipher length
128    - To enable with 128-bit cipher length
```

Specifying Access Control Filters

You can define access control filters to exclude spammers and DNS spoofers from your system and improve the general security of your network.

For detailed information about TCP client access control features including complete filter syntax, see Chapter 6, “Security and Access Control.”

Filtering Unsolicited Bulk Email

Unsolicited Bulk Email (UBE) is email sent to large number of recipients without their knowledge or consent, often advertising commercial products or services. It is the electronic equivalent of paper “junk mail.”

Netscape Messaging Server provides an SMTP UBE plugin you can use to design and implement filters that block unsolicited bulk email from reaching your servers.

For details about the UBE plugin and how to use it to filter unwanted mail, see Chapter 8, “Filtering Unsolicited Bulk Email.”

Working with SMTP Plugins

Netscape Messaging Server provides an application programming interface (API) that allows third parties to create server plugins that can add site-specific functionality to Messaging Server.

For details on working with SMTP plugins, see Chapter 7, “Working with SMTP Plug-Ins.”

Message Queue Concepts

By default, Messaging Server attempts to deliver messages immediately; the server queues mail only if there is a problem, or if you have explicitly specified deferred delivery to other servers. (For information about specifying deferred delivery, see “Deferring Delivery” on page 96.)

This section describes two types of queue: logical and physical.

Logical Queue

A logical queue is a set of messages waiting to be processed. A logical queue might be the active queue or a deferred queue.

- **Active Queue.** The active queue is a logical queue containing messages waiting to be processed for the first time. You cannot specify actions on the active queue.
- **Deferred Queue.** A deferred queue is a logical queue holding messages that have encountered a temporary failure. For each domain that has deferred messages, there is a deferred queue. There might also be a deferred queue for each type of local delivery. Thus, there are zero or more deferred queues at any one time.

  You can specify actions on deferred queues and enable requests for deferred queue processing. For more information, see "Specifying Actions on Deferred Queues" on page 116 and "Enabling Requests for Deferred Queue Processing (ETRN)" on page 103.

Physical Queue

A physical queue is a path on the server's file system that the server uses to store queued messages and their associated control information. Physical queues are like message store partitions, except they don't have names. Multiple physical queues allow the server to distribute queueing information across several places in the file system.

You can specify alternate path names for the physical queue directories, as described in "Specifying Alternate Paths for Physical Queues" on page 118. By specifying alternate path names, you can distribute the load associated with delivering a message because the server can perform concurrent I/O operations. You can also reduce the overhead associated with large numbers of files accumulating in a single message queue.

Netscape Messaging Server stores logical queues across three physical queue directories: `control`, `deferred`, `messages`.

The control Directory

The `control` directory contains the information necessary to process messages in the active queue—the queue containing messages waiting to be processed for the first time.

When the server accepts a message, it logs an entry in the `control` directory. When the server is finished processing the message (the message has been delivered to the user's inbox, the message has been deferred, or the message has been relayed), the server logs another entry in the `control` directory.

The `control` directory entries contain references to files in the `messages` directory.

The deferred Directory

The `deferred` directory has zero or more subdirectories, which contain the control information for messages that have been deferred.

The `deferred` directory entries contain references to the files in the `messages` directory.

The messages Directory

The `messages` directory contains the text (header and body) of all messages in the active and deferred queues. This directory contains one file per message.

Specifying Actions on Deferred Queues

You can specify whether to return messages to the sender, move messages to the active queue, or delete messages from the queue.

To specify actions on a deferred message queue, go to the Queued Messages window:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Open the SMTP folder and select Message Queue.

4. Click the Queued Messages tab in the right pane.
5. Select a queue from the list.
6. Click the Select Action button.
7. Select an action from the Queued Messages Action window and click OK.
8. Click Save.

Command Line

See also "mailq" on page 415 and "processq" on page 424.

Specifying Message Handling for Deferred Queues

You can specify how often deferred queues are processed and how long messages can remain in the deferred queue.

You can enable requests for processing of deferred queues to limit the number of dial-up connections to your server. With deferred queue processing, when a client (in this case another MTA) connects to the server to send a message, it can also initiate processing of the deferred queue for the client domain. For more information, see "Enabling Requests for Deferred Queue Processing (ETRN)" on page 103.

You can also perform actions on the queue from the command-line interface. For more information about the command-line utilities for managing the queue, see "mailq" on page 415 and "processq" on page 424.

To specify message handling for deferred queues:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Open the SMTP folder and select Message Queue.
4. Click the Configuration tab in the right pane.

5. To specify how often messages in the queue are processed, in the "Message queue process interval" field, type a number and, from the drop-down list, choose `Second(s)`, `Minute(s)`, or `Hour(s)`.
6. To specify how long messages can remain in the queue, in the "Maximum Message Queue Time" field, type a number, and from the drop-down list, choose `Hour(s)` or `Day(s)`.
7. Click Save.

Command Line

You can also set values at the command line as follows.

To specify how often messages in the queue are processed:

```
configutil -o service.smtp.deferredperiod -v value
```

To specify how long messages can remain in the queue:

```
configutil -o service.smtp.maxqueuetime -v value
```

Specifying Alternate Paths for Physical Queues

To specify an alternate path for queue storage, go to the Message Queue Configuration window:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and open the Services folder in the left pane.
3. Open the SMTP folder and select Message Queue.
4. Click the Configuration tab in the right pane.
5. Click the Add button beside the "MTA Queue Path" field.
6. Type a queue path and click OK.
7. Click Save.

Command Line

You can also set this value at the command line as follows.

```
configutil -o service.smtp.altqueues -v queuepath
```

Chapter

4

Managing Mail Users and Mailing Lists

This chapter explains how mail accounts and mailing lists are implemented in Netscape Messaging Server, and it describes how to use the Netscape Console interface to create and manage your users' mail accounts and mailing lists.

This chapter has the following sections:

- About Users and Groups for Messaging
- Managing Mail Users
- Managing Mailing Lists

About Users and Groups for Messaging

Netscape Messaging Server requires close integration with an LDAP directory service such as Netscape Directory Server. One reflection of this close integration is the manner in which mail accounts and mailing lists are implemented.

Users and Mail Accounts

An LDAP user directory can contain a wide range of information about an organization's employees, members, clients, or other types of individuals that in one way or another "belong" to the organization. These individuals constitute the *users* of the organization. In the LDAP directory, the information about users is structured for efficient searching, with each user entry identified by a set of attributes. Directory attributes associated with a user can include the user's name and other identification, division membership, job classification, physical location, name of manager, names of direct reports, access permission to various parts of the organization, and preferences of various kinds.

In an organization with electronic messaging services, many if not all users hold mail accounts. For Netscape Messaging Server, mail-account information is not stored locally on the server; it is part of the LDAP user directory. The information for each mail account is stored as mail attributes attached to a user's entry in the directory. To retrieve or modify information for a specific user's mail account, an administrator uses the Messaging Server interface to access that user's mail attributes in the directory.

Groups and Mailing Lists

An LDAP user directory can contain entries that represent collections of users. These directory *groups* can consist of a specific set of users or they can be rule-based, with membership defined by job classification or any other user attributes.

Groups can exist for a wide variety of purposes, and they have their own sets of attributes in the user directory. Groups may be used for information sharing in departments or on projects, for providing selective access to sensitive data, for discussion on shared interests, for disseminating company or division policy, and so on.

Netscape Messaging Server provides support for mailing lists, which can be thought of as group addresses (similar to `sendmail` aliases) with additional associated information (such as a set of access permissions for posting to the list). As with mail-account information, Messaging Server stores mailing-list information in the LDAP user directory rather than locally. The information is stored as a set of attributes belonging to a particular group. To retrieve or

modify information for a specific mailing list, an administrator uses the Netscape Console interface to access the appropriate group's attributes in the user directory.

You can create a group with mail attributes, or you can add mailing-list capability to any existing directory group.

Mail-Administration Features

You use the mail-administration portion of the Netscape Console interface to configure and administer the mail accounts and mailing lists hosted by your Messaging Server.

For any user in your user directory, you can perform the following tasks:

- Access the user's mail account
- Specify mail addressing information for the account
- Define the delivery method(s) and attributes for the account
- Specify forwarding addresses and attributes for the account
- Specify auto-reply procedures for the account

For any group in your user directory, you can perform the following tasks:

- Access the group's mailing list
- Specify mail addressing information for the mailing list
- Specify *email-only* members for the mailing list
- Define restrictions for posting messages to the mailing list
- Define and enable message-rejection actions for the mailing list

Subsequent sections in this chapter give detailed discussions of these administrative tasks. Before you can perform them, however, you must first enable the mail-administration interface, as described in the next section.

Note: For entry or manipulation of large numbers of mail accounts, it may be more efficient to use bulk methods than to use the Netscape Console interface described here. For more information, see the discussion on migration tools in Appendix A, "Command-line Utilities" and Appendix B, "sendmail Migration and Compatibility."

Managing Mail Users

Accessing Mail Users

This section describes how to open the mail administration interface for your users. Messaging Server mail accounts are stored as attributes of user entries in your enterprise's central LDAP user directory. Therefore, to manage mail accounts, you modify user entries in that directory.

Creating a New User

To create a new mail account, you create a new user in the directory. You must also install a mail account for that user; if you do not install the mail account, the mail-administration portion of Netscape Console is not available for that user. (The full process of creating a user and specifying other kinds of user information is described in more detail in Chapter 4, "User and Group Administration," of *Managing Servers with Netscape Console.*)

To create a new mail user:

1. In the Netscape Console main window, click the Users and Groups tab.
2. From the drop-down list, choose New User and click Create.
3. Select an organizational unit for the user and click OK. The Create User window opens (see Figure 4.1).
4. Enter information about the user as described in Chapter 4, "User and Group Administration," of *Managing Servers with Netscape Console*.
5. Leave the Create User window open and click the Account tab. A list of installed products for the new user's account appears in the right pane (see Figure 4.2).
6. Click the Mail Account Install box. The Mail tab becomes visible in the Create User window.
7. Click the Mail tab in the Create User window, then click the tab you want in the right pane (see Figure 4.3).

8. Enter your changes, then click OK at the bottom of the Create User window.

   Note: Make sure you complete all setup procedures in the relevant tabs before clicking OK.

Figure 4.1 Create User window (User tab selected) without Mail tab

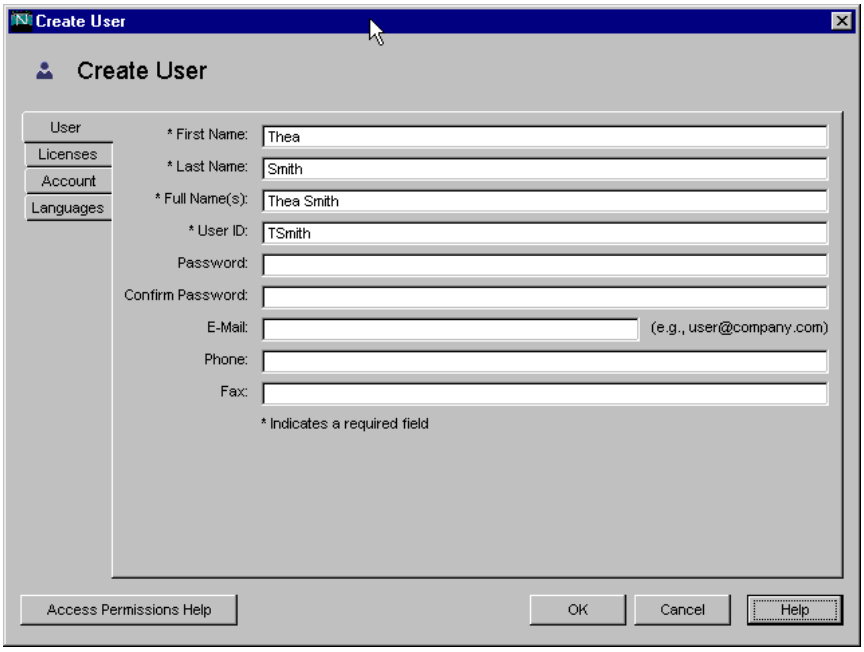


Figure 4.2 Create User window (Account tab selected) with Mail Account option

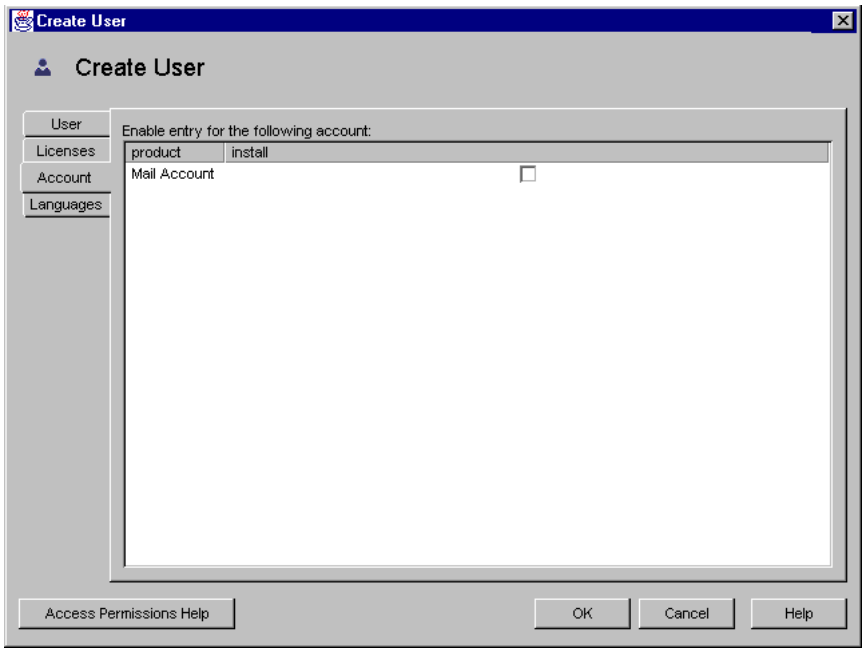
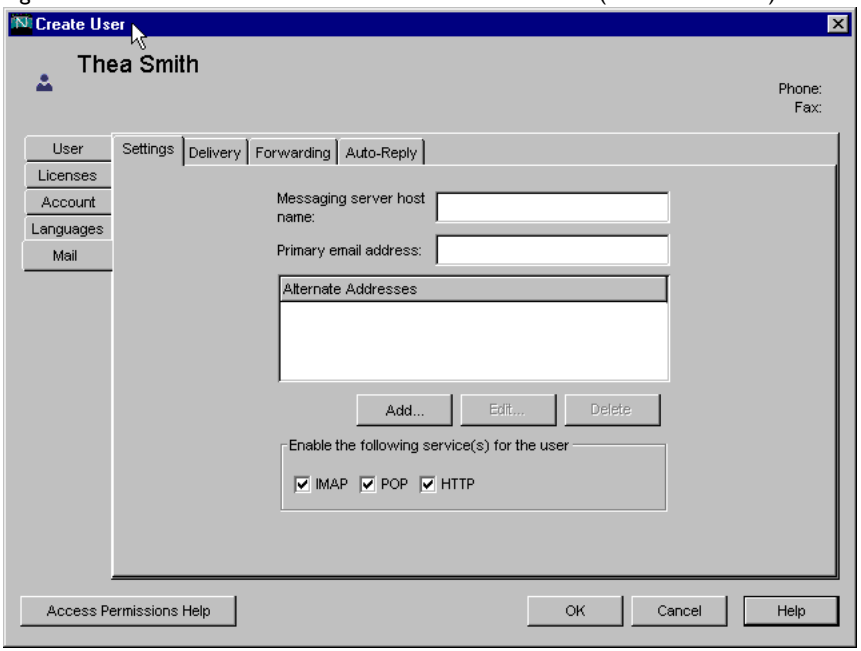


Figure 4.3 Create User window with mail account installed (Mail tab selected)



Accessing an Existing User

To modify an existing mail account or to add mail capabilities to an existing user, you access the appropriate user in the user directory and then add or modify that user's mail-account attributes.

To access mail information for an existing user:

1. In the Netscape Console main window, click the Users and Groups tab.
2. In the Users and Groups main window, Click Search or Advanced Search.
3. Enter your search criteria (such as the user's last name) in the Search window, and perform the search of the user directory.
4. Return to the Users and Groups main window, select a user from the search results and click Edit.

5. If the Mail tab is not visible in the Edit Entry window, do this:

   - Click the Account tab. A list of installed accounts appears in the right pane (see Figure 4.2).
   - Check the Mail Account box. The Mail tab displays in the Edit Entry window.

6. Click the Mail tab in the Edit Entry window, then click the tab you want in the right pane.

7. Enter your changes, then click OK at the bottom of the Edit Entry window.

Specifying User Email Addresses

Before mail can be delivered successfully to a user, you must specify the mail addressing information for that user. This consists of the Messaging Server host name, the user's primary address, and any alternate addresses. The host name and primary address information is mandatory; alternate address information is optional.

To specify a user's mail addressing information:

1. In Netscape Console, access the Create User or Edit Entry window, as described in "Accessing Mail Users" on page 124.

2. Click the Mail tab.

3. Click the Settings tab, if it is not already active.

4. (Required) Enter the Messaging Server host name.

   This is the machine hosting the Messaging Server that will process this user's mail. This must be the fully-qualified domain name (FQDN) known to the Messaging Server on that machine.

5. (Required) Enter the user's primary email address.

   This is the publicized address to which this user's mail is sent. There can be only one primary address for a user, which must be a valid, correctly formatted SMTP address conforming to RFC 821 specifications.

If you want to implement host name hiding (the host name in the user's address is not shown in the outgoing mail header), do not specify the host name in the Primary email address field. Instead, enter an alternate address that includes the host name as described in the next step.

6. (Optional) Add an address to the Alternate Address list.

An alternate address is essentially an alias for the user's primary address. You can use this feature to:

- Ensure proper delivery of frequently misspelled addresses (such as "Smith" as an alias for "Smythe").
- Enable host name hiding in outgoing mail headers. To do so, supply an alternate address that includes the host name and do not include the host name in the user's Primary email address. For example, enter `jsmith@airius.com` as a Primary email address and then enter `jsmith@airco.com` as an Alternate address. When this user sends mail, the outgoing header will show `jsmith@airius.com`, but all mail sent to that address (including replies) are actually routed to `jsmith@airco.com` (assuming that `airco.com` is a valid host name).

You can specify any number of alternate addresses for a particular user, as long as each address is unique. Messages that arrive for any of these aliases are directed to the primary address.

To add an alternate address:

- Click the Add button beneath the Alternate Addresses field.
- In the Alternate Addresses window, enter an alternate address. (You can add as many alternate addresses as you like, but you can enter only one address each time you open this window.)
- Click OK to add the alternate address and close the Alternate Addresses window. (To enter another alternate address, click Add again to re-open the Alternate Addresses window.)

7. To enable mail services for a specific type of server, check one or more of the available options: IMAP, POP, and HTTP.

8. Click OK at the bottom of the Edit Entry window if you have finished making changes to this user's mail information. Otherwise, click other tabs to continue making changes.

Configuring Delivery Options

Messaging Server supports three principal mail-delivery options that you can enable and configure, in any combination, for each user. You can provide regular POP/IMAP delivery, program delivery, and Unix delivery (for clients of a Unix Messaging Server host).

Messaging Server also provides an end-user HTML interface through which users can themselves enable and configure these options. The Netscape Console (administrator) interface and the HTML (user) interface both manipulate the same directory attributes; when opened, each shows the current settings, whether they were set by the administrator or by the user.

To configure delivery options for a user:

1. In Netscape Console, access the Create User or Edit Entry window, as described in "Accessing Mail Users" on page 124.

2. Click the Mail tab.

3. Click the Delivery tab.

4. Select the delivery method or methods you want to enable for this user:

   - To specify POP/IMAP delivery, follow the instructions in "Specifying POP/IMAP Delivery" on page 131.

   - To specify program delivery, follow the instructions in "Specifying Program Delivery" on page 131.

   - To specify Unix delivery, follow the instructions in "Specifying Unix Delivery" on page 132.

5. Click OK at the bottom of the Edit Entry window if you have finished making changes to this user's mail information. Otherwise, click other tabs to continue making changes.

Specifying POP/IMAP Delivery

Specifying this option enables mail delivery to the user's regular POP3 or IMAP4 mailboxes. To enable POP/IMAP delivery for this user:

1. Click the Delivery tab.
2. Check the POP/IMAP box, and click the Properties button to open the POP/IMAP Delivery window.
3. (Optional) Enter the nickname (not the path name or absolute physical path) of the message-store partition to which the user's messages will be delivered and stored for processing. If you leave this field blank, the current primary partition is used. For more information, see Chapter 5, "Managing the Message Store."
4. (Optional) Enter the storage limit, or disk quota, to be allotted to the user. The quota can be either unlimited (no maximum storage limit), or you can specify a limit (in KB or MB). Unlimited is the default.
5. (Optional) Specify the access domains from which the user can retrieve mail.

   Note. If no access domains are specified in the Access domain field, the user can retrieve mail from any domain.

   To specify an access domain:

   - Click Add (next to the Access Domains field) to open the Access Domain name window, then enter either a regular domain name or an IP address. You must enter only one access domain each time you open the Access Domain name window. **Note:** If you specify a domain that does not exist, or enter `none`, you effectively block access for the user.
   - Click OK to add the access domain name and dismiss the Access Domain name window. (To enter another access domain, click Add again to re-open the Access Domain name window.)

Specifying Program Delivery

Specifying this option provides a mechanism for forwarding messages to an external application for processing before delivery to the user.

Note: This section describes only how to make the program delivery option available to an individual user. Before you can make it available to a user, you must first enable the program delivery module as a whole, which requires performing several other administrative tasks. For details, see Chapter 12, "Program Delivery."

To enable program delivery for this user:

1. Click the Delivery tab.
2. Check the Program delivery box, and click the Properties button to open the Program Delivery window.
3. Enter the external application command(s) to be used for processing this user's mail.
4. Click OK.

Specifying Unix Delivery

Specifying this option selects Unix delivery for this user. The Unix delivery feature allows messages to be delivered to the user's designated Unix mailbox. Unix delivery is available only to users whose Messaging Server runs on a Unix host machine.

To enable Unix delivery for this user:

1. Click the Delivery tab.
2. Check the Unix delivery box.

Note: To provide Unix delivery to Messaging Server users, you must also perform normal Unix mail administrative tasks.

Specifying Forwarding Addresses

The mail-forwarding feature of Messaging Server enables a user's mail to be forwarded to another address instead of or in addition to the primary address for that user.

Messaging Server also provides an end-user HTML interface through which users can themselves specify forwarding addresses. The Netscape Console (administrator) interface and the HTML (user) interface both manipulate the same directory attributes; when opened, each shows the current settings, whether they were set by the administrator or by the user.

To specify forwarding-address information for a user:

1. In Netscape Console, access the Create User or Edit Entry window, as described in "Accessing Mail Users" on page 124.
2. Click the Mail tab.
3. Click the Forwarding tab.

   The Forwarding Address field shows the current set of forwarding addresses, if any, for the user.
4. To add a forwarding address, Click Add.
5. In the Forwarding Address window, enter a forwarding address.
6. Click OK to add the address to the Forwarding address field in the Mail Forwarding tab and close the Forwarding Address window.
7. Click OK at the bottom of the Edit Entry window if you have finished making changes to this user's mail information. Otherwise, click other tabs to continue making changes.

Note: Do not set up forwarding address for two users on the same Messaging Server to point to each other if both user accounts have no other delivery type enabled. Doing so can cause mail delivery problems.

Configuring Auto-Reply Settings

The auto-reply feature of Netscape Messaging Server lets you specify an automatic response to incoming mail for a user. You can specify three different auto-reply modes: echo mode, vacation mode, and auto-reply mode.

Messaging Server also provides an end-user HTML interface through which users can themselves enable and configure auto-reply settings. The Netscape Console (administrator) interface and the HTML (user) interface both manipulate the same directory attributes; when opened, each shows the current settings, whether they were set by the administrator or by the user.

To enable an auto-reply service for a user:

1. In Netscape Console, access the Create User or Edit Entry window, as described in "Accessing Mail Users" on page 124.

2. Click the Mail tab.

3. Click the Auto-Reply tab.

4. Select one of the auto-reply modes:

   Off: Disables auto-reply for this user.

   Echo: An automatic reply is sent for each received message and the received message appended as a MIME attachment to the reply. If you select this mode, you can enter a reply message in the Message field.

   Vacation: The first message received by this user from a given sender generates an automatic response; subsequent messages from that sender do not generate a response. If you select this mode, you use the Vacation start/end date options and enter a reply message in the Reply text field.

   Auto-reply: Every incoming message received by the user generates the specified automatic response. (The received message is not attached to the reply.) If you select this mode, you can enter a reply message in the Message field.

5. If you selected vacation mode, supply dates and times to determine when the auto-reply message should start and end:

   - Check the Vacation start/end date checkbox.
   - Click the Edit buttons for Start and End then use the calendar that displays to specify a date and time.

6. If you selected echo, vacation, or auto-reply mode, type a reply message to be returned to the sender.

   You can create one message in each of several available languages that you select with the drop-down list located above the message text area.

7. Click OK at the bottom of the Edit Entry window if you have finished making changes to this user's mail information. Otherwise, click other tabs to continue making changes.

Managing Mailing Lists

Accessing Mailing Lists

This section describes how to get to the administration interface for your mailing lists. Because Messaging Server mailing lists are stored as attributes of group entries in an LDAP user directory, managing mailing lists means accessing and modifying directory groups.

Creating a New Group

To create a new mailing list, you create a new group in the directory. You must also install a mail account for that group; if you do not install the mail account, the mail-administration portion of Netscape Console is not available for that group. (The full process of creating a directory group and specifying other kinds of group information is described in more detail in Chapter 4, "User and Group Administration," of *Managing Servers with Netscape Console*.)

To create a new mailing list:

1. In the Netscape Console main window, click the Users and Groups tab.
2. From the drop-down list, choose New Group and click Create.
3. Select an organizational unit for the group and click OK.
4. In the Create Group window (see Figure 4.4), enter the information required to create the group entry as described in Chapter 4, "User and Group Administration," of *Managing Servers with Netscape Console*.

Note: For mailing-list purposes only, you do *not* have to add members using the Users and Groups Members tab; you can instead add them using the Mail account Email-Only Members tab:

- Regular group members have full mailing-list privileges, but they also have any other privileges that their group membership indicates. You add regular members (either static or dynamic) through the Members tab.
- Mailing-list members have group privileges limited to those provided by the mailing-list component of the group (which may or may not be the only purpose for the group's existence). Mailing-list members are called *email-only members,* and you add them through the Mail tab.

5. Leave the Create Group window open and click the Account tab.

   A list of installed products for the group account appears in the right pane.

6. Click the Mail Account box.

   The Mail tab becomes visible in the Create Group window.

7. Click the Mail tab in the Create Group window, then click the appropriate tab in the right pane (see Figure 4.5).

8. Enter your changes, then click OK at the bottom of the Create Group window.

   This action submits your entries and dismisses the Create Group window.

   Note: Clicking OK at the bottom of any mail administration window submits all of the current mail configuration information entered in all of the mail administration tabs. Make sure you complete all setup procedures in the relevant windows before clicking OK.

Figure 4.4 Create Group window (without Mail tab)

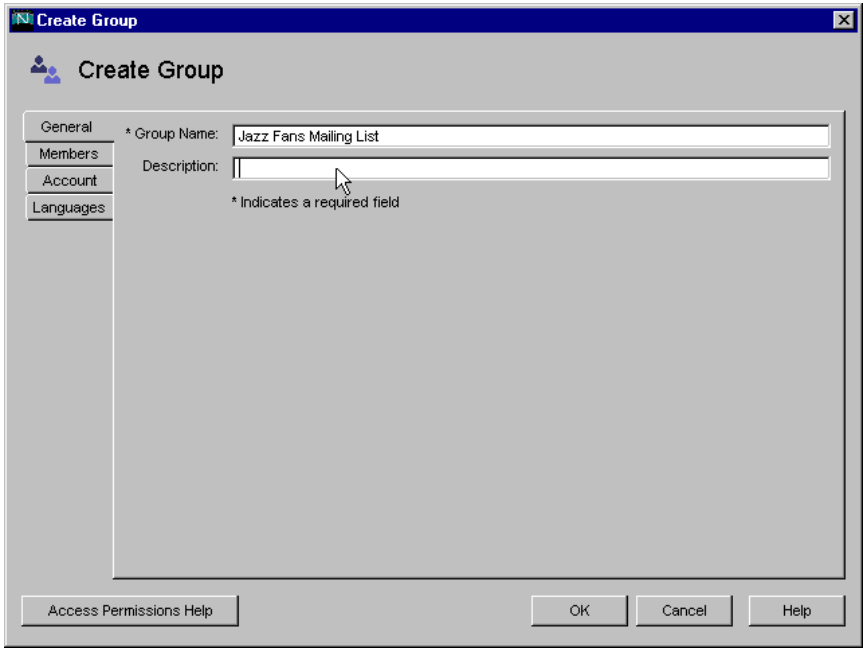
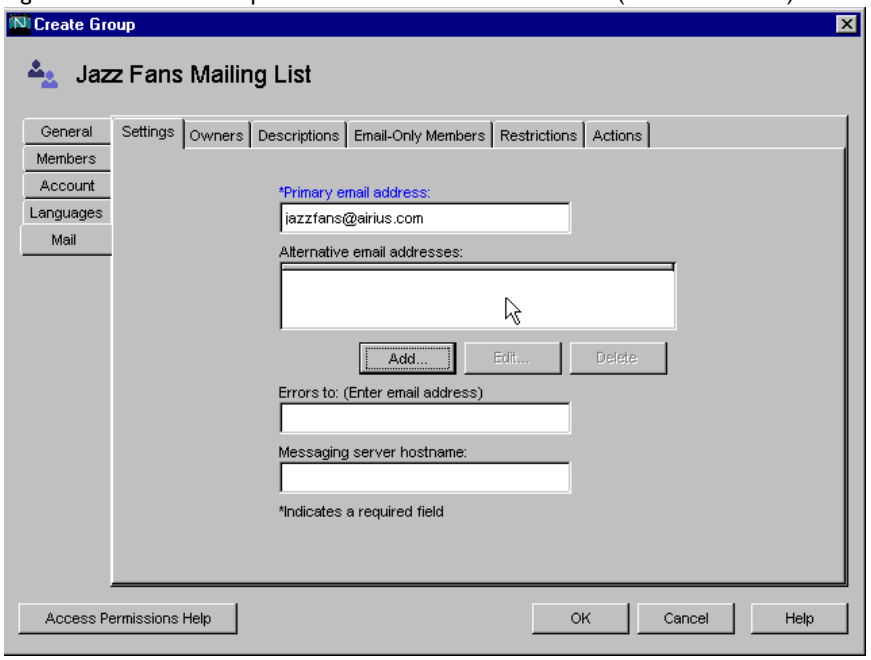


Figure 4.5 Create Group window with mail account installed (Mail tab selected)



Accessing an Existing Group

To modify an existing mailing list, or to add mailing-list capabilities to an existing group, you access the appropriate group in the user directory and then add or modify its mail-account attributes.

To access mailing-list information for an existing group:

1. In the Netscape Console main window, click the Users and Groups tab.
2. In the Users and Groups main window, Click Search or Advanced Search.
3. Enter your search criteria (such as the group's name) in the Search window, and perform the search of the user directory.
4. Return to the Users and Groups main window, select a group from the search results and click Edit.

5. If the Mail tab is not visible in the Edit Entry window, do this:

   - Click the Account tab. A list of installed accounts appears in the right pane (see Figure 4.2).
   - Check the Mail Account box. The Mail tab displays in the Edit Entry window.

6. In the Edit Entry window, click the Mail tab, then click the tab you want in the right pane.

   (These tabs are identical to those you access through the Create Group window.)

7. Enter your changes, then click OK at the bottom of the Edit Entry window to submit your modifications.

Specifying Mailing List Settings

Before mail can be delivered successfully to your mailing list, you must specify its mail-addressing information. This consists of the primary address for the group and any alternate addresses you want to accept as aliases to the primary address. You can also specify the owner(s) of the list, optional descriptive information, members, attributes, restrictions, and actions (email responses) of the mailing list.

To specify mailing-list information:

1. In Netscape Console, access the Create Group or Edit Entry window, as described in "Accessing Mailing Lists" on page 135.

2. Click the Mail tab.

3. Click the Settings tab, if it is not already the active tab.

4. (Required) Enter the mailing list's primary email address.

   This is the publicized address to which this list's mail will be delivered. There can be only one primary address for a list. It must be a correctly formatted SMTP address that conforms to RFC 821 specifications.

5. (Optional) Specify an alternate address for the mailing list.

An alternate address is an alias for the group's primary address. You can use this feature to:

- Ensure proper delivery of a frequently misspelled address.
- Enable host name hiding in outgoing mail headers. To do so, supply an alternate address that includes the host name and do not include the host name in the group's Primary email address.

You can specify any number of alternate addresses for a group, as long as each address is unique. Messages that arrive for any of these aliases are directed to the primary address.

To add an alternate email address:

- Click the Add button beneath the Alternative email addresses field.
- In the Alternative Email Addresses window, enter an alternate address. (You can add as many alternate addresses as you like, but you can enter only one address each time you open this window.)
- Click OK to add the alternate address and close the Alternative Email Addresses window. (To enter another alternate address, click Add again to re-open the Alternative Email Addresses window.)

6. (Optional) In the "Errors to" field, enter the email address of a person to whom errors in posting messages to the list should be sent.

7. (Optional) Enter the host name of the machine hosting this mailing list.

   If the "Primary email address" field for this mailing list includes a host name, you can leave this field blank. If you implement host-name hiding by having no host name in the primary email address, specify the host name in this field.

   Unlike a user mail account, if you do not specify a host name for a mailing list, any host that has access to the list's LDAP entry will be able to process the list (which, in most cases, is what you want). If you want to restrict processing of the list to one or more specific hosts, you should specify one or more host names. For example, you may want to force a large group to be processed on an under-utilized server to reduce stress on a server that is more heavily used.

   Note: This window lets you enter only one host name at a time. To enter multiple host names, use the `ldapmodify` command line utility.

8. (Optional) Enter a mailing list owner.

   A list owner has administrative privileges for adding or removing users, modifying configuration settings, or deleting the list.

   To specify a new mailing list owner, click the Owners tab and then either:

   - Click Add, then enter the distinguished name (DN) of a new mailing list owner (such as `uid=jsmith, ou=people, o=airius.com)` in the Enter List Owner's DN window and click OK.
   - Click Search to open the Search Users and Group window to locate an owner.

   Note: Selecting an owner from the Search Users and Group window automatically adds the correct syntax of the DN for you. For more details on the Search Users and Groups window, see Chapter 4, "User and Group Administration," of *Managing Servers with Netscape Console*.

9. (Optional) Add descriptive information.

   To add text or a URL for information purposes (not for use by Messaging Server), click the Descriptions tab, then use one or both of the following options:

   - Enter a description of the purpose or nature of the mailing list.
   - Enter a URL to an HTML page providing additional information about the mailing list. This is for informational purposes only; the URL is not used by Messaging Server.

10. Click OK at the bottom of the Edit Entry window if you have finished making changes to this mailing list. Otherwise, click other tabs to continue making changes.

Specifying List Members

To add email-only members to your mailing list, use one or both of the following methods:

- Explicitly add each member to the mailing list.
- Define dynamic criteria to be applied to the user directory as a filter for determining group membership.

The mailing-list members described here are called *email-only members* in the Users and Groups interface of Netscape Console because they have group privileges limited to those provided by the mailing-list component of the group. "Regular" group members, which you add using a different part of the interface (described in *Managing Servers with Netscape Console*), might have additional privileges or responsibilities beyond those of mailing-list members. For more information on groups, see Chapter 4, "User and Group Administration," of *Managing Servers with Netscape Console*.

Defining Dynamic Membership Criteria

Dynamic criteria consist of LDAP search URLs that are used as filters in searching the user directory for determining membership. This mechanism is dynamic in that, when a message arrives for the group, the individuals that receive it are determined by a directory search rather than by consulting a static list of names. You can thus create and maintain very large or complex groups without having to track each member explicitly.

LDAP search filters must be formatted in LDAP URL syntax. For more detailed information on constructing LDAP filters, see Chapter 4, "User and Group Administration," of *Managing Servers with Netscape Console*. See also the Netscape Directory Server documentation and RFC 1959.

An LDAP URL has the following syntax:

```
ldap://hostname:port/base_dn?attributes?scope?filter
```

where the options of the URL have the following meanings:

option	Description
hostname	Host name of the Directory Server (Defaults to the Directory server host name used by Messaging Server).
port	Port number for the LDAP server. If no port is specified, it defaults to the standard LDAP port used by Messaging Server.
base_dn	Distinguished name of an entry in the directory, to be used as the search base. This component is required.
attributes	The attributes to be returned. These attributes are supplied by Messaging Server.

option	Description
scope	Scope of search: • A scope of `base` retrieves information only on the search base (*base_dn*) itself. • A scope of `one` retrieves information one level below the search base (the search-base level is not included). • A scope of `sub` retrieves information on the search base and all entries below the search base.
filter	Search filter to apply to entries within the specified scope of the search. If no filter is specified, (`objectclass=*`) is used.

The following is an example of an LDAP search URL that filters for users who have Sunnyvale as their mail host:

```
ldap:///o=Airius Corp,c=US??sub?(&(mailHost=sunnyvale.ace.com)
  (objectClass=inetOrgPerson))
```

The above URL filters for users who are members of the organization of Airius (`o=Airius`), in the United States (`c=US`), and have a mail host of Sunnyvale (`mailHost=sunnyvale`). The objectClass attribute defines the type of entry for which to search, in this case inetOrgPerson (`objectClass=inetOrgPerson`).

Note that when you create a search filter using Netscape Console, all group names are ignored; that is, only user names are included in the search results whereas group members are not. The purpose of this setting is to avoid duplicating users that are also group members in the search results. This setting can be overridden using the command line configuration utility (`configutil`), but it is not recommended.

As noted in the next section, Netscape Console provides a template window (the Construct LDAP Search URL window) that you can use as an aid in building a search URL.

Adding Mailing-List Members

To add (email-only) members to a mailing list:

1. In Netscape Console, access the Create Group or Edit Entry window, as described in "Accessing Mailing Lists" on page 135.

2. Click the Mail tab.

3. Click the Email-only Members tab.

4. (Optional) To specify an LDAP Search URL for determining membership, click the Add button beneath the "Dynamic criteria for email-only membership" field, then in the Add Dynamic Criterion window:

   - Enter an LDAP Search URL in the field or click the Construct button to open the Construct LDAP Search URL window, a template that aids construction of the search URL.

   - Click OK to add your entry to the "Dynamic criteria for email-only membership" field and dismiss the Add Dynamic Criterion window.

   For instructions on creating an LDAP Search URL, see "Defining Dynamic Membership Criteria" on page 142.

5. (Optional) To add an individual member to the mailing list, click the Add button beneath the "Members with email only membership" field, then in the Add Email-Only Member window:

   - Enter the primary address for the new member in the field. The primary address must be a correctly-formatted SMTP address that conforms to RFC 821 specifications. You should not enter an alternate address—especially if you specify restrictions for the group. You can add only one new member each time you open this window; the field cannot hold more than one address.

   - Click OK to add the user to the members list and dismiss the Add Email-Only Member window. To enter another address, click Add again to re-open the Add Email-Only Member window.

6. Click OK at the bottom of the Edit Entry window if you have finished making changes to this mailing list. Otherwise, click other tabs to continue making changes.

Defining Message-Posting Restrictions

You can impose various kinds of restrictions on messages sent to a mailing list. You can define the set of people allowed to post messages, you can require authentication of senders, you can restrict where posted messages can come from, and you can limit the size of a posted message. Messages that violate the restrictions are rejected.

Note: Although these restrictions are useful for controlling several aspects of the incoming messages for a group, they are not intended to provide high-security access control.

To define message-posting restrictions for a group:

1. In Netscape Console, access the Create Group or Edit Entry window, as described in "Accessing Mailing Lists" on page 135.

2. Click the Mail tab.

3. Click the Restrictions tab.

4. (Optional) Define the allowed senders by choosing one of the following options:

   - **Anyone:** No restrictions on senders. (This is the default.) Note that if you choose this option, you cannot select SMTP authentication described in the next step.

   - **Anyone in the mailing list:** Only mailing-list members (including group members that are not email-only members) can post messages.

   - **Anyone in the following list:** Only those users explicitly listed in the following field can post messages.

   If you choose "Anyone in the following list", click Add below the Allowed Senders field to add a sender. The Add Allowed Sender window opens. Enter the email address or distinguished name (DN) of the allowed sender into the field. You can enter the address or DN directly, or you can click Search to open the Search Users and Groups window. Click OK to add the sender to the Allowed Senders field and dismiss the Add Allowed Sender window. Repeat this step for all other allowed senders you want to add.

   For a description of the Search Users and Groups window, see *Managing Servers with Netscape Console*.

5. (Optional) Define the sender authentication policy. You can accept the default (senders do not have to be authenticated), or make one of these choices:

   - **Only allow senders with SMTP authentication:** If this box is checked, only senders that have authenticated to their SMTP server can post messages. For information on authenticated SMTP, see "SMTP Password Login" on page 173. (This option is not available if you chose **Anyone** in step 4.)

   - **Only allow messages with the following password:** If this box is checked, only messages that include the proper password are accepted.

   If you choose "Only allow messages with the following password," you can set up (or change) the password for message authentication by entering it (twice) into the fields below the button. Then, only messages that include that password will be accepted.

6. (Optional) Define the allowed sender domains to restrict where senders can post messages from:

   - Click the Add button beneath the Allowed sender domains field.

   - In the Add Allowed Sender Domain window, enter a domain name, then click OK to add the domain to the list.

   Note: A domain automatically includes any of its subdomains. For example, `airius.com` includes `sales.airius.com`.

7. (Optional) Define the maximum permitted message size.

   Enter the size (in bytes).

8. Click OK at the bottom of the Edit Entry window if you have finished making changes to this mailing list. Otherwise, click other tabs to continue making changes.

Defining Message-Rejection Actions

You can configure Messaging Server to execute certain notification actions automatically when messages to your mailing list are rejected because they violate the list's message-posting restrictions.

This feature lets you define the action to be executed upon rejection of a mail message, and to specify group moderators. The actions that the server can take include notification to a moderator and reply to the sender (with or without appending the original message).

To define message-rejection actions for a mailing list:

1. In Netscape Console, access the Create Group or Edit Entry window, as described in "Accessing Mailing Lists" on page 135.

2. Click the Mail tab.

3. Click the Actions tab.

4. (Optional) To automatically forward rejected messages to a moderator, check the "Send message to the moderator(s)" box.

   When a moderator receives the forwarded message, that person decides how to process the message. (In the case of multiple moderators, processing of the message is determined by the action taken by the first moderator.) Processing might include approving the message and forwarding it back to the list (perhaps with a password) or deleting it. By checking this box you can thus institute a fully-moderated mailing list.

   To specify a moderator:

   - Click the Add button beneath the List moderators field.

   - In the Add Moderator window, enter a moderator's primary email address or distinguished name (DN) in the field. You can enter the address explicitly or you can click Search to use the Search Users and Groups window to locate an address. Note that you can add only one moderator each time you open the Add Moderator window.

   - Click OK to add the moderator to the List Moderators list and dismiss the Add Moderator window. (To enter another address, click Add again to re-open the Add Moderator window.)

   For a description of the Search Users and Groups window, see *Managing Servers with Netscape Console*.

5. (Optional) To automatically reply to rejected messages, check the "Send the following reply to the sender" box and use one or both of the following options:

   - Enter the text of the reply message in the Reply text field. If this field is empty, Messaging Server uses a brief default message in the reply.
   - If you want the original message to be returned to the sender (as a MIME attachment) along with the reply message, check the "Include original message with reply" box.

   Note: If neither the "Send the following reply to the sender" box nor the "Send message to the moderator(s)" box is checked, Messaging Server acts as if the "Send the following reply to the sender" box were checked.

6. Click OK at the bottom of the Edit Entry window if you have finished making changes to this mailing list. Otherwise, click other tabs to continue making changes.

Chapter

5

Managing the Message Store

This chapter describes the message store and the message store administration interface. This chapter contains the following sections:

- Overview
- Message Store Directory Layout
- How Messages Are Erased from the Store
- Specifying Administrator Access to the Store
- Configuring User Disk Quotas
- Configuring Message Store Partitions
- Specifying Aging Policies
- Performing Maintenance and Recovery Procedures

Overview

The message store contains the user mailboxes for a particular Messaging Server instance. The size of the message store increases as the number of mailboxes, folders, and log files increase. You can control the size of the store by specifying limits on the size of mailboxes (disk quotas) and by setting aging policies for messages in the store.

As you add more users to your system, your disk storage requirements increase. Depending on the number of users your server supports, the message store might require one physical disk or multiple physical disks. If you have a very large user base, you might have multiple Messaging Server instances, each responsible for a particular message store.

To manage the message store, Netscape Messaging Server provides a set of command-line utilities in addition to the Netscape Console interface. Table 5.1 describes these command-line utilities. For information about using these utilities, see "Performing Maintenance and Recovery Procedures" on page 165 and Appendix A, "Command-line Utilities."

Table 5.1 Message store command-line utilities

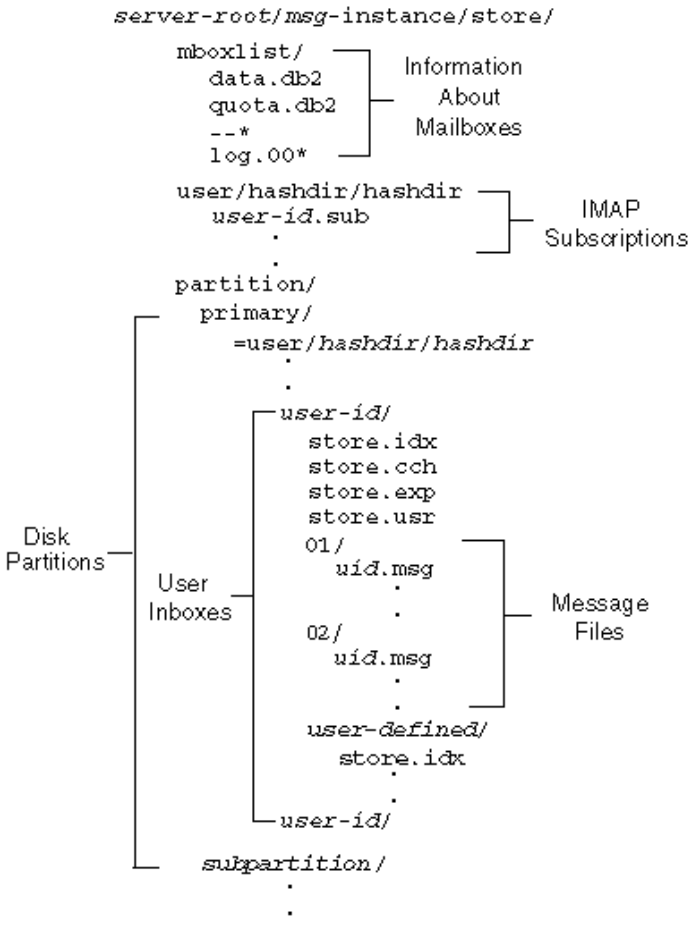
Utility	Description
`hashdir`	Identifies the directory that contains the message store for a particular user.
`mboxutil`	Lists, creates, deletes, renames, or moves mailboxes.
`quota`	Reports quota usage.
`readership`	Collects readership information on mailboxes.
`reconstruct`	Reconstructs mailboxes that have been damaged or corrupted.
`stored`	Performs background and daily tasks, expunges, and erases messages stored on disk.

Netscape Messaging Server also provides utilities to help you upgrade from a 3.x server to a 4.x server. For details about the upgrade process, see the *Messaging Server Installation Guide*.

Message Store Directory Layout

Figure 5.1 shows the message store directory layout for a server instance. The message store is designed to provide fast access to mailbox contents. The store directories are described in Table 5.2.

Figure 5.1 Message store directories



For example, a sample directory path might be:

server-root/msg-*instance*/store/partition/primary/=user/53/53/=mack1

Table 5.2 Message store directories

Location	Content/Description
server-root/msg-*instance*/store/	Top-level directory of the message store. Contains the mboxlist, user, and partition subdirectories.
.../store/mboxlist/	Contains a database (Berkley DB) that stores information about the mailboxes on the server and stores quota information about the mailboxes. The file data.db2 contains information about mailboxes, including the name of the partition where the mailbox is stored, the ACL, and a copy of some of the information in store.idx. There is one entry in data.db2 per mailbox The file quota.db2 contains information about quotas and quota usage. There is one entry in quota.db2 per user.
.../store/user/	Contains information about the IMAP folders to which each user subscribes. Information for each user is stored in a file called *user-id*.sub. These files are stored in a hash structure for fast searching. To find the directory that contains a particular user's files, use the hashdir utility.
.../store/partition/	Contains the default primary partition. You can also place any other subpartitions you define in this partition.
/*subpartition*/=user/	Contains all the user mailboxes in the subdirectory of the partition. The mailboxes are stored in a hash structure for fast searching. To find the directory that contains a particular user's mailbox, use the hashdir utility.
/=user/*hashdir*/*hashdir*/*user-id*/	The top-level mail folder for the user whose ID is *user-id*. Messages are delivered to this mail folder.
/*user-id*/*folder-name*	A user-defined folder.

Table 5.2 Message store directories (Continued)

Location	Content/Description
*/user-id/*store.idx	An index that provides the following information about mail stored in the */user-id/* directory: number of messages, disk quota used by this mailbox, the time the mailbox was last appended, pointers within the store.cch file for each message, message flags, and the size of each message. The index also includes a backup copy of mboxlist information for each user and a backup copy of quota information for each user.
*/user-id/*store.cch	A cache file for frequently requested message information. The file contains variable-length information for each message including the headers and the MIME structure. **Note:** The authenticated sender information in this file can be lost by running the reconstruct utility.
*/user-id/*store.exp	Contains a list of message files that have been expunged, but not removed from disk.
*/user-id/*store.usr	Contains a list of users who have accessed the folder. For each user listed, contains information about the last time the user accessed the folder, the list of messages the user has seen, and the list of messages the user has deleted.
/user-id/nn/	A hash directory that contains messages in the format *msg-id*.msg; *nn* can be a number from 00 to 99. For example, messages 1 through 99 are stored in the 00 directory; messages 100 through 199 are stored in the 01 directory; messages 9990 through 9999 are stored in the 99 directory; messages 10000 through 10099 are in the 00 directory, and so on.

How Messages Are Erased from the Store

Messages are erased from the store in three stages:

1. **Delete**. A client marks the message to be deleted. At this point, the client can restore the message by removing the "deleted" marking.

2. **Expunge**. A client, or the aging policies you have specified, expunges messages that have been marked deleted from the mailbox. Once messages are expunged, the client can no longer restore them, but they are still stored on disk. (A second client with an existing connection to the same mailbox may still be able to fetch the messages.)

3. **Cleanup**. The `stored` utility erases from the disk any messages that have been expunged for at least one hour.

Specifying Administrator Access to the Store

Message store administrators can view and monitor user mailboxes and specify access control for the message store. Store administrators have proxy authentication privileges to any service (POP, IMAP, HTTP, or SMTP), which means they can authenticate to any service using the privileges of any user. These privileges allow store administrators to run certain utilities for managing the store. For example, using `MoveUser`, store administrators can move user accounts and mailboxes from one system to another (for more information, see "MoveUser" on page 419).

To specify administrator access to the store:

1. From Netscape Console, open the Messaging Server you want to configure.

2. Click the Configuration tab and select Message Store in the left pane.

3. Click the Administrator tab in the right pane.

From this tab, you can perform the following tasks:

- Adding an Administrator
- Modifying an Administrator Entry
- Deleting an Administrator Entry

Adding an Administrator

To add an administrator entry:

1. Click the Administrator tab.

   The tab contains a list of existing administrator IDs.

2. Click the Add button beside the Administrator UID window.

3. In the Administrator UID field, type the user ID of the administrator you want to add.

   The user ID you type must be known to the Netscape Directory Server.

4. Click OK to add the administrator ID to the list displayed in the Administrator tab.

5. Click Save in the Administrator tab to save the newly modified Administrator list.

Command Line

You can also add store administrators at the command line as follows:

```
configutil -o store.admins -v "adminlist"
```

where *adminlist* is a comma-separated list of administrator IDs. If you specify more than one administrator, you must enclose the list in quotes.

Modifying an Administrator Entry

To modify an existing entry in the message store Administrator UID list:

1. Click the Administrator tab.

2. Click the Edit button beside the Administrator UID window.

3. Enter your changes to the Administrator UID field.

4. Click OK to submit your changes and dismiss the Edit Administrator window.

5. Click Save in the Administrator tab to submit and preserve the modified Administrator list.

Command Line

You can also modify the store administrator list at the command line as follows:

```
configutil -o store.admins -v "adminlist"
```

Deleting an Administrator Entry

To delete an entry from the message store Administrator UID list:

1. Click the Administrator tab.
2. Select an item in the Administrator UID list.
3. Click Delete to delete the item.
4. Click Save to submit and preserve your changes to the Administrator list.

Command Line

You can also delete store administrators at the command line by editing the store administrator list as follows:

```
configutil -o store.admins -v "adminlist"
```

Configuring User Disk Quotas

You can specify disk quotas to control disk space usage. Disk quotas allow you to limit the amount of disk space allotted to each user. Quotas apply to the total size of all the user's messages, regardless of how many mail folders the user has. If disk space is limited, you might want to set user disk quotas.

If the total size of all the user's messages exceeds the specified limit, messages destined for the user remain in the message queue until one of the following occurs: (1) The size of all the user's messages no longer exceeds the limit, at

which time the server delivers the message to the user. (2) The message has been in the queue longer than the specified grace period and the user is still over quota, at which time the server bounces the message.

Note: The server does not consider the size of the message when it is attempting to deliver to an account that is still under quota. If the message causes the user to go over quota, the message is still delivered, but the next message will be held in the queue.

Disk space becomes available when a user deletes and expunges messages or when the server deletes messages according to the maintenance policies you have established.

You can set disk quotas for individual users by using the Users and Groups interface. You can set default disk quotas for all users by choosing the Message Store Quota tab.

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and select Message Store in the left pane.
3. Click the Quota tab in the right pane.

From this form, you can perform the following tasks:

- Specifying a Default User Disk Quota
- Specifying a Quota Threshold
- Defining a Quota Warning Message
- Setting a Grace Period

Specifying a Default User Disk Quota

The default disk quota applies to users who do not already have individual disk quotas set for them. A quota set for an individual user overrides the default quota.

1. Click the Quota tab.
2. Select one of the following options:
    - **Unlimited**. Select this option if you do not want to set a default disk quota.

- **Size specification**. Select this option if you want to restrict the default user disk quota to a specific size. In the field beside the button, type a number, and from the drop-down list, choose Mbytes or Kbytes.

3. Click Save.

Command Line

You can also specify a default user disk quota at the command line as follows:

```
configutil -o store.defaultmailboxquota -v [ -1 | number ]
```

where `-1` indicates no quota; *`number`* indicates a number in bytes.

Specifying a Quota Threshold

You can send a warning message to IMAP users before they reach their disk quota by specifying a quota threshold. When a user's disk usage exceeds the specified threshold, the server sends a warning message to the user.

For IMAP users whose clients support the IMAP ALERT mechanism, the message is displayed on the user's screen each time the user selects a mailbox (a message is also written to the IMAP log).

To specify a quota threshold:

1. Click the Quota tab.
2. In the "Quota warning threshold" field, enter a number for the warning threshold.

   This number represents a percentage of the allowed quota. For example, if you specify 90%, the user is warned after using 90% of the allowed disk quota. The default is 90%. To turn off this feature, enter 100%.

3. Click Save.

Command Line

You can also specify a quota threshold at the command line as follows:

```
configutil -o store.quotawarn -v number
```

where *number* indicates a percentage.

Defining a Quota Warning Message

You can define the message that will be sent to users who have exceeded their disk quota as follows. Messages are sent to the user's mailbox.

1. Click the Quota tab.
2. From the drop-down list, choose the language you want to use.
3. Type the message you want to send in the message text field below the Threshold field.
4. Click Save.

Command Line

You can also define a quota warning message at the command line as follows:

```
configutil -o store.quotaexceededmsg -v message
```

The message must be in RFC 822 format.

Setting a Grace Period

If a user mailbox exceeds the disk quota, the grace period you specify determines how long messages will be held in the message queue before the server starts bouncing the messages. Messages will remain in the queue until one of the following occurs:

- The mailbox no longer exceeds the quota, at which time the server will deliver the message to the mailbox.

- The user has remained over quota longer than the specified grace period, at which time the server will bounce the message.
- The message has remained in the queue longer than the maximum message queue time. (For information on setting the maximum message queue time, see "Specifying Message Handling for Deferred Queues" on page 117.)

To set a grace period for how long messages are held in the queue:

1. Click the Quota tab.
2. In the "Over quota grace period" field, enter a number.
3. From the drop-down list, specify `Day(s)` or `Hour(s)`.
4. Click Save.

Command Line

You can also specify a quota grace period at the command line as follows:

```
configutil -o store.quotagraceperiod -v number
```

where *`number`* indicates number of hours.

Configuring Message Store Partitions

All user mailboxes are stored by default in the `/store/partition/`directory. The `partition` directory is a logical directory that might contain a single subpartition or multiple subpartitions. The subpartitions might map to a single physical drive or to multiple physical drives. At start-up time, the `partition` directory contains one subpartition called the `primary` partition.

You can add partitions to the `partition` directory as necessary. For example, you might want to partition a single disk to organize your users as follows:

```
/partition/mkting/
/partition/eng/
/partition/sales/
```

As disk storage requirements increase, you might want to map these partitions to different physical disk drives.

You should limit the number of mailboxes on any one disk. Distributing mailboxes across disks improves message delivery time (although it does not necessarily change the SMTP accept rate). The number of mailboxes you allocate per disk depends on the disk capacity and the amount of disk space allocated to each user. For example, you can allocate more mailboxes per disk if you allocate less disk space per user.

If your message store requires multiple disks, you can use RAID (Redundant Array of Inexpensive Disks) technology to ease management of multiple disks. With RAID technology, you can spread data across a series of disks but the disks appear as one logical volume so disk management is simplified. You might also want to use RAID technology for redundancy purposes; that is, to duplicate the store for failure recovery purposes.

To add a partition to the store, do the following:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and select Message Store in the left pane.
3. Click the Partition tab in the right pane.
4. Click the Add button.
5. Enter the Partition nickname.

   The name you enter must be an alphanumeric name and must use lowercase letters.

   The partition nickname allows you to map users to a logical partition name regardless of the physical path. When setting up user accounts and specifying the message store for a user, you can use the partition nickname.
6. Enter the Partition path.

   This is the absolute path name for the specified partition. The partition will be created at this location. The user ID used to run the server must have permission to write to this location, in order to create and manage the partition.
7. To specify this as the default partition, click the selection box labeled Make This the Default Partition.
8. Click OK to submit this partition configuration entry and dismiss the window.

9. Click Save to submit and preserve the current Partition list.

Note: After adding a partition, you must stop then restart the server to refresh the configuration information.

Note: To improve disk access, the message store and the message queue should reside on separate disks.

Command Line

You can also add a partition to the store at the command line as follows:

```
configutil -o store.partition.nickname.path -v path
```

where *nickname* is the logical name of the partition and *path* indicates the absolute path name where the partition is stored.

Specifying Aging Policies

Aging policies are another way to control disk usage on your server. You can control how long messages are stored in one or more mailboxes. If you have limited disk space, you might want to set aging policies to remove messages from the store. If you set aging policies, you should educate your users about these policies because the server will not send warning messages before it deletes messages from the store.

You can create aging rules based on the following criteria:

- Number of messages in the mailbox
- Total size of the mailbox
- Number of days that messages remain in the mailbox
- Number of days that messages exceeding a given size remain in the mailbox

If you specify more than one rule for a mailbox, all expiration rules will apply, but the most restrictive rule takes precedence. For example, assume two rules apply to a single mailbox. The first rule allows 1000 messages; the second rule allows 500 messages. When expiration occurs, the server will delete messages from the mailbox until 500 remain. For another example, if the first rule allows a message size of 100,000 bytes for 3 days and the second rule allows a message size of 1000 bytes for 12 days, the resulting union of rules allows a

message size of 1000 bytes for 3 days. The server will delete messages over 1000 bytes that have been in the mailbox over 3 days. If you want to ensure that a specific rule is the only rule for a particular mailbox or set of mailboxes, use the Exclusive parameter.

To create a new rule:

1. From Netscape Console, open the Messaging Server you want to configure.
2. Click the Configuration tab and select Message Store in the left pane.
3. Click the Aging tab in the right pane.
4. Click Add to go to the Add Rule window.
5. Enter a name for the new rule.
6. Specify the target folders for which this rule applies.

   You can enter a path name, filename, or partial string. You can use IMAP wildcards as follows:

   * - Match any character.
   % - Match any character except a slash character.

   The new rule applies only to folders matching the pattern you specify.
7. If this rule is to be the only rule applied to the target folders, click the Exclusive selection box.
8. If you want to create a rule based on folder size, do the following:

   - In the "Message count" field, specify the maximum number of messages that will be retained in a folder before the oldest messages are removed.
   - In the "Folder size" field, specify a number for the folder size; from the associated drop-down list, choose Mbyte(s) or KByte(s).

   When the specified folder size is exceeded, the server removes the oldest messages until this size is no longer exceeded.
9. If you want to create a rule based on message age, in the "Number of days" field, specify a number to indicate how long messages should remain in the folder.

10. If you want to create a rule based on message size:

    - In the "Message size limit" field, enter a number to indicate the maximum size message allowed in the folder; from the associated drop-down list, choose Mbytes or Kbytes.
    - In the "Grace period" field, enter a number to indicate how long oversized messages should remain in the folder.

    After the grace period, the server deletes messages that exceed the maximum size.

11. Click OK to add the new rule to the Aging Rule list and dismiss the Add window.

12. Click Save to submit and preserve the current Aging Rule list.

Command Line

You can also specify aging rules at the command line as follows. In the examples, *name* represents the name you give the rule.

To specify the target folders for which this rule applies:

```
configutil -o store.expirerule.name.folderpattern -v pattern
```

To specify that this rule is to be the only rule applied to the target folders:

```
configutil -o store.expirerule.name.exclusive -v [ yes | no ]
```

To specify the maximum number of messages that will be retained in a folder before the oldest messages are removed:

```
configutil -o store.expirerule.name.messagecount -v number
```

To specify the folder size:

```
configutil -o store.expirerule.name.foldersizebytes -v number
```

where *number* is a size in bytes.

To specify message age:

```
configutil -o store.expirerule.name.messagedays -v number
```

where *number* indicates the number of days.

To specify message size:

```
configutil -o store.expirerule.name.messagesize -v number
```

where *number* is a size in bytes.

To indicate how long over-sized messages should remain in the folder:

```
configutil -o store.expirerule.name.messagesizedays -v number
```

where *number* indicates number of days.

Performing Maintenance and Recovery Procedures

This section provides information about the utilities you use to perform maintenance and recovery tasks for the message store. You should always read your postmaster mail for warnings and alerts that the server might send. You should also monitor the log files for information about how the server is performing.

You can configure the server error handler so that error messages about disk quota are sent to the postmaster account and to the log file. For more information about configuring the error handler, see "Specifying Error Handling" on page 106. For more information about log files, see Chapter 11, "Logging and Log Analysis."

Using the stored Utility

The `stored` utility performs the following monitoring and maintenance tasks for the server:

- Background and daily messaging tasks
- Deadlock detection and rollback of deadlocked database transactions
- Cleanup
- Implementation of aging policies
- Monitoring server state and issuing alarms as necessary

This utility automatically performs cleanup and expiration operations once a day.

For more information about using the `stored` utility, see Appendix A, "Command-line Utilities."

Managing Mailboxes

The mailboxes in the message store are stored in a hash structure for fast searching. Consequently, to find the directory that contains a particular user's mailbox, use the `hashdir` utility as follows:

```
hashdir userid
```

You use the `mboxutil` command to perform typical maintenance tasks on mailboxes. These tasks include the following:

- List mailboxes
- Create mailboxes
- Delete mailboxes
- Rename mailboxes
- Move mailboxes

For more information about using the `mboxutil` utility, see Appendix A, "Command-line Utilities."

Repairing Mailboxes and the Mailboxes Database

If one or more mailboxes becomes corrupt, you can use the `reconstruct` utility to rebuild the mailboxes or the mailboxes database, and repair any inconsistencies. For more information about using the `reconstruct` command, see Appendix A, "Command-line Utilities."

Monitoring Disk Space

You can monitor disk space by configuring the following alarm attributes. You configure these attributes by using the `configutil` utility. You can specify how often the system should monitor disk space and under what circumstances the system should send a warning.

```
alarm.diskavailmsgalarmstatinterval
alarm.diskavail.msgalarmthreshold
alarm.diskavail.msgalarmwarninginterval
```

For example, if you want the system to monitor disk space every 600 seconds, specify the following command:

```
configutil -o alarm.diskavail.msgalarmstatinterval -v 600
```

If you want to receive a warning whenever available disk space falls below 20%, specify the following command:

```
configutil -o alarm.diskavail.msgalarmthreshold -v 20
```

For more information about setting alarm attributes, see Appendix A, "Command-line Utilities."

Monitoring Disk Quota Usage

You can monitor disk quota usage by using the `quota` utility. The `quota` utility generates a report that lists defined quotas and limits, and provides information on quota usage. For more information on the `quota` utility, see Appendix A, "Command-line Utilities."

Backing Up and Restoring the Message Store

For information about backing up the message store and restoring the message store, contact your Netscape technical support person.

Chapter

6

Security and Access Control

Netscape Messaging Server supports a full range of flexible security features that allow you to keep messages from being intercepted, prevent intruders from impersonating your users or administrators, and permit only specific people access to specific parts of your messaging system.

The Messaging Server security architecture is part of the security architecture of Netscape servers as a whole. It is built on industry standards and public protocols for maximum interoperability and consistency. To implement Messaging Server security policies, therefore, you will need not only this chapter but several other documents as well. In particular, information in *Managing Servers with Netscape Console* is required for setting up Messaging Server security.

This chapter has the following sections:

- About Server Security
- About HTTP Security
- User Password Login
- Configuring SSL Encryption and Authentication
- Configuring Administrator Access to Messaging Server
- Configuring Client Access to TCP Services

About Server Security

Server security encompasses a broad set of topics. In most enterprises, ensuring that only authorized people have access to the servers, that passwords or identities are not compromised, that people do not misrepresent themselves as others when communicating, and that communications can be held confidential when necessary are all important requirements for a messaging system.

Perhaps because the security of server communication can be compromised in many ways, there are many approaches to enhancing it. This chapter focuses on setting up encryption, authentication, and access control. It discusses the following security-related Messaging Server topics:

- **User ID and password login:** requiring users to enter their user IDs and passwords to log in to IMAP, POP, HTTP, or SMTP, and the use of SMTP password login to transmit sender authentication to message recipients.
- **SSL encryption and authentication:** setting up your server to use the SSL protocol to encrypt communication and authenticate clients.
- **Administrator access control:** using the access-control facilities of Netscape Console to delegate access to a Messaging Server and its individual tasks.
- **TCP client access control:** using filtering techniques to control which clients can connect to your server's POP, IMAP, HTTP, and SMTP services.

Not all security and access issues related to Messaging Server are treated in this chapter. Security topics that are discussed elsewhere include the following:

- **Physical security:** Without provisions for keeping server machines physically secure, software security can be meaningless.
- **Program delivery:** Program delivery of messages has significant security implications. For a discussion on this topic, see Chapter 12, "Program Delivery."
- **Encrypted messages (S/MIME):** With Secure Multipurpose Internet Mail Extensions, senders can encrypt messages prior to sending them, and recipients can store the encrypted messages after receipt, decrypting them only to read them. Using S/MIME requires no special Messaging Server configuration or tasks; it is strictly a client action. See your client

documentation for information on setting it up. Note that the Messenger Express client interface does not support encryption of email messages.

- **Message-store access:** You can define a set of message-store administrators for the Messaging Server. These administrators can view and monitor mailboxes and can control access to them. For details, see "Specifying Administrator Access to the Store" on page 154.

- **End-user account configuration:** Messaging Server provides limited end-user access (as HTML forms), through which your messaging users can view and change certain information (such as password and vacation message) in their own mail accounts. The end-user forms are described in online help; how to configure the forms is described in "Configuring End-User Information" on page 55.

- **Filtering unsolicited bulk email (UBE):** A Messaging Server plug-in provides flexible filtering options for preventing unwanted commercial email from clogging your users' mailboxes. The UBE plug-in and its use are described in "Filtering Unsolicited Bulk Email" on page 215.

Netscape has produced a large number of documents that cover a variety of security topics. For additional background on the topics mentioned here and for other security-related information, see the Security page on the Netscape DevEdge Web site.

About HTTP Security

Messaging Server supports the same security features for the HTTP protocol as it does for the IMAP protocol—both user ID/password authentication and client certificate authentication are supported. There are some differences, however, in how the protocols handle network connections between client and server.

When a POP, IMAP, or SMTP client logs in to Messaging Server, a connection is made and a session is established. The connection lasts for the duration of the session; that is, from login to logout. When establishing a new connection, the client must reauthenticate to the server.

When an HTTP client logs in to Messaging Server, the server provides a unique session ID to the client. The client uses the session ID to establish multiple connections during a session. The HTTP client need not reauthenticate for each

connection; the client need only reauthenticate if the session is dropped and the client wants to establish a new session. (If an HTTP session remains idle for a specified time period, the server will automatically drop the HTTP session and the client is logged out; the default time period is 2 hours.)

HTTP session connections remain secure because:

- The session IDs are bound to a specific IP address, so they cannot be used by another host.
- Each session ID has a timeout value associated with it; if the session ID is not used for a specified time period, the session ID becomes invalid.
- The server keeps a database of all open session IDs, so a client cannot forge an ID.
- The session ID is stored in the URL, but not in any cookie files.

For information about specifying configuration parameters for improved connection performance, see "Performance Parameters" on page 74.

User Password Login

Requiring password submission on the part of users logging into Messaging Server to send or receive mail is a first line of defense against unauthorized access. Messaging Server supports password-based login for its IMAP, POP, HTTP, and SMTP services.

IMAP, POP, and HTTP Password Login

By default, users must submit a password to retrieve their messages from Messaging Server. You enable or disable password login separately for POP, IMAP, and HTTP services. For more information about password login for POP IMAP, and HTTP Services, see "Password-Based Login" on page 73.

User passwords can be transmitted from the user's client software to your server as clear text or in encrypted form (IMAP and HTTP only). If both the client and your server are configured to enable SSL and both support encryption of the required strength (as explained in "Enabling SSL" on page 181), encryption occurs.

User IDs and passwords are stored in your installation's LDAP user directory. Password security criteria, such as minimum length, are determined by directory policy requirements; they are not part of Messaging Server administration.

Certificate-based login is an alternative to password-based login. It is discussed in this chapter along with the rest of SSL; see "Setting Up Certificate-Based Login" on page 184.

SMTP Password Login

By default, users need not submit a password when they connect to the SMTP service of Messaging Server to send a message. You can, however, enable password login to SMTP in order to enable authenticated SMTP.

Authenticated SMTP is an extension to the SMTP protocol, in which message-sender authentication accompanies a message, thus permitting the receiver of the message to know that the sender was authenticated by the sender's messaging server. Authenticated SMTP is designed for use within a set of trusted servers in which no server-to-server certificate authentication is being employed. For instructions on enabling SMTP password login (and thus Authenticated SMTP), see "Enabling Authenticated SMTP" on page 112.

Figure 6.1 summarizes how Authenticated SMTP works. When SMTP password login is enabled, clients and servers follow this process:

1. When the client connects to its SMTP server to send a message, it first transmits an `auth plain` command after its `EHLO` command to let the server know that authentication should accompany this message.

   The client typically requires the user to enter a password only once per session, regardless of how many messages the user sends.

2. When the sending client's server connects to the next server in the chain of transmission, it performs these two tasks related to client authentication:

   - It authenticates itself to the next server with the `AUTH` command. If the next server supports the `AUTH` command, a password exchange occurs. If the next server does not support `AUTH`, message transmission proceeds but client authentication is dropped.

- The server passes client authentication on to the next server, using the `AUTH` parameter of the `MAIL FROM` command.

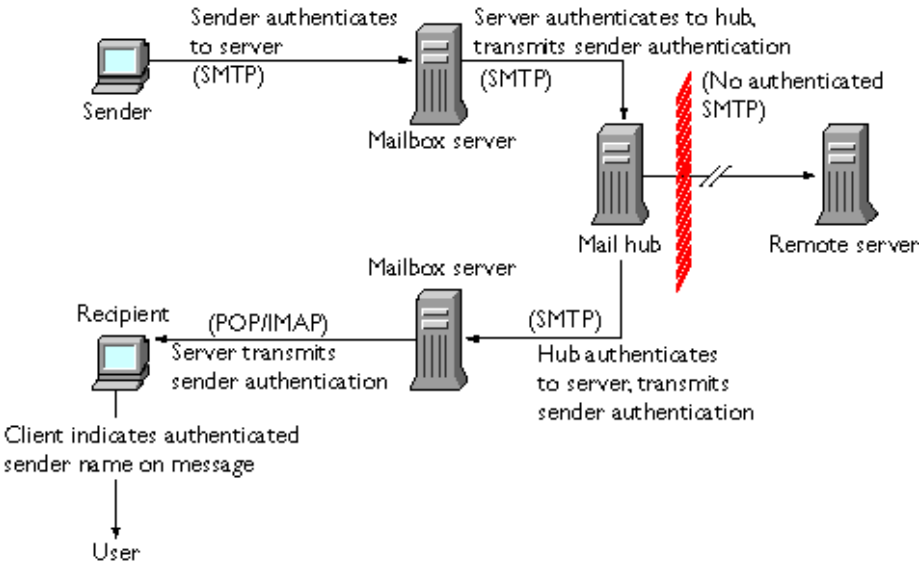
Note: Because server-password information is stored in an enterprise's LDAP directory, the authentication exchange can occur only between servers that share an LDAP directory—which typically means only servers within a single enterprise. Authenticated SMTP is usually not applied to external message transmission.

3. When the recipient's messaging server receives the message, it stores the message in the user's mailbox along with an indication that the sender's name is authenticated.

4. When the user's mail client retrieves the message, it appends an indication of the authentication to the message header. In the Netscape mail client, the word "Internal" appears next to the authenticated sender's name.

You can use Authenticated SMTP with or without SSL encryption.

Figure 6.1 Message transmission with Authenticated SMTP



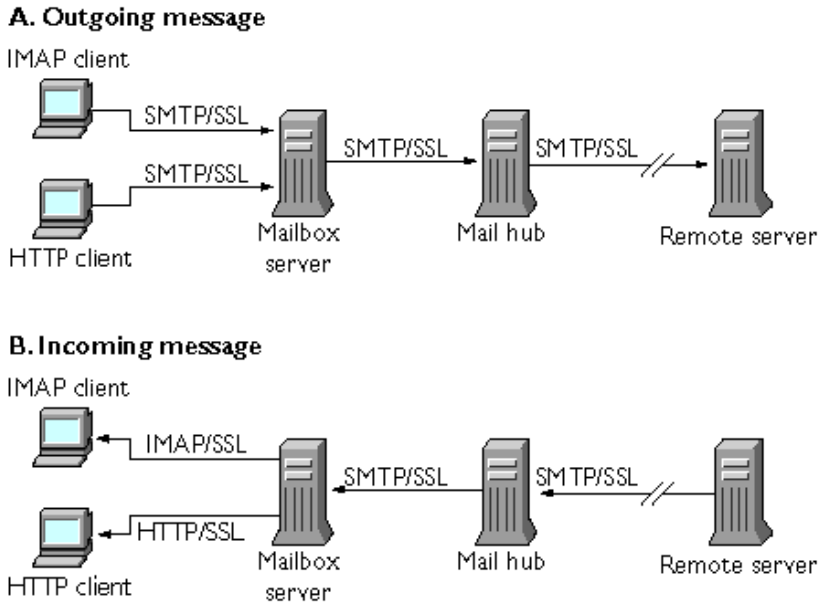
Configuring SSL Encryption and Authentication

Messaging Server uses the Secure Sockets Layer (SSL) protocol for encrypted communications and for certificate-based authentication of clients and servers. See *Introduction to SSL* (reproduced as an appendix to *Managing Servers with Netscape Console*) for background information on SSL. SSL itself is based on the concepts of public-key cryptography, described in *Introduction to Public-Key Cryptography* (also reproduced as an appendix to *Managing Servers with Netscape Console*).

If transmission of messages between a Messaging Server and its clients and between the server and other servers is encrypted, there is little chance for eavesdropping on the communications. If connecting clients are authenticated, there is little chance for intruders to impersonate (spoof) them.

SSL functions as a protocol layer beneath the application layers of IMAP4, HTTP, and SMTP. SMTP and SMTP/SSL use the same port; HTTP and HTTP/SSL require different ports; IMAP and IMAP/SSL can use the same port or different ports. SSL acts at a specific stage of message communication, as shown in Figure 6.2 for both outgoing and incoming messages.

Figure 6.2 Encrypted communications with Messaging Server



Complete end-to-end encryption of message transmission may require the use of all SSL-related protocols.

Note: To enable SSL encryption for outgoing messages, you must use the `configutil` command to modify the configuration parameter, `service.smtp.sslusesslrelay`. This parameter is not turned on by default. For more information on using `configutil`, see "configutil" on page 406.

Keep in mind that the extra overhead in setting up an SSL connection can put a performance burden on the server. In designing your messaging installation and in analyzing performance, you may need to balance security needs against server capacity.

Netscape Messaging Server supports only SSL version 3.0.

Note: Because all Netscape servers support SSL, and the interface for enabling and configuring SSL through Netscape Console is nearly identical across all servers, several of the tasks described in this section are documented more completely in the SSL chapter of *Managing Servers with Netscape Console*. For those tasks, this chapter gives summary information only.

Obtaining Certificates

Whether you use SSL for encryption or for authentication, you need to obtain a server certificate for your Messaging Server. The certificate identifies your server to clients and to other servers.

Managing Internal and External Modules

A server certificate establishes the ownership and validity of a key pair, the numbers used to encrypt and decrypt data. Your server's certificate and key pair represent your server's identity. They are stored in a certificate database that can be either internal to the server or on an external, removable hardware card (smartcard).

Likewise, the software module that manages the keys and certificates database can be either internal or external. Netscape servers support both internal and external modules that conform to the Public-Key Cryptography System (PKCS) #11 protocol.

Setting up the server for a certificate involves creating a database for the certificate and its keys and installing a PKCS #11 module. If you do not use an external hardware token, you create an internal database on your server, and you use the internal, default module that is part of Messaging Server. If you do use an external token, you connect a hardware smartcard reader and install its PKCS #11 module.

You can manage PKCS #11 modules, whether internal or external, through Netscape Console. To install a PKCS #11 module:

1. Connect a hardware card reader to the Messaging Server host machine and install drivers.

2. Use the PKCS #11 Management interface in Netscape Console to connect the PKCS #11 module to the installed driver.

(For more complete instructions, see the chapter on SSL in *Managing Servers with Netscape Console*.)

Installing Hardware Encryption Accelerators. If you use SSL for encryption, you may be able to improve server performance in encrypting and decrypting messages by installing a hardware encryption accelerator. An encryption accelerator typically consists of a hardware board, installed

permanently in your server machine, plus a software driver. Netscape Messaging Server supports accelerator modules that follow the PKCS #11 protocol. (They are essentially hardware tokens that do not store their own keys; they use the internal database for that.) You install an accelerator by first installing the hardware and drivers as specified by the manufacturer, and then completing the installation—as with hardware certificate tokens—by installing the PKCS #11 module.

Requesting a Server Certificate

You request a server certificate by opening your server in Netscape Console and running the Certificate Setup Wizard. You can access the Wizard from the Console menu or from the Messaging Server Encryption tab. Using the Wizard, you perform the following tasks:

1. Generate a certificate request.
2. Send the request by email to the certificate authority (CA) that is to issue the certificate.

When the email response from the CA arrives, you save it as a text file.

(For more complete instructions, see the chapter on SSL in *Managing Servers with Netscape Console*.)

Creating a Password File

On any Netscape server, when you use the Certificate Setup Wizard to request a certificate, the wizard creates a key pair to be stored in either the internal module's database or in an external database (on a smartcard). The wizard then prompts you for a password, which it uses to encrypt the stored key pair. Only that same password can later be used to decrypt the keys. The wizard does not retain the password nor store it anywhere.

On most Netscape servers for which SSL is enabled, the administrator is prompted at startup to supply the password required to decrypt the key pair. On Messaging Server 4.1, however, to alleviate the inconvenience of having to enter the password multiple times (it is needed by at least three server processes), and to facilitate unattended server restarts, the password is read from a password file.

The password file is named `sslpassword.conf` and is in the directory *`installDirectory`*`/config/`. Entries in the file are individual lines with the format

```
moduleName:password
```

where *`moduleName`* is the name of the (internal or external) PKCS #11 module to be used, and *`password`* is the password that decrypts that module's key pair. The password is stored as clear (unencrypted) text.

Messaging Server 4.1 provides a default version of the password file, with the following single entry (for the internal module and default password):

```
Communicator Certificate DB:netscape!
```

If you specify anything but the default password when you install an internal certificate, you need to edit the above line of the password file to reflect the password you specified. If you install an external module, you need to add a new line to the file, containing the module name and the password you specified for it.

Caution: Because the administrator is not prompted for the module password at server startup, it is especially important that you ensure proper administrator access control to the server and proper physical security of the server host machine and server backups.

Installing the Certificate

Installing is a separate process from requesting. Once the email response to your request for a certificate has arrived from the CA and been saved as a text file, run the Certificate Setup Wizard once more to install the file as a certificate:

1. Specify that you are installing a certificate that you have already obtained.
2. Paste the text of your certificate into a field when prompted to do so.

(For more complete instructions, see the chapter on SSL in *Managing Servers with Netscape Console*.)

Note: This is also the process you follow to install a CA certificate (described next), which your server uses to determine whether to trust the certificates presented by clients.

Installing Certificates of Trusted CAs

You also use the Certificate Setup Wizard to install the certificates of certificate authorities. A CA certificate validates the identity of the CA itself. Your server uses these CA certificates in the process of authenticating clients and other servers.

If, for example, you set up your enterprise for certificate-based client authentication in addition to password-based authentication (see "Setting Up Certificate-Based Login" on page 184), you need to install the CA certificates of all CAs that are trusted to issue the certificates that your clients may present. These CAs may be internal to your organization or they may be external, representing commercial or governmental authorities or other enterprises. (See *Introduction to Cryptography* for more details on the use of CA certificates for authentication.)

When installed, Messaging Server initially contains CA certificates for several commercial CAs. If you need to add other commercial CAs or if your enterprise is developing its own CA for internal use (using Netscape Certificate Server), you need to obtain and install additional CA certificates.

Note: The CA certificates automatically provided with Messaging Server are not initially marked as trusted. You need to edit the trust settings if you want to trust client certificates issued by these CAs. For instructions, see "Managing Certificates and Trusted CAs" on page 181.

To request and install a new CA certificate, you:

1. Contact the certificate authority (possibly through the Web or by email) and request a CA certificate.
2. Save the received text of the certificate as a text file.
3. Use the Certificate Setup Wizard, as described in the previous section, to install the certificate.

(For more complete instructions, see the chapter on SSL in *Managing Servers with Netscape Console*.)

Managing Certificates and Trusted CAs

Your server can have any number of certificates of trusted CAs that it uses for authentication of clients.

You can view, edit the trust settings of, or delete any of the certificates installed in your Messaging Server by opening your server in Netscape Console and choosing the Certificate Management Command in the Console menu. For instructions, see the chapter on SSL in *Managing Servers with Netscape Console*.

Enabling SSL

You can use Netscape Console to enable SSL and to select the set of encryption ciphers that Messaging Server can use in its encrypted communications with clients.

About Ciphers

A *cipher* is the algorithm used to encrypt and decrypt data in the encryption process. Some ciphers are stronger than others, meaning that a message they have scrambled is more difficult for an unauthorized person to unscramble.

A cipher operates on data by applying a key—a long number—to the data. Generally, the longer the key the cipher uses during encryption, the harder it is to decrypt the data without the proper decryption key.

When a client initiates an SSL connection with a Messaging Server, the client lets the server know what ciphers and key lengths it prefers to use for encryption. In any encrypted communication, both parties must use the same ciphers. Because there are a number of cipher-and-key combinations in common use, a server should be flexible in its support for encryption. Netscape Messaging Server 4.1 can support up to 6 combinations of cipher and key length.

Table 6.1 lists the ciphers that Messaging Server supports for use with SSL 3.0. The table summarizes information that is available in more detail in *Introduction to SSL*; please consult that document before making final decisions on what ciphers to support.

Table 6.1 SSL ciphers for Messaging Server

Cipher	Description
RC4 with 128-bit encryption and MD5 message authentication	The fastest encryption cipher (by RSA) and a very high-strength combination of cipher and encryption key. Suitable only for domestic North American (non-export) use.
Triple DES with 168-bit encryption and SHA message authentication	A slower encryption cipher (a U.S. government-standard) but the highest-strength combination of cipher and encryption key. Suitable only for domestic North American (non-export) use.
DES with 56-bit encryption and SHA message authentication	A slower encryption cipher (a U.S. government-standard) and a moderate-strength combination of cipher and encryption key. Suitable only for domestic North American (non-export) use.
RC4 with 40-bit encryption and MD5 message authentication	The fastest encryption cipher (by RSA) and a lower-strength combination of cipher and encryption key. Suitable for international use.
RC2 with 40-bit encryption and MD5 message authentication	A slower encryption cipher (by RSA) and a lower-strength combination of cipher and encryption key. Suitable for international use.
No encryption, only MD5 message authentication	No encryption; use of a message digest for authentication alone.

Unless you have a compelling reason for not using a specific cipher, you should support them all. However, note that export laws restrict the use of certain encryption ciphers in certain countries. Basically, key lengths of greater than 40 bits can be used only in the United States and Canada. For details, see *Export Restrictions on International Sales* on the Netscape DevEdge Web site. In general, Messaging Server cannot use the higher-strength encryption for any communication with international versions of client software.

Enabling SSL and Selecting Ciphers

To enable SSL and select encryption ciphers, follow these steps:

1. In Netscape Console, open the Messaging Server whose cipher settings you want to modify.
2. Click the Configuration tab in the left pane and select the Services folder.
3. Click the Encryption tab in the right pane.
4. Check the Enable SSL box to enable SSL on your server.
5. Check the RSA box if you want to enable RSA ciphers.
6. Check the Fortezza box if you want to enable Fortezza ciphers.

   You'll see the Fortezza box only if you have an export version of the server.
7. From the Token drop-down list, choose the token you want to use.
8. From the Certificate drop-down list, choose the certificate you want to use.
9. Click Cipher Preferences to open the list of available ciphers.
10. Click the boxes to select the encryption cipher or ciphers that you want your server to support.

To disable SSL completely, deselect the Enable SSL box.

Note: To enable SSL encryption for outgoing messages, you must use the `configutil` command to modify the configuration parameter, `service.smtp.sslusesslrelay`. This parameter is not turned on by default. For more information on using `configutil`, see "configutil" on page 406.

Command Line

You can also set values for these parameters at the command line as follows:

To enable or disable SSL:

```
configutil -o nsserversecurity -v [ on | off ]
```

To enable or disable RSA ciphers:

```
configutil -o encryption.rsa.nssslactivation -v [ on | off ]
```

To specify a token:

```
configutil -o encryption.rsa.nsssltoken -v tokenname
```

To specify a certificate:

```
configutil -o encryption.rsa.nssslpersonalityssl -v certname
```

Note: If you enable RSA ciphers, you must also specify a token and a certificate.

To choose a cipher preference:

```
configutil -o encryption.nsssl3ciphers -v cipherlist
```

where *cipherlist* is a comma-separated list of ciphers.

Setting Up Certificate-Based Login

In addition to password-based authentication, Netscape servers support authentication of users through examination of their digital certificates. In certificate-based authentication, the client establishes an SSL session with the server and submits the user's certificate to the server. The server then evaluates whether the submitted certificate is genuine. If the certificate is validated, the user is considered authenticated.

To set up your Messaging Server for certificate-based login:

1. Obtain a server certificate for your server. (For details, see "Obtaining Certificates" on page 177.)
2. Run the Certificate Setup Wizard to install the certificates of any trusted certificate authorities that will issue certificates to the users your server will authenticate. (For details, see "Installing Certificates of Trusted CAs" on page 180.)

   Note that, as long as there is at least one trusted CA in the server's database, the server requests a client certificate from each connecting client.
3. Turn on SSL. (For details, see "Enabling SSL" on page 181.)

4. (Optional) Edit your server's `certmap.conf` file so that the server appropriately searches the LDAP user directory based on information in the submitted certificates.

   Editing the `certmap.conf` file is not necessary if the email address in your users' certificates matches the email address in your users' directory entries, and if you do not need to optimize searches or validate the submitted certificate against a certificate in the user entry.

   For details of the format of `certmap.conf` and the changes you can make, see the SSL chapter of *Managing Servers with Netscape Console*.

Once you have taken these steps, when a client establishes an SSL session so that the user can log in to IMAP or HTTP, the Messaging Server requests the user's certificate from the client. If the certificate submitted by the client has been issued by a CA that the server has established as trusted, and if the identity in the certificate matches an entry in the user directory, the user is authenticated and access is granted (depending on access-control rules governing that user).

There is no need to disallow password-based login (see "Password-Based Login" on page 73) to enable certificate-based login. If password-based login is allowed (which is the default state), and if you have performed the tasks described in this section, both password-based and certificate-based login are supported. In that case, if the client establishes an SSL session and supplies a certificate, certificate-based login is used. If the client does not use SSL or does not supply a certificate, the server requests a password.

For more details on setting up your entire installation of Netscape servers and clients to use certificate-based authentication, see the *Single Sign-On Deployment Guide*.

Configuring Administrator Access to Messaging Server

This section describes how to control the ways in which server administrators can gain access to Messaging Server. Administrative access to a given Messaging Server and to specific Messaging Server tasks occurs within the context of delegated administration. *Delegated administration* is a feature of all Netscape servers; it refers to the capability of an administrator to provide other

administrators with selective access to individual servers and server features. For an overview of delegated administration, see the chapter on delegating server administration in *Managing Servers with Netscape Console*.

Note: Most tasks described in this section are common to all Netscape servers, and are therefore described fully only in the chapter on delegating server administration in *Managing Servers with Netscape Console*. This chapter briefly summarizes the tasks.

Hierarchy of Delegated Administration

When you install the first Netscape server on your network, the installation program automatically creates a group in the LDAP user directory called the Configuration Administrators group. By default, the members of the Configuration Administrators group have unrestricted access to all hosts and servers on your network.

The Configuration Administrators group is at the top of an access hierarchy, such as the following, that you can create to implement delegated administration for Messaging Server:

1. **Configuration administrator.** The "super user" for the network of Netscape servers. Has complete access to all resources.

2. **Domain administrator.** The configuration administrator typically creates one or more other groups, each with more restricted access. For example the configuration administrator might create a Domain Administrators group and give it access to all servers for a specific administrative domain in the network. (An *administrative domain* is the set of hosts and servers on the network that share the same user directory for looking up account information.)

3. **Server administrator.** A domain administrator might create groups to administer each type of server. For example, a Messaging Administrators group might be created to administer all Messaging Servers in an administrative domain or across the whole network. Members of that group have access to all Messaging Servers (but no other servers) in that administrative domain.

4. **Task administrator.** Finally, any of the above administrators might create a group, or designate an individual user, with restricted access to a single Messaging Server or a set of Messaging Servers. Such a task administrator is permitted to perform only specific, limited server tasks (such as starting or stopping the server only, or accessing logs of a given service).

Netscape Console provides convenient interfaces that allow an administrator to perform the following tasks:

- Grant a group or an individual access to a specific Messaging Server, as described in "Providing Access to the Server as a Whole" (next).
- Restrict that access to specific tasks on a specific Messaging Server, as described in "Restricting Access to Specific Tasks" on page 188.

Providing Access to the Server as a Whole

To give a user or group permission to access a given instance of Messaging Server, you:

1. Log in to Netscape Console as an administrator with access to the Messaging Server you want to provide access to.
2. Select that server in the Console window, and choose Set Access Permissions in the Console menu.
3. Add or edit the list of users and groups with access to the server.

(For more complete instructions, see the chapter on delegating server administration in *Managing Servers with Netscape Console*.)

Once you have set up the list of individuals and groups that have access to the particular Messaging Server, you can then use ACIs, as described next, to delegate specific server tasks to specific people or groups on that list.

Restricting Access to Specific Tasks

An administrator typically connects to a server to perform one or more administrative tasks. Common administrative tasks are listed in the Messaging Server Tasks form in Netscape Console (see Figure 1.13).

By default, access to a particular Messaging Server means access to all of its tasks. However, each task in the Task form can have an attached set of access-control instructions (ACIs). The server consults those ACIs before giving a connected user (who must already be a user with access permissions to the server as a whole) access to any of the tasks. In fact, the server displays in the Tasks form only those tasks to which the user has permission.

If you have access to a Messaging Server, you can create or edit ACIs on any of the tasks (that is, on any of the tasks to which you have access), and thus restrict the access that other users or groups can have to them.

To restrict the task access that a connected user or group can have, you:

1. Log in to the Netscape Console as an administrator with access to the Messaging Server you want to provide restricted access to.
2. Open the server and select a task in the server's Tasks form.
3. From the Edit menu, choose Set Access Permissions, and add or edit the list of access rules to give a user or group the kind of access you want them to have.
4. Repeat the process for other tasks, as appropriate.

(For more complete instructions, see the chapter on delegating server administration in *Managing Servers with Netscape Console*.)

For example, suppose you want to create a group of administrators responsible for log analysis, and you also have an individual (Dimitria) who is responsible for filtering out unsolicited bulk email (UBE). You can:

1. Create a group called Logging Admins and add the appropriate people to it.
2. Use Set Access Permissions for the Netscape Console Configuration window to give both Logging Admins and Dimitria access to the Messaging Server.

3. Use Set Access Permissions for the Messaging Server Tasks form to place ACIs on the various tasks, making sure that the Logging Admin group sees (and thus has access to) only the logging-related tasks, and that Dimitria sees only the UBE filter-configuration task.

ACIs and how to create them are described more fully in the chapter on delegating server administration in *Managing Servers with Netscape Console*.

Configuring Client Access to TCP Services

Messaging Server supports sophisticated access control on a service-by-service basis for its TCP-based services (IMAP, POP, HTTP, and SMTP), so that you can exercise far-ranging and fine-grained control over which clients can gain access to your server.

If you are managing messaging services for a large enterprise or an Internet service provider, these capabilities can help you to exclude spammers and DNS spoofers from your system and improve the general security of your network. For control of spam mail (unsolicited bulk email) specifically, see also Chapter 8, "Filtering Unsolicited Bulk Email."

Note: If controlling user access is *not* an important issue for your enterprise, you do not have to create any of the filters described in this section. If minimal access control is all you need, see the section "Mostly Allowing" on page 196 for instructions on setting it up.

How Client Access Filters Work

The Messaging Server access-control facility for TCP clients is an implementation of the TCP wrapper concept. A *TCP wrapper* is a program that listens at the same port as the TCP daemon it serves; it uses access filters to verify client identity, and it gives the client access to the daemon if the client passes the filtering process. The design of the Messaging Server TCP wrapper is based on the Unix `Tcpd` access-control facility (created by Wietse Venema) and the `identd` service (described in Internet draft RFC 1413).

As part of its processing, the Messaging Server TCP client access-control system performs (when necessary) the following analyses of the socket end-point addresses:

- Reverse DNS lookups of both end points (to perform name-based access control)
- Forward DNS lookups of both end points (to fight DNS spoofing)
- `Identd` callback (to check that the user on the client end is known to the client host)

The system compares this information against access-control statements called *filters* to decide whether to grant or deny access. For each service, separate sets of Allow filters and Deny filters control access. Allow filters explicitly grant access; Deny filters explicitly forbid access.

When a client requests access to a service, the access-control system compares the client's address or name information to each of that service's filters—in order—using these criteria:

- The search stops at the first match. Because Allow filters are processed before Deny filters, Allow filters take precedence.
- Access is granted if the client information matches an Allow filter for that service.
- Access is denied if the client information matches a Deny filter for that service.
- If no match with any Allow or Deny filter occurs, access is granted—except in the case where there are Allow filters but no Deny filters, in which case lack of a match means that access is denied.

The filter syntax described here is flexible enough that you should be able to implement many different kinds of access-control policies in a simple and straightforward manner. You can use both Allow filters and Deny filters in any combination, even though you can probably implement most policies by using almost exclusively Allows or almost exclusively Denies.

The following sections describe filter syntax in detail and give usage examples. The section "Creating Access Filters for Services" on page 198 gives the procedure for creating access filters.

Filter Syntax

Filter statements contain both server information and client information. The server information can include the name of the service, names of hosts, and addresses of hosts. The client information can include host names, host addresses, and user names. Both the server and client information can include wildcard names or patterns.

The very simplest form of a filter is:

```
service: hostSpec
```

where *service* is the name of the service (such as `smtp`, `pop`, `imap`, or `http`) and *hostSpec* is the host name, IP address, or wildcard name or pattern that represents the client requesting access. When a filter is processed, if the client seeking access matches *hostSpec*, access is either allowed or denied (depending on which type of filter this is) to the service specified by *service*. Here are some examples:

```
imap: roberts.newyork.airius.com

pop: ALL

http: ALL
```

If these are Allow filters, the first one grants the host `roberts.newyork.airius.com` access to the IMAP service, and the second and third grant all clients access to the POP and HTTP services, respectively. If they are Deny filters, they deny those clients access to those services. (For descriptions of wildcard names such as `ALL`, see "Wildcard Names" on page 192.)

Either the server or the client information in a filter can be somewhat more complex than this, in which case the filter has the more general form of:

```
serviceSpec: clientSpec
```

where *serviceSpec* can be either *service* or *service@hostSpec*, and *clientSpec* can be either *hostSpec* or *user@hostSpec*. *user* is the user name (or a wildcard name) associated with the client host seeking access. Here are two examples:

```
smtp@mailServer1.airius.com: ALL

imap: srashad@xyz.europe.airius.com
```

If these are Deny filters, the first filter denies all clients access to the SMTP service on the host `mailServer1.airius.com`. The second filter denies the user `srashad` at the host `xyz.europe.airius.com` access to the IMAP service. (For more information on when to use these expanded server and client specifications, see "Server-Host Specification" on page 194 and "Client User-Name Specification" on page 195.)

Finally, at its most general, a filter has the form:

```
serviceList: clientList
```

where *`serviceList`* consists of one or more *`serviceSpec`* entries, and *`clientList`* consists of one or more *`clientSpec`* entries. Individual entries within *`serviceList`* and *`clientList`* are separated by blanks and/or commas.

In this case, when a filter is processed, if the client seeking access matches any of the *`clientSpec`* entries in *`clientList`*, then access is either allowed or denied (depending on which type of filter this is) to all the services specified in *`serviceList`*. Here is an example:

```
pop, imap, http: .europe.airius.com .newyork.airius.com
```

If this is an Allow filter, it grants access to POP, IMAP, and HTTP services to all clients in either of the domains `europe.airius.com` and `newyork.airius.com`. For information on using a leading dot or other pattern to specify domains or subnet, see "Wildcard Patterns" on page 193.

Wildcard Names

You can use the following wildcard names to represent service names, host names or addresses, or user names:

Table 6.2 Wildcard names

Wildcard Name	Explanation
ALL	The universal wildcard. Matches all names.
LOCAL	Matches any local host (one whose name does not contain a dot character). However, if your installation uses only canonical names, even local host names will contain dots and thus will not match this wildcard.

Table 6.2 Wildcard names (Continued)

Wildcard Name	Explanation
UNKNOWN	Matches any user whose name is unknown, or any host whose name or address is unknown. Use this wildcard name carefully: • Host names may be unavailable due to temporary DNS server problems—in which case all filters that use UNKNOWN will match all client hosts. • A network address is unavailable when the software cannot identify the type of network it is communicating with—in which case all filters that use UNKNOWN will match all client hosts on that network.
KNOWN	Matches any user whose name is known, or any host whose name *and* address are known. Use this wildcard name carefully: • Host names may be unavailable due to temporary DNS server problems—in which case all filters that use KNOWN will fail for all client hosts. • A network address is unavailable when the software cannot identify the type of network it is communicating with—in which case all filters that use KNOWN will fail for all client hosts on that network.
DNSSPOOFER	Matches any host whose DNS name does not match its own IP address.

Wildcard Patterns

You can use the following patterns in server or client addresses:

- A string that begins with a dot character (`.`). A host name is matched if the last components of its name match the specified pattern. For example, the wildcard pattern `.airius.com` matches all hosts in the domain `airius.com`.

- A string that ends with a dot character (`.`). A host address is matched if its first numeric fields match the specified pattern. For example, the wildcard pattern `123.45.` matches the address of any host in the subnet `123.45.0.0`.
- A string of the form `n.n.n.n/m.m.m.m`. This wildcard pattern is interpreted as a *`net/mask`* pair. A host address is matched if *`net`* is equal to the bitwise AND of the address and *`mask`*. For example, the pattern `123.45.67.0/255.255.255.128` matches every address in the range `123.45.67.0` through `123.45.67.127`.

EXCEPT Operator

The access-control system supports a single operator. You can use the `EXCEPT` operator to create exceptions to matching names or patterns when you have multiple entries in either *`serviceList`* or *`clientList`*. For example, the expression:

```
list1 EXCEPT list2
```

means that anything that matches *`list1`* is matched, *unless* it also matches *`list2`*.

Here is an example:

```
ALL: ALL EXCEPT isserver.airius.com
```

If this were a Deny filter, it would deny access to all services to all clients except those on the host machine `isserver.airius.com`.

`EXCEPT` clauses can be nested. The expression:

```
list1 EXCEPT list2 EXCEPT list3
```

is evaluated as if it were:

```
list1 EXCEPT (list2 EXCEPT list3)
```

Server-Host Specification

You can further identify the specific service being requested in a filter by including server host name or address information in the *`serviceSpec`* entry. In that case the entry has the form:

```
service@hostSpec
```

You might want to use this feature when your Messaging Server host machine is set up for multiple internet addresses with different internet host names. If you are a service provider, you can use this facility to host multiple domains, with different access-control rules, on a single server instance.

Client User-Name Specification

For client host machines that support the `identd` service as described in RFC 1413, you can further identify the specific client requesting service by including the client's user name in the *`clientSpec`* entry in a filter. In that case the entry has the form:

`user@hostSpec`

where *`user`* is the user name as returned by the client's `identd` service (or a wildcard name).

Specifying client user names in a filter can be useful, but keep these caveats in mind:

- The `identd` service is not authentication; the client user name it returns cannot be trusted if the client system has been compromised. In general, do not use specific user names; use only the wildcard names `ALL`, `KNOWN`, or `UNKNOWN`.

- User-name lookups take time; performing lookups on all users may slow access by clients that do not support `identd`. Selective user-name lookups can alleviate this problem. For example, a rule like:

  `serviceList``: @xyzcorp.com ALL@ALL`

  would match users in the domain `xyzcorp.com` without doing user-name lookups, but it would perform user-name lookups with all other systems.

The user-name lookup capability can in some cases help you guard against attack from unauthorized users on the client's host. It is possible in some TCP/IP implementations, for example, for intruders to use `rsh` (remote shell service) to impersonate trusted client hosts. If the client host supports the `ident` service, you can use user-name lookups to detect such attacks. For an example and discussion, see "Allowing Only Identified Users" on page 197.

Filter Examples

The examples in this section show a variety of approaches to controlling access. In studying the examples, keep in mind that Allow filters are processed before Deny filters, the search terminates when a match is found, and access is granted when no match is found at all.

The examples listed here use host and domain names rather than IP addresses. Remember that you can include address and netmask information in filters, which can improve reliability in the case of name-service failure.

Mostly Denying

In this case, access is denied by default. Only explicitly authorized hosts are permitted access.

The default policy (no access) is implemented with a single, trivial deny file:

```
ALL: ALL
```

This filter denies all service to all clients that have not been explicitly granted access by an Allow filter. The Allow filters, then, might be something like these:

```
ALL: LOCAL @netgroup1
ALL: .airius.com EXCEPT externalserver.airius.com
```

The first rule permits access from all hosts in the local domain (that is, all hosts with no dot in their host name) and from members of the group `netgroup1`. The second rule uses a leading-dot wildcard pattern to permit access from all hosts in the `airius.com` domain, with the exception of the host `externalserver.airius.com`.

Mostly Allowing

In this case, access is granted by default. Only explicitly specified hosts are denied access.

The default policy (access granted) makes Allow filters unnecessary. The unwanted clients are listed explicitly in Deny filters such as these:

```
ALL: externalserver.airius1.com, .airius.asia.com
ALL EXCEPT pop: contractor.airius1.com, .airius.com
```

The first filter denies all services to a particular host and to a specific domain. The second filter permits nothing but POP access from a particular host and from a specific domain.

Allowing Only Identified Users

You can use the following Deny filter to exclude all users but those known to a client host's `identd` service:

```
ALL: UNKWOWN@ALL
```

The filter denies all services to all unknown users from all domains.

You could write a more specific Deny filter, with a *`clientSpec`* entry of `UNKNOWN@`*`host`*. When it receives a request from *`host`*, the access-control system then uses the `ident` service on *`host`* to find out whether that host actually sent the request and what the user name of the requestor is. If the host responds that the sending user is unknown, that may be evidence of an attack. (However, note also that, if the client's host does not support the `identd` service, all requestors will match the `UNKNOWN@`*`host`* filter.)

Employing user-name lookups in Allow filters is less trustworthy. Suppose you write an Allow filter with a *`clientSpec`* entry of `KNOWN@`*`host`*. Because an intruder can spoof both the client connection and the `ident` lookup, a match with the `KNOWN@`*`host`* filter is not strong evidence of absence of spoofing. Furthermore, if the client system has been compromised (as noted earlier), `ident` may return false information.

For more information, see "Client User-Name Specification" on page 195.

Denying Access to Spoofed Domains

You can use the `DNSSPOOFER` wildcard name in a filter to detect host-name spoofing. When you specify `DNSSPOOFER`, the access-control system performs forward or reverse DNS lookups to verify that the client's presented host name matches its actual IP address. Here is an example for a Deny filter:

```
ALL: DNSSPOOFER
```

This filter denies all services to all remote hosts whose IP addresses don't match their DNS host names.

Controlling Access to Virtual Domains

If your messaging installation uses virtual domains, in which a single server instance is associated with multiple IP addresses and domain names, you can control access to each virtual domain through a combination of Allow and Deny filters. For example, you can use Allow filters like:

```
ALL@msgServer.airius1.com: @.airius1.com
ALL@msgServer.airius2.com: @.airius2.com
...
```

coupled with a Deny filter like:

```
ALL: ALL
```

Each Allow filter permits only hosts within `domain`*`N`* to connect to the service whose IP address corresponds to `msgServer.airius`*`N`*`.com`. All other connections are denied.

Denying an Individual User

If you must deny access to an especially notorious individual user, the most general Deny filter you can apply is the following:

```
ALL: badUser@ALL
```

This filter cannot, of course, guard against the same person attempting to gain access under a different user name.

Creating Access Filters for Services

You can create Allow and Deny filters for the IMAP, POP, HTTP, or SMTP services. Follow these steps to create filters:

1. In Netscape Console, open the Messaging Server that you want to create access filters for.
2. Click the Configuration tab.
3. Open the Services folder in the left pane and select IMAP, POP, HTTP, or SMTP beneath the Services folder.

4. Click the Access tab in the right pane.

   The Allow and Deny fields in the tab show the existing Allow and Deny filters for that service. Each line in the field represents one filter. For either of the fields, you can specify the following actions:

   - Click Add to create a new filter. An Allow Filter window or Deny filter window opens; enter the text of the new filter into the window, and click OK.
   - Select a filter and click Edit to modify the filter. An Allow Filter window or Deny filter window opens; edit the text of the filter displayed in the window, and click OK.
   - Select a filter and click Delete to remove the filter.

Note: If you need to rearrange the order of Allow or Deny filters, you can do so by performing a series of Delete and Add actions.

For a specification of filter syntax and a variety of examples, see "Filter Syntax" on page 191. For additional examples, see "Filter Examples" on page 196.

Command Line

You can also specify access and deny filters at the command line as follows:

To create or edit access filters for services:

```
configutil -o service.service.domainallowed -v filter
```

where *service* is `smtp`, `pop`, `imap`, or `http` and *filter* follows the syntax rules described in "Filter Syntax" on page 191.

To create or edit deny filters for services:

```
configutil -o service.service.domainnotallowed -v filter
```

where *service* is `smtp`, `pop`, `imap`, or `http` and *filter* follows the syntax rules described in "Filter Syntax" on page 191.

Creating Access Filters for HTTP Proxy Authentication

Any store administrator can proxy authenticate to any service. (For more information about store administrators, see "Specifying Administrator Access to the Store" on page 154.) For the HTTP service only, any user can proxy authenticate to the service if the user is granted access via a proxy authentication access filter.

Proxy authentication allows other services, such as a portal site, to authenticate users and pass the authentication credentials to the HTTP login service. For example, assume a portal site offers several services, one of which is Messenger Express web-based email. By using the HTTP proxy authentication feature, end users need only authenticate once to the portal service; they need not authenticate again to access their email. The portal site must configure a login server that acts as the interface between the client and the service. To help configure the login server for Messenger Express authentication, Netscape offers an authentication SDK for Messenger Express.

This section describes how to create allow filters for HTTP proxy authentication. This section does not describe how to set up your login server or how to use the Messenger Express authentication SDK. For more information about setting up your login server for Messenger Express and using the authentication SDK, contact your Netscape representative.

To create access filters for proxy authentication to the HTTP service:

1. In Netscape Console, open the Messaging Server that you want to create access filters for.
2. Click the Configuration tab.
3. Open the Services folder in the left pane and select HTTP beneath the Services folder.
4. Click the Proxy tab in the right pane.

   The Allow field in the tab shows the existing Allow filters for proxy authentication.
5. To create a new filter, click Add.

An Allow filter window opens. Enter the text of the new filter into the window and click OK.

6. To edit an existing filter, select the filter and click Edit.

   An Allow filter window opens. Edit the text of the filter display din the window, and click OK.

7. To delete an existing filter, select a field from the Allow field, and click Delete.

8. When you are finished making changes to the Proxy tab, click Save.

For more information about allow filter syntax, see "Filter Syntax" on page 191.

Command Line

You can also specify access filters for proxy authentication to the HTTP service at the command line as follows:

```
configutil -o service.service.proxydomainallowed -v filter
```

where *filter* follows the syntax rules described in "Filter Syntax" on page 191.

Chapter

7

Working with SMTP Plug-Ins

This chapter describes how to install and configure SMTP plug-ins, dynamic libraries that extend the capabilities of Netscape Messaging Server. For instructions on how to develop SMTP plug-ins, and for a detailed description of the plug-in programming interface, see *Messaging Server Plug-In API Guide*.

For information on using the SMTP plug-ins that allow you to filter unsolicited bulk email (UBE) and prevent relay of UBE from your server, see Chapter 8, "Filtering Unsolicited Bulk Email."

This chapter contains the following sections:

- About SMTP Plug-Ins
- Managing SMTP Plug-Ins with Netscape Console
- Managing SMTP Plug-Ins Manually
- Installing and Configuring Protocol-Level Plug-Ins

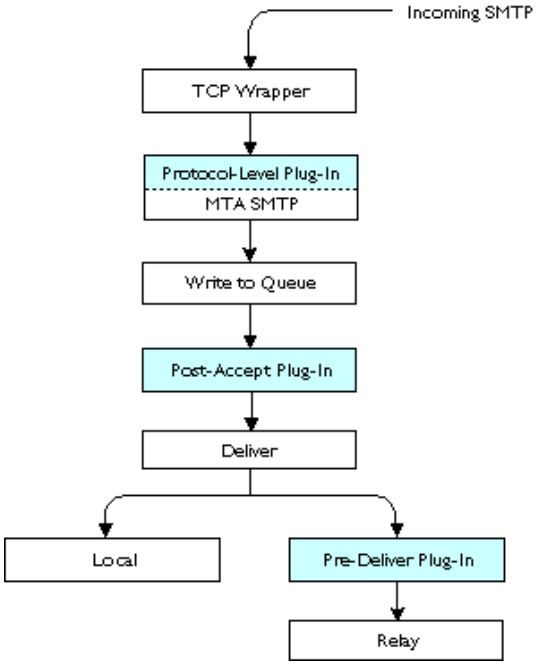
About SMTP Plug-Ins

There are three kinds of SMTP plug-in, each of which act at a different stage in message processing, as shown in Figure 7.1 .

- PreSMTPAccept (protocol-level) plug-ins act before SMTP has accepted the incoming message and written it to disk.

- PostSMTPAccept plug-ins act immediately after the message is received.
- PreSMTPDeliver plug-ins act just before the message is handed off to another host.

Figure 7.1 SMTP Plug-Ins



The SMTP plug-ins that you can use with Netscape Messaging Server might include the following:

- The UBE and anti-relay plug-ins supplied by Netscape
- Plug-ins that your enterprise has developed internally
- Commercially developed plug-ins that you have purchased

As an administrator, you control which plug-ins are installed and active at any given time. Only one plug-in (the UBE plug-in) is installed automatically when you install Messaging Server. Messaging Server includes the following two pre-built plug-ins that are described in Chapter 8, "Filtering Unsolicited Bulk Email":

- The UBE PostSMTPAccept plug-in provides a flexible, customizable filtering capability for removing or redirecting unwanted messages. This plug-in is automatically installed when you install Messaging Server.
- The protocol-level (PreSMTPAccept) anti-relay plug-in is used to prevent others from disguising the UBE that they send out to appear as if it is coming from your site.

Plug-ins might have very different purposes, but they are all installed, configured, and activated in standard ways:

- The PostSMTPAccept and PreSMTPDeliver type of plug-ins can be installed through the Netscape Console as described in "Managing SMTP Plug-Ins with Netscape Console" on page 206, or by manually editing the plug-in configuration file as described in "Managing SMTP Plug-Ins Manually" on page 210.
- Protocol-level (PreSMTPAccept) type plug-ins are installed and configured manually as described in "Installing and Configuring Protocol-Level Plug-Ins" on page 212.

Netscape Messaging Server provides an application programming interface (API) that allows you or third parties to create server plug-ins that can add site-specific functionality to Messaging Server. (See your Netscape sales or support representative for information about the availability of plug-ins for Netscape Messaging Server.)

Developers can use the Messaging Server Plug-in API to:

- Process the message header and body at specific times, before the message undergoes further processing by Messaging Server
- Split the message among envelope recipients and change the envelope sender

Message-processing plug-ins can include character-set converters or content filters that implement site-specific firewall functionality, filtering out suspicious incoming or outgoing attachments. Message-splitting plug-ins can be used to customize the message content for a subset of the original recipients.

For more information about server plug-in APIs, see *Messaging Server Plug-In API Guide*. For general Netscape developer information, see the Netscape DevEdge site.

Managing SMTP Plug-Ins with Netscape Console

You can install, configure, activate, and delete PostSMTPAccept and PreSMTPDeliver type plug-ins with the Netscape Console as described in this section. (Working with protocol-level type plug-ins is described in "Installing and Configuring Protocol-Level Plug-Ins" on page 212.)

Installing Plug-Ins

Follow these steps to install a PostSMTPAccept or PreSMTPDeliver type plug-in with Netscape Console:

1. Copy the plug-in library file to a suitable location on your server host machine.
2. In Netscape Console, open the Messaging Server on that host machine.
3. Click the Configuration tab and open the Services folder in the left pane.
4. Open the SMTP icon under the Services folder.
5. Select Plug-ins under the SMTP icon.
6. Click Add in the right pane to open the Add Plug-in window.
7. In the Add Plug-in window, specify the following information for your new plug-in:
    - In the Entry field, use the drop-down list to select the entry point in SMTP processing at which your plug-in operates (`PostSmtpAccept` or `PreSmtpDeliver`).

- In the Path field (required), enter the path name to the plug-in library file.
- In the Functions field (required), enter a comma-separated list of the names of the functions that this plug-in provides. Your plug-in documentation should state exactly what to enter in this field.
- In the Init field (required), enter the name of the initialization function that allows the plug-in to initialize any global or static data before mail reception or delivery begins. This function is called at MTA startup (`smtpd startup`).
- In the Options field (optional), enter any other required information (in the form of space-separated *`name=value`* pairs) as specified in the documentation accompanying the plug-in.

8. Click OK to commit your entries and install the plug-in.
9. Restart the server.

Note: Your plug-in might have additional installation procedures not described here. See the documentation accompanying the plug-in for more information.

By default, a newly installed plug-in is not active. To activate your new plug-in, see "Activating and Deactivating Plug-Ins" on page 208.

The information you enter in the Add Plug-in window is written to your Messaging Server's `plugins.cfg` file. For information on installing plug-ins by directly editing `plugins.cfg`, see "Installing and Configuring Plug-Ins" on page 210.

Deleting (Uninstalling) Plug-Ins

Follow these steps to remove an installed PostSMTPAccept or PreSMTPDeliver type of plug-in from your Messaging Server, using Netscape Console:

1. Open the SMTP Plug-ins tab, as described in "Installing Plug-Ins" on page 206.
2. In the Plug-in Configuration table, select the path name of the plug-in you want to delete.

3. Click Delete.

Note: This uninstallation removes the plug-in library file and makes corresponding modifications to the configuration file `plugins.cfg`. If your plug-in includes other files, you might have additional uninstallation steps to follow. See the documentation accompanying your plug-in for more information.

You can also remove a plug-in by editing and deleting files manually. For instructions, see "Installing and Configuring Plug-Ins" on page 210.

Activating and Deactivating Plug-Ins

All installed PostSMTPAccept and PreSMTPDeliver type plug-ins appear in the Messaging Server SMTP Plug-ins form in Netscape Console. Follow these steps to activate or deactivate one of these plug-ins:

1. Open the SMTP Plug-ins tab, as described in "Installing Plug-Ins" on page 206.
2. In the Plug-in Configuration table, locate the path name of the plug-in you want to activate or deactivate.
3. Check the Status box to the left of the plug-in's path name to activate the plug-in; uncheck the Status box to deactivate the plug-in.
4. Click OK.

Note: If the plug-in you want to activate or deactivate does not appear in the table, it might not be installed or it might have been installed improperly. If it was installed improperly, remove it by following the instructions in "Deleting Plug-Ins" on page 212, and then try again.

The active or inactive status of a plug-in is stored in the `plugins.cfg` file. For information on activating and deactivating plug-ins by directly editing `plugins.cfg`, see "Installing and Configuring Plug-Ins" on page 210.

Configuring Plug-Ins

You initially configure a PostSMTPaccept or PreSMTPdeliver type plug-in when you first install it (see "Installing Plug-Ins" on page 206). You can also use Netscape Console to change the configuration of these kinds of plug-in. (Protocol level plug-ins cannot be configured from Netscape Console.)

Follow these steps to reconfigure an installed plug-in:

1. Open the SMTP Plug-ins tab, as described in "Installing Plug-Ins" on page 206.
2. In the Plug-in Configuration table, select the path name of the plug-in you want to reconfigure.
3. Click Edit. The Edit Plug-in window opens.
4. In the Edit Plug-in window, modify any of the following information as needed:
   - In the Entry field, change the entry point in SMTP processing at which this plug-in operates (`PostSmtpAccept` or `PreSmtpDeliver`).
   - In the Path field, edit the path name to the plug-in library.
   - In the Functions field, edit the comma-separated list of the functions that this plug-in provides.
   - In the Options field, edit any other required information, as specified in the documentation accompanying the plug-in.
5. Click OK to commit your reconfiguration of the plug-in.

The information you enter in the Edit Plug-in window is written to your Messaging Server `plugins.cfg` file. For information on reconfiguring plug-ins by directly editing `plugins.cfg`, see "Installing and Configuring Plug-Ins" on page 210.

Note: Your plug-in might have additional configuration files and procedures not described here. See the documentation accompanying the plug-in for more information.

Managing SMTP Plug-Ins Manually

You are not required to use Netscape Console to manage your PostSMTPAccept and PreSMTPDeliver type SMTP plug-ins. This section describes how to manage them by editing the plug-in configuration file. (See "Installing and Configuring Protocol-Level Plug-Ins" on page 212 for information on working with PreSMTPAccept plug-ins.)

Installing and Configuring Plug-Ins

You can install and configure PostSMTPAccept and PreSMTPDeliver plug-ins by directly editing your Messaging Server `plugins.cfg` file. That file is located at *`instanceDirectory`*`/smtp-bin/plugins/`, where *`instanceDirectory`* is the directory containing the files of the specific instance of the Messaging Server that uses the plug-in.

The `plugins.cfg` file holds basic configuration information for all installed PostSMTPAccept and PreSMTPDeliver type plug-ins. In the configuration file, plug-ins are listed in the order in which you installed them in Messaging Server. When they run, the plug-ins for a given entry point are also executed in that order. If order of execution is important, either make sure you install plug-ins in the order you want, or edit `plugins.cfg` to achieve that order.

Each plug-in's configuration is a line with these elements:

`entry pluginPath` `funcs=`*`functionList`* `init=`*`initfunction`* `[`*`optionList`*`]`

where the elements have the following meanings:

`entry`	The entry point (`PostSmtpAccept` or `PreSmtpDeliver`) at which this plug-in operates.
`pluginPath`	The path name to the plug-in library's binary (executable) file.
`functionList`	A comma-separated list of the functions supported by this plug-in. Messaging Server calls these functions, in the order in which they appear in this list, to execute the plug-in.

initfunction	A function called at MTA startup (`smtpd startup`) which allows the plug-in to initialize any global or static data before mail reception or delivery begins. For example, this can be used to perform preprocessing tasks such as reading `config` files.
optionList	An optional set of space-separated *`name=value`* pairs that the plug-in can use for any purpose. Messaging Server passes this information to the plug-in when it executes the plug-in.

Here is an example for a Unix server with the UBE filter plug-in:

```
PostSmtpAccept \
/export2/server4/msg-Airius1/smtp-bin/plugins/libUBEfilter.so \
funcs=filter_msg_plugin \
init=filter_msg_init \
config=/export2/server4/msg-Airius1/smtp-bin/plugins/UBEfilter.cfg \
option=/export2/server4/msg-Airius1/smtp-bin/plugins/UBEfilter.opt
```

These command lines tell the server to:

- Use the file `libUBEfilter.so` (the library that contains the UBE plug-in)
- Call the function `filter_msg_plugin` to start filter processing
- Call the initialization function `filter_msg_init` at server startup

The command includes two options, named `config` and `option` that specify path names to configuration files:

- Use the filter configuration file named `UBEfilter.cfg`
- Use the filter options file named `UBEfilter.opt`

Here is a similar example for a Windows NT server with the UBE filter plug-in (all of which would be on a single command line):

```
PostSmtpAccept
i:\Netscape\Server4\msg-Airius1\smtp-bin\libUBEfilter.dll
funcs=filter_msg_plugin
init=filter_msg_init
config=i:\Netscape\Server4\msg-Airius1\smtp-bin\plugins\UBEfilter.cfg
option=i:\Netscape\Server4\msg-Airius1\smtp-bin\plugins\UBEfilter.opt
```

This plug-in also acts at `PostSmtpAccept`, its library name is `libUBEFilter.dll`, it has a single function (`filter_msg_plugin`), and it uses two options (named `config` and `option`) that specify path names to configuration files.

You can modify the information in any of these lines to change the configuration of the plug-in. For example, if you change the location of the plug-in library file, you can enter the new path here instead of using the Netscape Console interface. (This would be equivalent to reinstalling the plug-in, and would require a server restart.)

If a plug-in is installed but has been deactivated through Netscape Console, its line in the `plugins.cfg` file is commented out; that is, it starts with a number sign (#) character. You also can manually deactivate a plug-in by commenting out its lines in the file.

Note: If your plug-in uses other files besides its library file and `plugins.cfg`, configuring the plug-in might mean editing those files as well. For details, see the documentation that accompanies your plug-in.

Deleting Plug-Ins

To manually delete, or uninstall, a PostSMTPAccept, or PreSMTPDeliver type plug-in, follow these steps:

1. Deactivate the plug-in if it is active.
2. Delete the plug-in's lines from `plugins.cfg`
3. Remove the plug-in library file from your server host machine.
4. Remove any other plug-in-specific configuration files, as noted in the documentation that accompanies the plug-in.

Installing and Configuring Protocol-Level Plug-Ins

Protocol level (PreSMTPAccept) plug-ins are installed using the `configutil` utility as described below.

There is no standard method of configuring a protocol-level plug-in. Configuration methods and options for protocol-level plug-ins are specified in the plug-in design. Consult the documentation that accompanies the plug-in for configuration information.

To install and activate a protocol-level plug-in:

1. In Unix environments, make sure that your configuration environment is correctly set to the server instance's configuration directory. Typically this is *server-root*/config/smtp. (This step is not necessary in NT environments.)

2. Set your shared library path variable to the directory containing your platform's shared libraries.

   For example, in a typical Unix environment you set the LD_LIBRARY_PATH variable to *server-root*/bin/msg/lib.

3. Use configutil to specify that SMTP is to load the plug-in when it starts up.

   A full path to the plug-in must be specified.

   If the plug-in must be run locally, use the configutil -l option.

   For example, to load a plug-in named antirelay.so stored in the plugins subdirectory in a Unix environment, you enter:

   ```
   configutil -l -o service.smtp.protplugmodules -v \
   '/usr/netscape/suitespot4/plugins/antirelay.so'
   ```

4. Configure the anti-relay plug-in as described in the plug-in's documentation.

Chapter

8

Filtering Unsolicited Bulk Email

This chapter describes the unsolicited bulk email (UBE) plug-in and the filters it uses to help you screen out unsolicited email. It also describes the anti-relay filter and plug-in that you can use to prevent your site from being used to relay UBE to others. For information on Messaging Server plug-ins in general, see Chapter 7, "Working with SMTP Plug-Ins."

This chapter contains the following sections:

- About UBE
- About the UBE Plug-In
- UBE Filter Format
- Managing Filters with Netscape Console
- Creating Filters Manually
- Extending the UBE Plug-In

About UBE

Unsolicited bulk mail (UBE) is email sent to large numbers of recipients without their knowledge or consent, often advertising commercial products or services. It is the electronic equivalent of paper junk mail. For a full definition,

see the Web-based document *Unsolicited Bulk Email: Definitions and Problems* at `http://www.imc.org/uce-def.html`. Another name for UBE is unsolicited commercial email (UCE).

Users of email services are sometimes inundated with unsolicited bulk mail and might expect their service providers to address the problem. One approach that providers can follow is to use filters that selectively eliminate email whose characteristics mark it as UBE. Netscape Messaging Server provides both a plug-in architecture and a scriptable plug-in module that allows you to build filters that prevent most unsolicited bulk email from being delivered to your users. Such filters can be used to cut down on (though probably never completely eliminate) UBE.

UBE filters block UBE messages according to information in the message header and envelope as described in "How UBE Filters Work" on page 217 and "UBE Filter Format" on page 218. For example, a filter might be set up to refuse messages originating from a domain known to send out UBE. Some people who send out UBE try to circumvent these kind of anti-UBE filters by disguising their messages' true origin. They do this by relaying the messages through other systems so that the messages appear to come from somewhere else. We refer to this as *unauthorized-relaying*. See "Anti-Relaying Defenses" on page 244 for information on how to prevent unauthorized-relaying at your site.

About the UBE Plug-In

The UBE plug-in is an SMTP plug-in that works as an extension to Netscape Messaging Server. You can use it to design and implement filters that can block unsolicited bulk email from reaching your users.

UBE Filters and the UBE Plug-In

The UBE plug-in is a post-accept SMTP plug-in to Netscape Messaging Server. SMTP plug-ins are dynamically loaded software modules that access the Messaging Server Plug-in API, an application programming interface. The plug-ins function by intercepting incoming messages, analyzing them for certain

characteristics, and taking appropriate action before possibly passing them on. SMTP Plug-ins are described in general in Chapter 7, "Working with SMTP Plug-Ins."

The UBE plug-in is provided as part of Messaging Server. It is a customizable plug-in that works by examining all incoming mail before it is routed throughout the system (to users' mailboxes or to other messaging servers). The UBE plug-in uses a set of rules (called *filters*), contained in a configuration file, to decide how to handle each piece of mail. Any mail not affected by any of the filters continues on its normal course, untouched by the plug-in.

The UBE plug-in supports a simple scripting language that is useful for processing and directing the flow of messages. You use the language to create the filters that intercept and block or otherwise handle UBE messages. You can use the Netscape Console interface to create and edit filters, or you can perform those tasks manually by editing the UBE configuration file directly.

How UBE Filters Work

A UBE filter is a line in a text file (default name = `UBEfilter.cfg`). When you create a filter, you define a criterion to which email messages are compared. Whenever an email message meets the criterion defined in the filter definition, the filter triggers the action that you designated in the filter definition.

For example, if you don't want to get any messages from `uglymail.com`, you can create a UBE filter that identifies any email message from `uglymail.com`. You can then designate `REJECT` as the action to be performed by the UBE plug-in.

In using the UBE plug-in, you can create many different types of UBE filters and activate or deactivate them individually. You can manipulate your set of UBE filters through the Netscape Console interface, or through direct editing of the configuration file in which they are stored. This chapter describes both methods.

UBE Filter Format

A UBE filter is a single text line containing 3 to 5 parts, in this order:

`[:`*`label`*`] `*`messageField matchCriterion action`*` [`*`argument`*`]`

Here is an example filter that includes all five parts:

```
:DontSend Subject "Bad mail" REJECT "Do not send mail"
```

The following subsections use this example to describe, in order, the parts that make up a filter.

Label

The label is an identifying name for the filter. This part is optional and is used only if this filter is the destination of another filter's `JUMP` action. (For a description of the `JUMP` action, see Table 8.1 on page 221.)

The label is always preceded by a colon (`:`).

Example: `:DontSend`

Message Field

The message field is the portion of the message header or envelope that the filter analyzes. Each arriving message has header fields that hold information about that message. The filter can use any of the fields in deciding how to handle the mail. There is no complete list of possible header fields since the format is open (see Internet Draft RFC 822 for a list of the defined standard fields), but common fields include `To`, `From`, `Sender`, `Reply-To`, `Content-type`, and so on. For descriptions of some of the most common fields, see Table 8.3 on page 226.

Example: `Subject`

After the message field name you can place a colon (`:`) followed by special tags that affect how the message field is processed. Currently, two special tags are supported: `case` and `envonly`.

case Tag

The `case` tag causes the matching criterion (which follows the message field in the filter) to be treated as case sensitive. By default, filters are not case sensitive. Here's an example of two filters, identical except for the `case` tag:

```
Subject:case "Bad mail" REJECT "Do not send mail"

Subject "Bad mail" REJECT "Do not send mail"
```

The first filter would intercept only those messages whose subject is `Bad mail`; the second filter would intercept messages whose subject is either `Bad mail` or `bad mail` (or even `bAd mAiL`).

envonly Tag

The `envonly` tag instructs the filter to consider only envelope information, rather than all message-header information, for the message field. Because header information is more easily altered by a mail sender than is envelope information, you can use this tag to help resist spoofing attacks based on header information.

For example, `Auth-Sender` is an envelope field, added by a server sending a message by authenticated SMTP, that identifies the sender of the message.You might design a filter that examines that field to ensure that only authenticated mail gets through. A tricky user could, however, add a phony `Auth-Sender` field to the header information in a message, possibly defeating your filter. If you restrict the message field to `envonly`, only those fields that are created by the server, and thus are harder to tamper with, will be considered.

Note: Using the `envonly` tag makes sense only if your UBE plug-in has been configured to include header fields by default; for an explanation, see "Envelope Fields and Header Fields" on page 225.

Match Criteria

The match criterion is the string (or regular expression) that the filter looks for in the contents of the message field to see whether a match is achieved. The combination of message field and match criterion is called the filter's *predicate;* the state of the predicate (matched or unmatched) determines whether or not the filter is to be applied.

Predicate example: `Subject "Bad mail"`

In this example, if the Subject field of the message being analyzed is `Bad mail`, the filter is applied; otherwise, the filter is not applied. Note that, in the absence of a `case` tag, the match criterion is not case sensitive.

The match criterion is a regular expression, so you can use it to match a wide variety of strings. For example, to match any Subject field containing the phrase "Get Rich Quick", you can specify `".*Get Rich Quick.*"` as the match criterion. For more information, see "Regular Expressions for Match Criterion" on page 223.

Action

The action field defines the action that the UBE plug-in performs if a match occurs. As soon as a match occurs with any filter predicate, the action associated with that filter is taken. Filter processing stops when a *terminal action* (one that stops processing) is taken, or when the end of the filter file is reached. For descriptions of the actions, see Table 8.1 on page 221.

Example: `REJECT`

This action causes the message to be returned to the sender.

You can also apply the negation modifier to reverse the conditions under which an action occurs; see "Negation Modifier" on page 230.

Argument

The argument field contains any additional information that may be needed to perform the specified action. Some actions do not take an argument, whereas others do. For descriptions of the arguments used by each of the actions, see Table 8.1 on page 221.

Example: `"Do not send mail"`

In this case, `"Do not send mail"` is the argument to the `REJECT` action; it is a message that accompanies the returned mail.

Available Actions for UBE Filters

Table 8.1 lists the actions that you can specify in a UBE filter. Each action is classified either as terminal (filter processing stops as soon as it is taken) or non-terminal (processing continues after it is taken).

Table 8.1 UBE filter actions

Action	Argument	Description
`COPY`	Comma-separated list of addresses	Adds these recipients to the original set of recipients, then continues processing the next filter. (Non-terminal action.)
	Example: `Channel-To "user1@domain1\.com" COPY "postmaster, user2"` In this case, any mail sent to `user1@domain1.com` is also sent to `postmaster` and to `user2`.	
`DROP`	One address	Replaces original set of recipients with this one and sends the message on. (Terminal action.)
	Example: `User-From ".*@bulk\.com" DROP "postmaster"` In this case, any mail from any account at `bulk.com` is sent to `postmaster` but no one else.	
`EXIT`	(none)	Immediately stops filter processing and sends the message on. Subsequent filters in the configuration file are not applied to this message. (Terminal action.)
	Example: `User-From "CEO@.*" EXIT` In this case, any mail from a user named `CEO` is sent through.	

Table 8.1 UBE filter actions (Continued)

Action	Argument	Description
HOLDCOPY	Comma-separated list of addresses, followed by a vertical bar (|) and a message string	Holds the message. (Terminal action.)
	Example: `Subject "Free stuff!" HOLDCOPY "postmaster | please handle"` In this case, any mail containing the subject `"Free stuff!"` is held, with a copy sent to `postmaster`.	
HOLDONLY	Comma-separated list of addresses, followed by vertical bar (|) and a message string	Holds the message. (Terminal action.)
	Example: `Subject "Free stuff!" HOLDCOPY "postmaster | please handle"` In this case, any mail containing the subject `"Free stuff!"` is held, with a notification sent to `postmaster`.	
JUMP	Label (filter name)	Shifts filter processing to the named filter, skipping any intervening filters. The named filter must exist in the configuration file. (Non-terminal action.)

	Example:
	`Subject "Easy $$$" JUMP "MoneyReject"`
	`subject".*$$$.*" HOLDCOPY "postmaster | evaluate for $$$"`
	`:MoneyReject Subject "Easy $$$" REJECT "No commercials, please"`
	In this case, mail with the specific subject `"Easy $$$"` is rejected, whereas other mail with triple dollar signs in the subject is held for the postmaster to evaluate.	

Table 8.1 UBE filter actions (Continued)

Action	Argument	Description
REJECT	Reject reason (message string)	Returns the mail to the sender and includes the reject reason in the message. (Terminal action.)
	Example: `User-from "Pitchman@cheapstuff\.com" REJECT "Do not advertise to our users"` In this case, any mail from `Pitchman@cheapstuff.com` is automatically returned along with the message `"Do not advertise to our users"`.	
RUN	Command line to execute	Executes the specified program, passing the message header and body to the program. The program must be located in the directory `INSTANCEDIR/smtp-bin/plugins` where `INSTANCEDIR` is the instance directory (name = `msg-`*`instancename`*). You can use the special field name $& in the subsequent filter to match the return value of the program. (Non-terminal action.)
	Example: `Subject "May contain a virus" RUN "VirusScan.exe" $& "1" REJECT "Message rejected due to presence of virus!"` In this case, any mail with the subject `"May contain a virus"` is sent to the program `VirusScan.exe` for analysis. If the return value from the program is 1, the message is returned along with the message `"Message rejected due to presence of virus!"`.	

Regular Expressions for Match Criterion

The UBE filter supports extended regular expressions compliant with POSIX 1003.2. There are many sources of information on regular expressions; this document is not intended as a reference.

Table 8.2 lists some of the common regular-expression special characters and constructions that you can use in creating match criteria for UBE filters.

Table 8.2 Regular-expression special characters

Characters	Usage
`.`	(Dot). Matches any single character.
`[]`	(Brackets). Matches any single character from the set of characters specified within the brackets. The brackets may enclose the entire set of permissible characters (`[2468]`), or they may specify an ASCII range, using end points and a hyphen (`[a-z]`). For example, `r[eo]d` matches `red` and `rod`, but not `rid` or `reed`. `x[0-9]` matches `x0`, `x1`, `x2`, and so on, but not `x11`.
`[^]`	(Caret in brackets). Matches any single character that is *not* one of those specified within the brackets following the caret. (The caret must be the first character in the brackets.) For example, `r[^eo]d` matches `rid` but not `red` or `rod`. `x[^0-9]` matches `xa`, `xb`, `xc`, and so on, but not `x2`.
*	(Asterisk). Matches zero or more of the immediately preceding character or expression. For example, `ba*c` matches `bc`, `bac`, `baac`, and so on. The expression `.*` matches any string.
+	(Plus) Matches one or more of the immediately preceding character or expression. For example, `ba+c` matches `bac`, `baac`, and so on. `r[eo]+d` matches `red`, `rod`, `reed` and `rood`, but not `reod` or `roed`.
\	(Backslash) The Escape character. It removes the pattern-matching significance of the character it precedes, so that special characters (like those in this table) can be matched as regular ASCII characters. For example, `.` matches any character, but `\.` matches only the dot character. (To match the backslash character itself, you must precede it with another backslash: `\\`.)
\~	(Escaped tilde) Matches any character but the subsequent character. For example, `b\~ad` matches `bbd`, `bcd`, `b3d`, but not `bad`.

Table 8.2 Regular-expression special characters (Continued)

Characters	Usage
`\{\}`	(Escaped braces) Matches any number of occurrences of the exact sequence of characters between the escaped braces. For example, `\{ju\}+fruit` matches `jufruit`, `jujufruit`, `jujujufruit`, and so on. It does not match `jfruit`, `ufruit`, or `ujfruit`. You can use the expression `\{ \}` to match any number of spaces between words.
`\{`*x*`\!`*x*`\}`	(Escaped bang in escaped braces) Matches any one of the characters *x* separated by the alternation symbol (`\!`). For example, `\{j\!u\}+fruit` matches `jfruit`, `jjfruit`, `ufruit`, `ujfruit`, `uufruit`, `uuufruit`, and so on.
^	(Caret) Matches the beginning of a line.
$	(Dollar sign) Matches the end of a line.
`()`	(Parentheses) Delimits arguments. You can explicitly limit the extent of a regular expression by enclosing it in parentheses. For example, `uuufruit` is a single regular expression, but you can force it to be considered two separate expressions by applying parentheses, as in `(uuu)(fruit)`. For an example of the use of parentheses, see the example filter following Table 8.4 on page 228.

Many other rules define, for example, what a character is and how characters in regular expressions may be juxtaposed for various purposes. For more information, please consult any POSIX 1003.2-compliant regular-expression documentation.

Envelope Fields and Header Fields

Every message has two types of fields that you can include in the message-field part of your filters. Header fields are created by a mail client when it sends a message. Envelope fields are created by mail servers that send or resend the message during its transit from sender to receiver. By default, the UBE plug-in examines envelope fields only. This restriction exists for performance reasons, because the header can contain any number of fields.

The envelope fields most useful for filter purposes include those listed in Table 8.3.

Table 8.3 Common envelope fields for UBE filters

Envelope field	Description
`Submitted-Date`	The date on which the server received the message.
`Host-From`	The IP address of the machine that directly connected to and transferred the message to this Messaging Server. You can use this field to, for example, separate external mail from mail of local origin.
`User-From`	The value of the SMTP `mail-from` command. The value in this field is not validated unless you have enabled the "Verify each recipient's address when accepting messages" setting. (For instructions, see "Verifying Recipient Addresses" on page 97.) You can use this field to, for example, ensure that you accept (or perhaps reject) messages only from specific known users.
`Auth-Sender`	If authenticated SMTP is being used for this message, this field contains the email address of the authenticated user that sent the message. (Note that this field contains the email address of the user, not the user name or password used to authenticate the user.) You can use this field to, for example, accept mail only from authenticated users.
`MAIL-Exts`	A list of any extensions to SMTP (such as delivery notification) that were passed to the SMTP daemon through the `mail-from` SMTP command. Extensions are listed exactly as entered in the command.
`Channel-To`	The list of recipients for this message, as listed in the `rcpt-to` SMTP command. Each recipient is listed on a separate line, with this format: *`user@xyzcorp.com`*
`RCPT-Exts`	A list of any extensions to SMTP (such as delivery notification) that were passed to the SMTP daemon through the `rcpt-to` SMTP command. Extensions are listed exactly as entered in the command.

Table 8.3 Common envelope fields for UBE filters (Continued)

Envelope field	Description
`Message-Size`	The total size, in bytes, of the header plus body of the received message. You can use the value in this field to, for example, drop messages that are too large.
`MTA-Hops`	The number of SMTP servers (not including this one) that the received message has passed through. You can use the value in this field to, for example, drop messages that have been relayed through too many machines.

If you want to include header fields as well as envelope fields in your filter predicates, you can use the Netscape Console interface. You can also accomplish this manually, by adding the following line to your filter options file (`UBEfilter.opt`):

```
parseheader: 1
```

This instruction tells the UBE plug-in to look at both envelope and header fields when comparing a message to a filter.

If you have set the default to include header fields, you can then, on a filter-by-filter basis, force the plug-in to look only at the envelope fields by using the `envonly` flag in the message field of your filter; for a description of this flag, see "envonly Tag" on page 219.

Note: If you are designing your filters to operate on header fields as well as envelope fields, you can use your mail client to help decide which header fields to analyze. With the Netscape Messenger mail client in Netscape Communicator, you can view the complete set of header fields in a received message by choosing All from the Headers menu (accessed from the View menu).

Special Message-Field Names

You can use special field names to combine filters and narrow your criteria. Table 8.4 describes these names, using the following filter as an example:

```
Subject "This is ."   REJECT "This is bad mail"
```

(This example includes a dot wildcard, which matches any single character, in the match criterion.)

Table 8.4 Special message-field names for UBE filters

Special name	Explanation
`$0`	Represents the complete message field content that was last matched. Assume that the message just compared against the above example filter had the subject "This is a test". Then the current value of `$0` would be `"This is a test"`.
`$1`	Represents the exact matching portion of the message field content that was last compared. In the above example, again assuming that the message subject was "This is a test", the value of `$1` after the comparison would be `"This is a"`. An example of the use of `$1` follows this table.
`$2..$9`	Represent the values, in order, of any parenthetical matches you may have specified in your match criterion. If, in the above example, your match criterion had been `"(This) (is) (a) (test)"`, and again assuming that the message subject was `"This is a test"`, the comparison would yield the results `$2=this`, `$3=is`, `$4=a`, and `$5=test`.
`$&`	Represents the numerical result code of the last action. You can use this special field name in conjunction with the `RUN` action (see Table 8.1 on page 221).

Table 8.4 Special message-field names for UBE filters (Continued)

Special name	Explanation
$#	Represents the number of recipients in the current message. A comparison of this value and a number is considered true if this value is equal to or greater than the given number. You can use this field name to, for example, block all messages with more than a maximum permitted number of recipients.

$ANY	Represents any field. Using this special name causes the plug-in to compare your match criterion with all fields. For example, you can use it to search all header fields for a specific phrase, as in this predicate:
	
	`$ANY "get free stuff"`
	
	You can also use it to match all messages, with a predicate such as
	
	`$ANY ".*"`
	
	which will match any field containing any string.

The following example uses both a special field name and parenthesized expression parsing:

```
:handleFrom   User-From   (.*)@airius.com   !JUMP   handleregular
              $1          "postmaster"       JUMP   handleAIRIUSpost
              ""          ""                 JUMP   handleregular
```

- The first line (labeled `:handleFrom`) looks at the senders of the message, using parentheses (as described in Table 8.2 on page 224) to match against the user name. Only if the mail is from someone at Airius corporation does execution continue to the next line. (See the next section for an explanation of the exclamation point modifier.)
- The second line states that, if the message is exactly from the `postmaster` account at Airius, execution should jump to the filter line that handles Airius postmaster messages.

- The third line causes processing of other messages to jump to another location. The message field and match criterion are not needed for this filter, so they are replaced by empty quotes; for an explanation, see "Omitting Parts of a Filter" on page 237.

Negation Modifier

You can modify an action by applying the negation operator to it. Normally, the action specified in a filter is taken if the match criterion is matched. If, however, the negation operator is applied to the action, the action is *not* taken if the criterion is matched; conversely, the action *is* taken if the criterion is *not* matched.

For example, the following filter accepts all messages originating from within Airius Corporation:

```
Sender ".*airius\.com" EXIT
```

A complementary filter might reject all messages that are not local:

```
Sender ".*airius\.com" !REJECT "local mail only"
```

If you are using the Netscape Console interface to create filters, you click a button to apply the negation modifier; if you are creating filters manually, you apply the modifier as a prefix to the action, as shown in the example.

Managing Filters with Netscape Console

The UBE plug-in is installed automatically when you install Messaging Server; for more information, see the *Messaging Server Installation Guide*. This section explains how to use Netscape Console to activate the UBE plugin and to create, edit, activate, and change the order of individual filters.

Details of filter format and instructions for designing filters using Netscape Console are given in "UBE Filter Format" on page 218. Note that you can also create and manipulate filters manually; for details, see "Creating Filters Manually" on page 235.

Activating the UBE Plug-In

To make the UBE plug-in available for use, use Netscape Console to access the SMTP Plugins form (see "Activating and Deactivating Plug-Ins" on page 208). Turn on the UBE plug-in in the Plugin Configuration table. (Its name is `libUBEfilter.so` or `libUBEfilter.sl` for Unix; `libUBEfilter.dll` for Windows NT.)

As with all SMTP plug-ins, basic configuration of the UBE plug-in is controlled by the contents of the configuration file `plugins.cfg`. Individual filter characteristics and other aspects of UBE plug-in execution are controlled by the files `UBEfilter.cfg` and `UBEfilter.opt`. For more information, see "Plug-In File and Configuration Files" on page 236.

Creating a New Filter

Follow these steps to create a new UBE filter from Netscape Console:

1. In Netscape Console, open the Messaging Server whose UBE plug-in you want to add a new filter to.
2. Follow either of these steps to access the UBE Configuration form:
   - Click the Tasks tab, then click "Configure Unsolicited Bulk Email Filters".
   - Click the Configuration tab, open the Services folder in the left pane, and open the SMTP icon beneath the Services folder. Open the Plugins icon beneath the SMTP folder icon, and select UBE beneath the Plugins icon. Click the Unsolicited Bulk Email tab in the right pane.
3. Click "Add a filter." The "Add a UBE Filter" window opens.
4. (Optional) Assign a label to the new filter. The label is needed only in certain circumstances; see "Label" on page 218.

5. Specify a message field to use for matching. Either select a field from the drop-down list in the "If" field or enter the field name directly into the "If" field. The field names in the drop-down list correspond to the parts of the header or envelope information attached to an email message. See "Message Field" on page 218.

6. Choose equals (=) or does not equal (!=). If you choose =, the UBE plug-in acts only when a match between a message and this filter occurs. If you choose !=, the plug-in acts only when a match between a message and this filter does *not* occur. See "Negation Modifier" on page 230.

7. Type a value (the match criterion) in the Value field. This is the value (generally a string or regular expression) that the UBE plug-in compares with the contents of the message field you specified above. See "Match Criteria" on page 219.

8. Select an action from the "Then" field. This is the action the UBE plug-in will perform on any email that matches this filter (or does not match, if you chose (!=) in Step 4). See "Action" on page 220.

9. If required by the action you selected, enter an argument for that action in the Argument field. Some actions require arguments, such as addresses to forward mail to. See "Argument" on page 220.

10. When you have finished creating the UBE filter, click OK. The new UBE filter appears in the list of UBE filters in the UBE Configuration form.

11. Back in the UBE Configuration form, click Save to save the new UBE filter.

    Note: Newly created filters are not saved until you click Save, even though they may appear in the UBE Configuration form.

New filters are automatically saved as active. If you want to deactivate your new filter, follow the instructions in "Activating and Deactivating Filters" on page 234.

You can also create a new UBE filter by directly editing the filter configuration file. See "Creating Filters Manually" on page 235.

Editing an Existing Filter

Follow these steps to edit an existing UBE filter from Netscape Console:

1. Access the UBE Configuration tab, as described in “Creating a New Filter” on page 231.
2. In the Filters field, click to highlight the UBE filter that you want to edit.
3. Click the Edit button. The “Edit a UBE Filter” window opens, displaying the parts of the UBE filter you’re editing.
4. Make any desired changes to the filter, specifying the contents of the fields, as described in “Creating a New Filter” on page 231.
5. When you have finished editing the filter, click OK. The revised filter appears in the same place it previously occupied in the UBE configuration form.

   Caution: When you edit a UBE filter, the edited filter replaces the original one. You cannot create a new filter by editing an existing filter and, for example, saving it with a new name (label). To create a new filter, see “Creating a New Filter” on page 231.
6. Back in the UBE Configuration form, click Save to save the edited filter.

   Note: Changes made to filters are not saved until you click Save, even though they may appear changed in the UBE Configuration form.

You can also modify an existing UBE filter by directly editing the filter configuration file. See “Creating Filters Manually” on page 235.

Activating and Deactivating Filters

Active filters are designated with a check in the Active box next to the filter definition in the UBE Configuration tab. You can activate or deactivate filters individually, and you can activate or deactivate all filters at once. Follow these steps:

1. Access the UBE Configuration tab, as described in "Creating a New Filter" on page 231.
2. Do any of the following:
   - To activate all filters, click "Activate all filters."
   - To deactivate all filters, click "Deactivate all filters."
   - To activate a single inactive filter, check the Active box next to it.
   - To deactivate a single active filter, uncheck the Active box next to it.
3. When you have finished activating or deactivating filters, click Save.

   Note: The changes you make to the active state of filters are not saved until you click Save.

You can also activate and deactivate UBE filters by directly editing the filter configuration file. See "Creating Filters Manually" on page 235.

Changing the Order of Filters

Incoming email messages are compared to UBE filters in the order in which the filters appear in the UBE Configuration tab. Depending on how you design your filters, the order in which they are applied to incoming email messages may be important. Follow these steps to change the order of the filters in the UBE Configuration tab:

1. Access the UBE Configuration tab, as described in "Creating a New Filter" on page 231.
2. Select the filter you want to move.

3. Click the Move up or Move down button to move the selected filter up or down. Repeat as necessary.

4. When you have finished changing the order of your UBE filters, click Save.

   Note: The changes you make in the order of your UBE filters are not saved until you click Save, even though they may appear reordered in the UBE Configuration tab.

You can also reorder UBE filters by directly editing the filter configuration file. See "Creating Filters Manually" on page 235.

Parsing Header Fields

You can specify that the UBE plug-in examine header fields as well as envelope fields when applying the filters to a message. (By default, the plug-in looks only at envelope fields.) Follow these steps:

1. Access the UBE Configuration tab, as described in "Creating a New Filter" on page 231.

2. Check the "Parse message header" box to specify that the UBE plug-in should parse message headers as well as envelopes. For more information, see "Envelope Fields and Header Fields" on page 225.

You can also set this option manually. For more information, see "Plug-In File and Configuration Files" on page 236.

Creating Filters Manually

If you are designing a lengthy and complex set of UBE filters, it may be more efficient to create and edit them manually, rather than through the Netscape Console interface. This section describes how to create, delete, modify, activate, and deactivate UBE filters without using Netscape Console. To do this, you must understand the configuration files used by the UBE plug-in and how to edit them.

Plug-In File and Configuration Files

The UBE plug-in is an SMTP plug-in that operates at the `PostSmtpAccept` point of message handling. Messaging Server uses the file `plugins.cfg` to configure the UBE plug-in. A typical UBE configuration in `plugins.cfg` (for a Unix Solaris installation) might look like this (for more information about `plugins.cfg`, see Chapter 7, "Working with SMTP Plug-Ins"):

```
PostSmtpAccept \
/export2/server4/msg-Airius1/smtp-bin/plugins/libUBEfilter.so \
funcs=filter_msg_plugin \
init=filter_msg_init \
config=/export2/server4/msg-Airius1/smtp-bin/plugins/UBEfilter.cfg \
option=/export2/sever4/msg-Airius1/smtp-bin/plugins/UBEfilter.opt
```

These command lines tells the server to:

- Use the file `libUBEfilter.so` (the library that contains the UBE plug-in)
- Call the initialization function `filter_msg_init` at server startup
- Call the function `filter_msg_plugin` to start filter processing
- Use the filter configuration file named `UBEfilter.cfg`
- Use the filter options file named `UBEfilter.opt`

Note: If a plug-in is installed but has been deactivated through the Netscape Console interface, its line in the `plugins.cfg` file is commented out; that is, it starts with a number sign (#) character.

The most important files to note from this `plugins.cfg` example are the filter configuration file and the filter options file. These two files drive the functioning of the UBE plug-in.

- The **filter configuration file** (`UBEfilter.cfg` by default) contains the list of filters to apply to any incoming mail. When you create, edit, delete, activate, or deactivate filters, whether manually or through the Netscape Console interface, you modify this file.

- The **filter options file** (`UBEfilter.opt` by default) controls whether to parse the message header file. You modify this option by editing the file directly or by using the Netscape Console interface (see "Parsing Header Fields" on page 235).

Editing the Filter Configuration File

To create and manipulate UBE filters without using Netscape Console, you need to manipulating several configuration files.

1. Start by ensuring that the UBE plug-in is activated.

   Examine the file `plugins.cfg` (see previous section) to make sure the proper configuration lines are in place to activate the plug-in.

2. If necessary, edit the file `UBEfilter.opt` or its equivalent to specify message-header parsing.

3. Create your set of UBE filters by editing the file `UBEfilter.cfg` or its equivalent. You can use any text editor (Notepad for Windows NT or vi for Unix, for example).

   Enter your filters in order, one per line, in the configuration file. To enter a comment use the # symbol at the beginning of the line. (For more information on comments, see "Entering Comments" on page 238.)

4. When you are satisfied with your edits, save the file in its proper location as defined in `plugins.cfg`.

5. To make changes later, open, edit, and resave the configuration file.

During execution, the UBE plug-in processes your filters, in order, from the top to the bottom of the file (except when it encounters a `JUMP` action; see Table 8.1 on page 221) and applies all rules that match, until it reaches a terminal action or the end of the file.

Note: When you change the configuration file, its new filters automatically take effect immediately.

Omitting Parts of a Filter

All parts of each filter line (other than the optional label) must be present, but you can use empty quotes (`""`) as a placeholder, if you wish to.

For example, you can replace the action part of a filter if you want filter processing to continue on the next line, rather than taking action immediately:

```
          Channel-To  (.*)@airius.com  ""
 :ceo     $1           ceo             COPY    postmaster@airius.com
 :cfo     $1           cfo             COPY    finance@airius.com
```

In this case, all mail for Airius Corporation is analyzed by subsequent lines to see whether it should be copied to the postmaster or the finance administrator.

For the message field or match-criterion parts of a filter, empty quotes have the effect of matching everything:

```
:cfo     $1           cfo           COPY    finance@airius.com
         ""           ""            JUMP     othermail
```

In this case, once a message has been analyzed to see whether a copy should go to finance, its processing continues with a jump to another location.

Entering Comments

Any line that starts with the number sign (#) or tilde (~) (leading spaces or tabs are ignored) is considered a comment and is ignored during execution by the UBE plug-in.

- You use the number sign (#) to mark comments if you edit the filter file manually. Lines starting with a number sign do not show up in the UBE Configuration tab and are not processed during execution.
- The UBE configuration tab uses the tilde (~) as a special comment marker. If you disable a filter in the tab, the software adds a leading tilde to that filter line in the configuration file. If you enable a previously disabled filter, the software removes the tilde from the line. Both disabled and enabled filters appear in the UBE Configuration tab (with their status marked appropriately), but during execution lines starting with a tilde are considered comments and are ignored.

Keep these points in mind when using comment markers:

- Use only the number sign to create comments. The tilde is reserved for use by the UBE Configuration tab only.
- Do not use the comment symbol anywhere but at the beginning of a line. Any other location is invalid syntax.

You can include either the number sign or tilde as a regular character in any part of your filter by enclosing the character in quotes. For example, if you want to filter based on a message field named `X-Accept#`, your filter line could be:

```
:Label "X-Accept#" "Free stuff" REJECT "Please don't
send this mail"
```

The following example, however, has invalid syntax:

```
:Label X-Accept# "Free stuff" REJECT "Please don't send
this mail"
```

Examples

Filter examples are shown throughout this section as illustrations of filter format. This section provides additional illustrations, including an example of a complexly interacting set of filters that performs many functions.

This sample filter configuration file for the fictional XYZ Corporation illustrates filter usage and the interactions among filters. Following the sample are several scenarios illustrating how the UBE plug-in would use this file to handle several kinds of mail.

```
        Channel-To   "louisr@xyzcorp\.com" COPY     "watch@domain.com"
        Subject      "weapons for sale"    DROP     "weap@xxx.gov"
        Channel-To   "CEO.*"               JUMP     "DoCEO"
:xCEO   $#           "50"                  REJECT   "No bulk mail"
        Subject      "May contain a virus" RUN      "VirusScan.exe"
        $&           "1"                   REJECT   "This had a virus"
        Content-Type "multipart/mixed"     JUMP     "DoMime"
        Client       "Netscape.*"         !JUMP     "TstCli"
:xCli   Subject      ".*"                  EXIT
:DoCEO  Subject      "Postmaster Eval"     HOLDCOPY "postmaster eval"
        $ANY         ".*"                  JUMP     "xCEO"
:DoMime Channel-To   ".*_.*@xyzcorp\.com"  REJECT   "Can't read MIME "
        $ANY         ".*"                  EXIT
:TstCli Host-From    ".*\.xyzcorp\.com"    COPY     "IS_department"
        $ANY         ".*"                  JUMP     "xCli"
```

Scenario 1: Message to the CEO for the postmaster to evaluate

These are the received message's `Channel-to` and `Subject` fields:

```
Channel-To: CEO@domain.com
Subject "Postmaster Eval"
```

1. This message is matched by the `Channel-To "CEO.*"` filter (line 3).
2. Processing therefore jumps to the filter labeled `:DoCEO` (line 10). The filter on that line again matches the message (because the subject is `"Postmaster Eval"`) and so this message is copied to the postmaster, who can then decide whether or not the CEO should see it.

Scenario 2: Message to the CEO about the shareholders' meeting

These are the received message's `Channel-to` and `Subject` fields:

```
Channel-To: CEO@domain.com
Subject "Shareholders meeting"
```

1. As in the previous case, this message is matched by the `Channel-To "CEO.*"` filter (line 3), so processing then jumps to the filter labeled `:DoCEO` (line 10).
2. But this time the filter on line 10 doesn't match the message (the subject is not `"Postmaster Eval"`), so processing passes to the next line (11).
3. The filter on line 11 matches the message (because its predicate can match any string in any field), so processing jumps back to the label `:xCEO` (line 4).
4. At this point the rest of the filters can be applied to this message. If none of lines 4 through 8 match the message, processing will halt with the `EXIT` action at line 9, because the `Subject` fields will match. The mail will then be forwarded to its addressee.

Scenario 3: Message sent to a monitored account:

The account of Louis R. is being monitored for illegal activity. These are the received message's `Channel-to` and `Subject` fields:

```
Channel-To "louisr@xyzcorp.com"
Subject "illegal stock trade"
```

This message is matched by the first filter. A copy of this message is sent to the `watch` account.

Scenario 4: A message arrives addressed to all 3000 employees:

This message is matched by the filter on line 4. That filter automatically rejects any message with 50 or more recipients, regardless of the content of any of its header fields.

Scenario 5: IS requires that all local mail be sent using a Netscape client:

1. If the `Client` field of the message doesn't start with "Netscape", the filter on line 8 transfers processing to the filter labeled `:TstCli` (line 14).

2. That filter tests whether the message was sent from any host in the domain `xyzcorp.com`. If it was, the sender did not use a Netscape client to send the mail, so the plug-in notifies the IS department of that fact by sending them a copy of the message.

3. If the message was not sent locally, processing passes to the last line (15), where it jumps back up to line 9 and continues.

Scenario 6: The account nomime can't read MIME messages:

At XYZ Corporation, mail accounts with underscores are reserved for clients that cannot handle MIME attachments. These are the received message's `Channel-to` and `Content-Type` fields:

```
Channel-To r_francisco
Content-Type "multipart/mixed"
```

1. The filter on line 7 matches the content type of the message, so processing jumps to the filter labeled `:DoMime` (line 12).

2. That filter checks the account name. Because the name contains an underscore, the plug-in rejects the mail.

Extending the UBE Plug-In

You can extend the UBE plug-in to give it capabilities beyond those delivered with Messaging Server. Examples in this chapter have already demonstrated one extension method: the use of programs executed by the RUN action. Another method of extension involves use of an extension library. This section gives brief overviews of how to implement both methods.

Using the RUN Action

The RUN action is built into the UBE plug-in. You can use this action to invoke an external program that can process messages in conjunction with the UBE filters. External programs called in this way can perform tasks such as scanning for viruses, matching message-field content against DNS or other databases of names, matching message-body text, and gathering statistics.

Your external program is executed through a command line that is the argument of the RUN action in a filter. The program provides a return value that a subsequent filter in the configuration file can make use of.

Your program is passed two parameters: the path name to a file containing the envelope of the message that triggered the RUN action, and the path name to a file containing the combined header and body of the message. Your program must return a numerical value.

The RUN action is described in "Available Actions for UBE Filters" on page 221.

Implementing a program executed through the RUN action can be very simple. In many cases, you may be able to use an existing text-manipulation utility, either directly or by writing a simple wrapper that is executed by the RUN action and that in turn calls the text utility.

Note: For security purposes, the external program must be located in the same directory of your Messaging Server as the UBE plugin configuration file (*instanceDirectory*/smtp-bin/plugins/).

Using an Extension Library

The UBE plug-in supports use of an extension library that is separate from the plug-in itself. You can write extensions to verify host name, reject relaying, perform DNS lookup, and perform virus checking, and any other task required. An extension library may be more appropriate than a program executed by the `RUN` action in situations where you want to alter the fundamental nature of a UBE plug-in action, or where you need to add a new kind of action to it.

In Unix environments, the extension library is defined as a shared object or shared library (extension `.so` or `.sl`), and in an NT environment it is referred to as a dynamically linked library (extension `.dll`). In the filter options file you can use the `extension_so` option to define the path to your shared library. For example, the line:

```
extension_so:/lib/HostNameChecker.so
```

instructs the plug-in to load the library `HostNameChecker.so` into memory.

Basically, an extension library can override existing UBE plug-in actions (such as `COPY` or `EJECT`), and it can also add new actions. If an extension library is present at the time the UBE plug-in reads an action in a filter configuration file, the plug-in takes these steps:

1. It tries to locate the action name in the extension library. If an entry point with that action name is found, the plug-in calls it.
2. If the plug-in finds no entry point with that action name in the extension library, the plug-in executes its own built-in default action.
3. If the plug-in has no built in default action of that name, it does nothing and continues processing the filter configuration file.

Each entry point in the extension library is a function that must use the following prototype:

```
int (*ExtpFuncAct) (
      const char *arg,
      char *control_file,
      char *msg_file,
      int * result
     );
```

where *ExtpFuncAct*, the name of the entry point, is the name of the action implemented by this entry point. (That is, COPY, DROP, EXIT, HOLDCOPY, HOLDONLY, JUMP, REJECT, RUN, or a new action defined by this library.)

The parameters to the function have the following meanings:

Table 8.5 Function parameters

Parameter	Description
arg	(input) A pointer to a string that is the argument to the filter action. (The pointer is null if this action takes no argument.)
control_file	(input) A pointer to the path name of a file containing the message's envelope fields.
msg_file	(input) A pointer to the path name of a file containing the message's header fields and body text.
result	(output) A pointer to a result code. A return value of 0 indicates that the UBE plug-in should continue processing the filter configuration file; a nonzero return value stops processing filters. This return value is stored in the variable $&, which can appear as the message-field name in the subsequent filter in the file. For more information, see "Special Message-Field Names" on page 227.

Note: When you write an extension library for Unix, be sure to put `extern "C" {}` around your function declaration. (The function name must be C symbols, not C++ symbols. You can verify that fact on most Unix platforms by displaying the library's symbol table with `nm` *`shared_obj`*`.so`, and making sure it's of type `T`. or in NT environments with `dumpbin /exports` *`dllname`*`.dll`.

Anti-Relaying Defenses

UBE filters block UBE messages according to information in the message header. For example, a filter might be set up to refuse messages originating from a domain known to send out UBE. Some people who send out UBE try to circumvent these kind of anti-UBE filters by disguising their messages' true

origin. They do this by relaying the messages through other systems so that the messages appear to come from somewhere else. This is called *unauthorized-relaying* (or sometimes *third-party relaying*).

Ordinary relaying is when one messaging server passes a message to another messaging server for eventual delivery to a user. In the normal course of operation, messages are often relayed from one server to another. Enterprises with multiple mail servers use relaying to route messages to their destinations, and mail may be relayed between servers in different domains or enterprises.

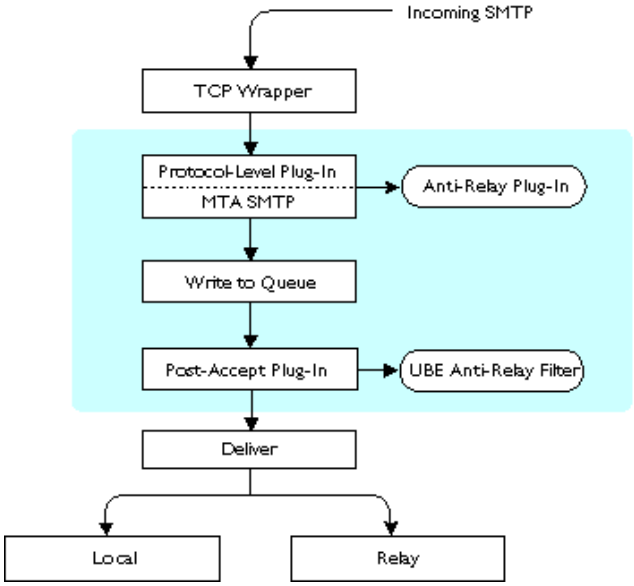
Unauthorized-relaying, however, is used to circumvent anti-UBE filters by disguising the true origin of a message. For example, suppose there is a site named `spam.com` that is known to send out UBE. In defense, site administrators set up UBE filters to bar all messages sent from `spam.com`. But if `spam.com` relays their UBE through your site, those messages now appear to come from your site and the `spam.com` filter no longer blocks them.

Netscape offers two ways to prevent your site from being used to relay unauthorized UBE:

- A UBE plug-in filter can be created to prevent others from relaying their messages through your site to other sites or to your own users. See "Creating an Anti-Relay Filter" on page 247 for details.
- A ready-to-run, protocol-level, anti-relay plug-in that you can use to prevent others from relaying their messages through your site to other sites. See "Using the Anti-Relay Plug-In" on page 248 for details.

The primary difference between the anti-relay plug-in filter and the anti-relay plug-in is that the plug-in is faster and more efficient because it rejects the UBE messages before they are accepted and written to disk. This is illustrated in Figure 8.1 .

Figure 8.1 Anti-Relay filter and Plug-In



Note: Neither the anti-relay filter, nor the anti-relay plug-in, prevent delivery of UBE that is addressed to your users. Only UBE filters as described in "About the UBE Plug-In" on page 216 can prevent UBE from being delivered to your users.

While you can use both the anti-relay plug-in and anti-relay UBE filter at the same time, doing so is redundant. The relative advantages and disadvantages between the plug-in and the filter methods are shown in Table 8.6.

Table 8.6 Advantages of UBE filter and anti-relay plug-in

Method	Advantages	Disadvantages
UBE filter method	• Easy to port existing 3.x filters • Can be administered through Netscape Console or command line • Supported by earlier versions of Messaging Server • You can specify the contents of the bounce message sent back to the UBE sender.	• Slower • Incoming messages are written to disk before being blocked • Fewer features and less easily customized
Plug-in method	• Faster performance • Messages are blocked before they are written to disk • Allows direct feedback to submitting client • More features and greater range of customization • Source code is included	• Cannot be administered through Netscape Console • Only supported by Messaging Server 4.x • Only an SMTP error code is returned to the UBE sender

Creating an Anti-Relay Filter

UBE filters block unsolicited messages according to information in the message header and envelope as described in “About the UBE Plug-In” on page 216.

You can create a set of UBE filters that work with the UBE Plug-in to hinder the practice of unauthorized-relaying. Such filters can prevent most UBE from being sent through your system to others.

If your installation has a separate external messaging server, outside your firewall, that receives external mail but does not forward outbound internal mail, an anti-relay filter can effectively stop relaying by examining the `Channel-To` envelope field, like this:

```
Channel-To ".*@xyzcorp\.com" EXIT
$ANY ".*" REJECT "We accept mail for XYZ Corporation only"
```

The first filter passes through any mail destined for the internal domain of XYZ Corporation and then exits. The second filter rejects any mail that does not match the first filter.

If your installation uses the same messaging server for both internal and external mail, the filtering requires an extra preliminary step to allow your users to send mail outside your installation. Add a filter that first checks the source of the message, by looking at the `Host-From` field:

```
Host-From "123.45.67.*" EXIT
Channel-To ".*@xyzcorp\.com" EXIT
$ANY ".*" REJECT "We accept mail for XYZ Corporation only"
```

In this case, the first filter passes through any mail that originates from the subnet belonging to XYZ Corporation and then exits. The second filter passes through any mail that does not match the first filter but is destined for the internal domain of XYZ Corporation. The third filter rejects any mail that does not match either of the first two filters.

Note: Using the same messaging server for both internal and external mail is not a recommended configuration. It makes your messaging system more open to external attacks.

Using the Anti-Relay Plug-In

Netscape supplies a protocol-level plug-in that you can use to prevent your servers from being used to relay UBE to other sites. Because this anti-relay plug-in operates at the protocol level, it intercepts and rejects unauthorized messages before the server accepts them and writes them to disk. This saves both disk space and processing resources. Netscape provides both the ready to use anti-relay plug-in, and the plug-in's source code so that you can modify or enhance it to meet your special requirements.

The anti-relay plug-in uses two sets of rules to prevent relaying:

- **Delivery rules** identify the users at your site for whom Messaging Server will accept incoming messages. Messages addressed to those identified by a delivery rule are accepted and delivered regardless of who submits the messages.

- **Submission rules** identify the hosts from whom Messaging Server will accept messages for initial transmission, relay, or delivery.

  Messaging Server will only pass along messages that it receives from hosts identified by a submission rule or messages from any source that are addressed to those identified by a delivery rule. (Note however, that by default the anti-relay plug-in is also configured to accept messages from users who have been authenticated by SMTP regardless of whether or not they are using a host identified by a submission rule.)

  If you use the anti-relay plug-in, the only users who will be able to send email to outside destinations (that is, destinations not identified by a delivery rule) are those who have been authenticated by SMTP (if you are using that option) or who are sending a message from a host identified by a submission rule.

Keep in mind that the anti-relay plug-in is not a substitute for UBE filters. Only appropriate filters can stop the delivery of UBE to users at your site. What the anti-relay plug-in does is prevent your site from being used to relay UBE to people at other sites.

In addition to the anti-relay plug-in, Netscape provides you with the plug-in's source code so that you can modify it to meet your special needs if you wish. The source code is found in `server-root/plugins/antirelay/`.

Enabling the Anti-Relay Plug-In

To enable the anti-relay plug-in:

1. Use `configutil` to specify that SMTP is to load the anti-relay plug-in when it starts up.

   The anti-relay plug-in must be loaded with the `configutil -l` option to specify that it run locally. A full path to the anti-relay plug-in must be specified.

In Unix environments, enter the command:

```
configutil -l -o service.smtp.protplugmodules -v \
'/usr/netscape/suitespot4/plugins/antirelay.so'
```

In NT environments, enter the command:

```
configutil -l -o service.smtp.protplugmodules -v
C:\Netscape\Server4\plugins\antirelay.dll
```

2. Configure the anti-relay plug-in as described in the next section.

Configuring the Anti-Relay Plug-In

You control how the anti-relay plug-in operates, and what relaying is permitted, through configuration options. The plug-in configuration options are specified in a plain text file named `antirelay.conf` that is stored in the server-instance configuration directory.

- In Unix environments, the configuration directory is specified by the `CONFIGROOT` environment variable and is typically *`server-root`*`/config/smtp/config`.
- In NT environments, the configuration directory is typically *`server-root`*`\msg-`*`instance`*`\config`.

To specify configuration options, you edit the `antirelay.conf` file. Specify one option per line in the file. Each line is colon delimited into two fields. The first field contains the key (option name) and the second field contains the data. Blank lines are ignored, and lines that begin with # are treated as comments.

For example, a typical `antirelay.conf` file might look like this:

```
# Anti-relay configuration file for server nsmail7
  resolvehostnames:0
  useauthinfo:0
  advertiseauthinfo:1

# We accept mail addressed to these recipients:
  delivery:*@airius.com

# We relay messages coming from these hosts:
  submission:*.airius.com
  submission:mailserver.ourfriend.com
  submission:127.0.0.1
```

Keep in mind that the anti-relay plug-in processes `submission` and `delivery` statements in reverse order (the last criteria is evaluated first) so you may want to optimize performance by listing the most common cases last in the file.

Anti-Relay Plug-In Configuration Options

The anti-relay plug-in configuration options are:

- **`delivery:`** Destination email addresses for which relayed messages will be accepted for delivery by Messaging Server. In other words, if a relayed message is addressed to a user identified in a `delivery:` statement, it will be delivered to the recipient's mailbox. Normally, your own users are identified so that their incoming messages are delivered. You can have multiple `delivery:` statements in an `antirelay.conf` file. You can use patterns such as `*@airius.com` in delivery statements. (For more detailed information on how to specify email addresses in `delivery:` statements, see "Specifying Anti-Relay User and Domain Names" on page 252 .)

- **`submission:`** A host or domain from which messages will be relayed by Messaging Server. In other words, if a message to be relayed comes from a source identified in a `submission:` statement, it will be relayed. Normally, your own hosts and domains are listed so that your users' outgoing messages are dispatched.You can have multiple `submission:` statements in an `antirelay.conf` file. You can use patterns such as `*.airius.com` in delivery statements. (For more detailed information on how to specify host and domain names in `submission:` statements, see "Specifying Anti-Relay User and Domain Names" on page 252 .)

- **`resolvehostnames:`** By default, the anti-relay plug-in resolves host names if Messaging Server is configured to resolve peer addresses into host names. If Messaging Server is not configured to resolve host names, you can only use IP addresses in `delivery:` and `submission:` statements. If your server is configured to resolve host names, you can choose to allow the anti-relay plug-in to use host names or force it to require IP addresses by setting the value of the `resolvehostnames` option. The allowed values are:

  - `resolvehostnames`:0
    Forces the anti-relay plug-in to us IP addresses

  - `resolvehostnames:1`
    Allows anti-relay plug-in to use host names. (This is the default value.)

- **useauthinfo:** By default, the anti-relay plug-in trusts all SMTP authenticated users as if they are connected from a valid submission domain. To change this, reset the useauthinfo: value. The allowed values are:

  - useauthinfo:0
    Do not automatically trust SMTP authenticated users.

  - useauthinfo:1
    Trust all users authenticated by SMTP. (This is the default value.)

- **advertiseauthinfo:** By default, the anti-relay plug-in does not provide non-authenticated clients who are denied relay privileges any indication that they might be allowed to relay if they were authenticated. To change this, reset the advertiseauthinfo: value. The allowed values are:

  - advertiseauthinfo:0
    Rejected users get the message: 551 delivery not allowed to non-local recipient. (This is the default value.)

  - advertiseauthinfo:1
    Rejected users get the message: 530 delivery not allowed to non-local recipient, try authenticating.

Specifying Anti-Relay User and Domain Names

The submission: statements in the anti-relay plug-in configuration file identify the hosts and domains authorized to submit messages to the server for relaying. Hosts and domains are specified in submission: statements with IP addresses or DNS domain names. If your server is not configured to resolve peer addresses into host names (configuration parameter service.smtp.doclientdnslookup) all hosts and domains in submission: statements must be specified using IP addresses.

The delivery: statements in the anti-relay plug-in configuration file identify the users the server is authorized to relay messages to. Users are specified in delivery: statements with email addresses.

For both submission: and delivery: statements, an asterisk can be used as a wildcard in IP, DNS, or email addresses. For example:

- submission:*.airius.com only permits hosts in that domain to submit messages for relay.

- `submission:*.airius.com` specifies that messages will be relayed from all hosts in `airius.com`, and all hosts in subdomains of `airius.com`.
- `submission:123.123.123.*` specifies that messages will be relayed from all hosts with IP addresses that begin with `123.123.123`.
- `delivery:*@*.airius.com` specifies that messages will be relayed to all users in `airius.com` and its subdomains.
- `delivery:*@airius.com` only permits delivery to users in that domain; messages will not be delivered to hosts in subdomains of `airius.com` unless they are identified in other `submission:` statements.

You can specify individual hosts, rather than all hosts in a domain by specifying the system's IP address or DNS name.

For example, `submission:mymachine.airius.com` specifies that messages originating from the `mymachine` host in `airius.com` are to be relayed but messages from other hosts in `airius.com` are not to be relayed unless they are identified in other `submission:` statements.

Chapter

9

Message Routing

This chapter describes how Netscape Messaging Server receives and delivers a message. This chapter contains the following sections:

- Overview
- How Messaging Server Routes Messages
- About Alternate Search Methods
- About Recipient Address Rewrites
- About Mailing List Expansion and Delivery
- About the Domain Name System (DNS)

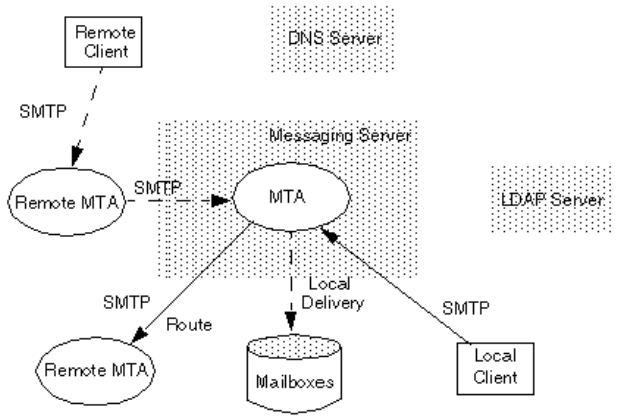
Overview

When a client sends a message to a server, the server must determine whether it can deliver the message locally to its own message store or if it must route the message to another server. If the message is sent to another server, the other server must make the same decisions about the message: deliver locally or route to yet another server. In this way, messages are routed throughout the Internet until they reach the intended recipient. After a message is delivered successfully to a server, the recipient can then retrieve the message.

Figure 9.1 shows the routes taken by two messages:

- The dotted lines show a message originating from a remote client and delivered to a local mailbox.
- The solid lines show a message originating from a local client and routed to a remote messaging server.

Figure 9.1 Message routing



The clients and servers use the Domain Name System (DNS) to find the IP addresses of machines to which they are sending. The Messaging Server Message Transfer Agent (MTA) is responsible for accepting messages and determining how to route them. The MTA uses information stored in the LDAP Directory Server to make its decisions.

The following sections briefly outline the steps taken to send, route, and retrieve a message. Details about how Netscape Messaging Server routes messages are described later in this chapter.

Sending a Message

To send a message, the client must know the IP address of the messaging server to which it is sending; the client then establishes a connection to the server and sends the message using the SMTP protocol. The following summarizes the steps a client takes to send a message:

1. Queries a DNS server to find the IP address of the server.
2. Establishes a TCP/IP connection to the server.
3. (Optional) Establishes an SSL connection to the server.
4. Sends the envelope information and the message to the server (SMTP-Deliver).

Routing a Message

Messaging Server listens for incoming mail on port 25 (the standard SMTP port). After accepting a message (SMTP-Accept), Messaging Server must determine whether the message is destined for a local recipient or a remote recipient. To do this, Messaging Server must establish a connection with the Directory Server that contains this information. The following briefly summarizes the steps Messaging Server takes to route a message.

1. Queries the Directory Server to determine whether the recipient is local or remote.
2. If the recipient is local, delivers the message, typically placing it in the message store.

   Otherwise, proceeds to step 3.
3. If the recipient is remote:

   a. Queries DNS to find the MX (mail exchange) servers for the domain.

   b. Queries DNS to find the IP address of the remote messaging server.

   c. Establishes a TCP/IP connection to the remote messaging server.

   d. (Optional) Establishes an SSL connection to the remote messaging server.

   e. Sends the message to the remote messaging server (SMTP-Deliver).

Retrieving a Message

To retrieve a message, the client must know the IP address of the messaging server, establish a connection to the server, then retrieve the message using one of the retrieval protocols: POP, IMAP, or HTTP. The following summarizes the steps the client takes to retrieve a message.

1. Queries DNS to find the IP address of the server.
2. Establishes a TCP/IP connection to the server.
3. (Optional) Establishes an SSL connection to the server.
4. Establishes a POP3, IMAP4, or HTTP connection to the server to retrieve the message.

How Messaging Server Routes Messages

This section describes, in detail, how Netscape Messaging Server routes messages. After the server accepts a message, it will handle the message by choosing one of the following options:

- **Delivers** the message to a local recipient (a mailbox or a program)
- **Routes** the message to another messaging server
- **Resends** the message (if the target is a mail group or a forwarded account)
- **Rejects** the message (if a recipient address cannot be resolved)

It is possible for Messaging Server to refuse to accept a message. For example, PreSMTPAccept plug-ins act before SMTP has accepted the incoming message (for information, see Chapter 7, “Working with SMTP Plug-Ins”). For another example, you can configure the server to refuse messages that exceed a given size (for information, see Chapter 3, “Configuring SMTP Services”).

To map addresses and route messages, Messaging Server uses information stored as LDAP attributes in Netscape Directory Server. These attributes are described in Table 9.1 and in Table 9.2. In Table 9.1, the addressing attributes determine which LDAP entry handles the address; the routing attributes determine which server handles an LDAP entry. In Table 9.2, the server configuration options determine domain information.

You can set values for the following attributes by using Netscape Console or by using the command-line interface. For information about setting the user attributes, see Chapter 4, "Managing Mail Users and Mailing Lists." For information about setting the server attributes, see Chapter 3, "Configuring SMTP Services." An exception is the `mailRoutingAddress` attribute. You can set this attribute only by using Netscape Directory Server tools, such as `ldapmodify`. For more information about LDAP object classes and setting attributes using LDAP tools, see the *Netscape Directory Server Administration Guide* and the *Netscape Directory Server Schema Reference Manual*.

Table 9.1 User attributes

Addressing	Description
`mail`	Identifies the user's email address
`mailAlternateAddress`	Identifies the user's alternate email address(es)
`uid`	Identifies the user
Routing	**Description**
`mailHost`	Identifies the host on which the user's mailbox resides
`mailRoutingAddress`	Identifies the address to place in the envelope when routing a message to this user

Table 9.2 Server options

Attribute	Description
`service.smtp.defaultdomain`	Identifies the domain name used to complete addresses that do not contain a domain name
`service.smtp.smtp-router.localmaildomains`	Identifies domains local to Messaging Server
`service.smtp.domainname`	Identifies the domain of this server machine
`service.smtp.messagehostname`	Identifies the host name on which Messaging Server resides

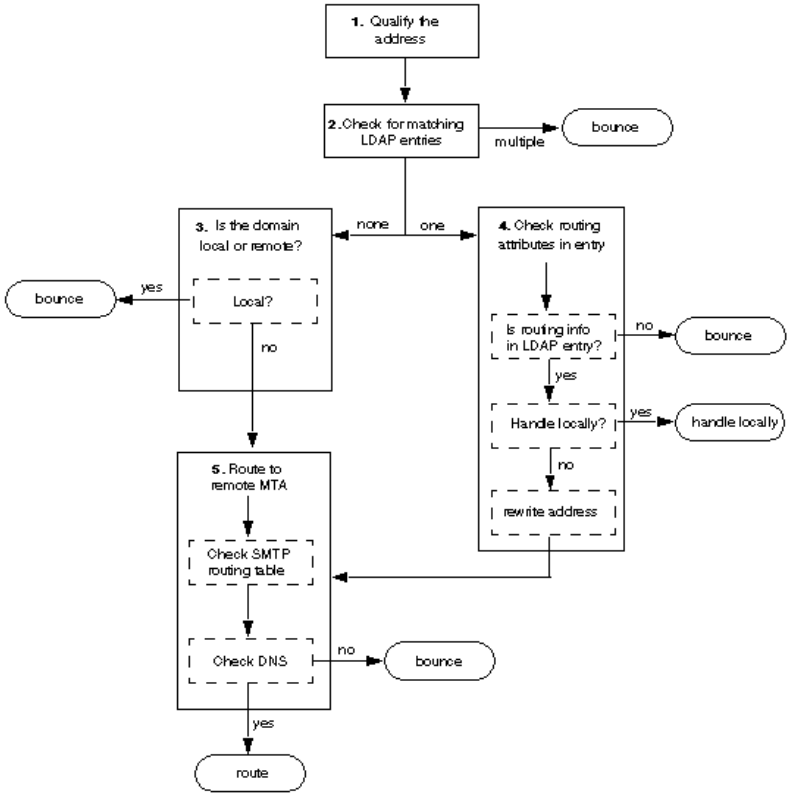
To determine how to route a message, Messaging Server must decide the following:

- Does the recipient address have a matching LDAP entry?
- Is the recipient address local or remote?
- If the recipient address is remote, to where should the message be routed?

To make these decisions, the server performs the following steps, which are illustrated in Figure 9.2 , and described in detail throughout this chapter.

1. Qualifies the SMTP Address.
2. Searches for matching LDAP entries. If no match is found, proceeds to step 3; if exactly one match is found, proceeds to step 4.
3. Checks to see if the domain is local or remote. If the domain is local, searches for the recipient using the specified search method. If the domain is remote, proceeds to step 5.
4. Checks routing attributes in LDAP entry.
5. Routes to remote server.

Figure 9.2 Routing diagram



Step 1: Qualifying the Address

Messaging Server requires standard, fully-qualified SMTP addresses in the `RCPT TO:` field of the message envelope (for example, `joe@airius.com`). If an envelope does not have a fully-qualified SMTP address (for example, `joe`), Messaging Server attempts to make the address standard by adding the domain to the address as follows:

1. If the `service.smtp.defaultdomain` attribute has been defined, Messaging Server appends the value defined for this attribute to the envelope address. (For information about specifying this attribute, see "Specifying an Address Completion Domain" on page 91.)

2. If the `service.smtp.defaultdomain` attribute has not been defined, Messaging Server appends the value defined for the `service.smtp.messagehostname` attribute to the envelope address (for example, `joe@airius.com`).

3. If the address is of the form `joe@server`, Messaging Server appends the value defined for its server domain extension (`service.smtp.domainname`) to the envelope address (for example, `joe@server.airius.com`).

4. If the address includes an IP address, such as `joe@[198.93.93.10]`, Messaging Server will perform IP address resolution and replace the IP address with the matching host name (for example, `joe@gethost`).

Step 2: Searching for matching LDAP Entries

Messaging Server searches for an LDAP entry with a `mail` or `mailAlternateAddress` attribute that matches the envelope recipient address. (For information about specifying these attributes, see "Specifying User Email Addresses" on page 128.)

The search includes only LDAP entries that 1) have object class `mailRecipient` or `mailGroup` and 2) are in the directory subtree specified by the configured Base DN.

- If no match is found, Messaging Server tries any alternate search methods that are enabled. If it still cannot find a match, Messaging Server proceeds to Step 3.

  For more information about alternate search methods, see "About Alternate Search Methods" on page 266.

- If exactly one matching address is found, Messaging Server proceeds to Step 4.

- If multiple matches are found, Messaging Server treats the address as invalid and bounces the message.

Step 3: Checking if Domain is Local or Remote

If Messaging Server cannot find a matching LDAP entry for an address, it must decide whether the domain specified in the address is local or remote. It considers the address local if the domain part of the address matches:

- The fully-qualified domain name of the host on which Messaging Server resides
- A local domain setting if set (For information on setting the local domain, see "Specifying the Domains Local to Your Server" on page 92.)

If the domain is local, Messaging Server assumes the message is addressed to an unknown user and bounces the message.

Otherwise, Messaging Server considers the address to be remote and proceeds to Step 5.

Step 4: Checking Routing Attributes

If Messaging Server finds exactly one LDAP entry that matches the address, it checks the routing attributes for the address and takes the following actions:

- If the LDAP entry contains a `mailHost` attribute, Messaging Server will deliver or route the message to the server specified by the attribute:

  Local Server. If the `mailHost` attribute indicates the local server (as defined by `MessageHostName`), then Messaging Server uses the entry's `uid` attribute for local delivery and handles the message locally.

  Remote Server. If the `mailHost` attribute indicates a remote server, Messaging Server rewrites the address (see "About Recipient Address Rewrites") and routes the message to the remote server as described in Step 5.

- If the LDAP entry does not contain a `mailHost` attribute, but does contain an enabled `mailRoutingAddress` attribute, Messaging Server rewrites the address (see "About Recipient Address Rewrites") and routes the message to the remote server as described in Step 5.

Note: You should not use the `mailRoutingAddress` attribute for local users.

- If the LDAP entry contains no routing information (that is, no `mailHost` attribute and no enabled `mailRoutingAddress` attribute) but specifies that the account is a mailing list, Messaging Server handles the message locally. For more information about how the server handles delivery to mailing lists, see "About Mailing List Expansion and Delivery" on page 270.
- If the LDAP entry contains no `mailHost` attribute and cannot be routed or handled locally, Messaging Server rejects the message.

For information about how to specify the `mailHost` attribute, see "Specifying User Email Addresses" on page 128. To set the `mailRoutingAddress` attribute, you must use Netscape Directory Server tools. For information, see the *Netscape Directory Server Administrator's Guide*.

Step 5: Routing to Remote Server

If Messaging Server assumes another messaging server is responsible for this recipient, it performs the following steps to route the message to a remote server:

1. Checks the SMTP routing table
2. Looks up address of the server in DNS
   - Use MX records if present
   - Otherwise, use A records

Checking the SMTP Routing Table

To see if mail for the recipient's domain should be routed to a specific messaging server host, Messaging Server checks its SMTP routing table.

- If the routing information is an asterisk "*", Messaging Server routes the message to the server specified in the MX record for the domain. (An "*" entry prevents entries occurring later in the routing table from handling the domain.)

- If the routing information includes an IP address in square brackets, Messaging Server routes the message to the remote server at that address (and need not perform a DNS lookup).
- If the routing information does not include an IP address, but includes a domain name, Messaging Server routes the message to a mail exchange server for that domain.

To find an IP address, Messaging Server checks the DNS definition for the domain. See "About the Domain Name System (DNS)" on page 273.

Example Routes

This section provides sample routing table entries. For information about how to specify routing table entries using Netscape Console, see "Editing SMTP Routing Table Entries" on page 111.

In the following example, all outgoing mail is routed to `bigserver`:

```
*:bigserver.airius.com
```

The next example routes all mail to subdomains of `airius.com` through a hub server:

```
*.airius.com:hub.airius.com
```

In the next example, messages for any subdomain inside the top-level domain `airius.com` are sent directly to the mail exchange server for the respective domain:

```
*.airius.com:*
```

The next example directs mail for the hub to the server with the specified IP address. Messaging Server need not perform a DNS lookup to find the IP address.

```
hub.airius.com:[123.345.456.7]
```

The next example directs all mail sent outside `airius.com` to a firewall server:

```
airius.com:*
*.airius.com:*
*:firewall.airius.com
```

Note: Keep in mind that the server processes entries in the SMTP routing table in order. If, for example, you have a routing entry that sends all non-local mail to a firewall messaging server, you want this entry to be the last entry in the routing table.

Looking Up Address of the Server in DNS

If a routing table entry does not include an IP address in brackets, Messaging Server must know the IP address of the host machine to which it is sending. To find the IP address, Messaging Server uses the Domain Name System (DNS) as follows. For more information on DNS and DNS records, see "About the Domain Name System (DNS)" on page 273.

1. Asks for Mail Exchange Records (MX records) defined for the domain. If MX records are found, tries to route the message to each messaging server defined in the MX records until routing is successful. The MX record must have an associated Address record (A record) that includes the IP address. (Note that Messaging Server caches MX records to decrease the number of DNS lookups required.)

2. If no MX record is found, asks for the A record for the domain. If the A record is found, routes the message to the host specified for the domain.

If Messaging Server cannot find an IP address for the machine to which it is routing a message, the message is considered undeliverable and Messaging Server bounces the message.

About Alternate Search Methods

If an exact match for an address is not found, but alternate search methods are enabled, the server tries each method in the order listed until a match is found:

- Search for custom domain
- Search using truncated domain
- Search by user ID

For information about how to enable alternate search methods using Netscape Console, see "Specifying Alternate Search Methods" on page 110.

Note: Specifying alternate search methods has a slight impact on performance.

Search for custom domain. In a hosting environment, you might want the server to search for custom domain addresses.

To set up a custom domain for a user, add an alternate address for the user that includes the custom domain name. For example, assume Joe wants to receive mail addressed to `joe@airius.com` and a mail addressed to the custom domain, `joecorp.com`. To enable Joe to receive mail addressed to *`anything`*`@joecorp.com`, you must add a `MailAlternateAddress` value for Joe as follows: `@joecorp.com`. You must also add the MX records in DNS as necessary to route messages sent to the custom domain to the desired messaging server. For more information about DNS and MX records, see "About the Domain Name System (DNS)" on page 273.

Search using truncated domain. In some environments, you might want the server to ignore the first component of a domain when searching for an address in LDAP. This allows the server to handle mail sent to `recipient@server` instead of `recipient@domain`.

For example, assume mail arrives for `joe@foo.airius.com`. With the "search using truncated domain" feature enabled, if no LDAP entry for `joe@foo.airius.com` is found, the server will search for `joe@airius.com`. Consequently, Joe can receive messages addressed to `joe@`*`anything`*`.airius.com` as well as `joe@airius.com`.

> **Note:** Use this method only if for all domains listed in LDAP, all domains one level down have the same email namespace. For example, if `user@host1.airius.com`, `user@host2.airius.com`, and `user@airius.com` are considered different accounts, do not enable this method.

Search by user ID. With the "search by user ID" method enabled, if an address domain is local, Messaging Server searches for an LDAP entry with a `uid` attribute matching the address (with the `@domain` removed). For example, if `airius.com` is local, and the address is `joe@airius.com`, Messaging Server searches for a uid of "joe".

> **Note:** This method is for compatibility with earlier versions of Netscape Messaging Server. If your installation is new, Netscape recommends that you disable this method.

About Recipient Address Rewrites

If the recipient was found in LDAP, but the `mailHost` attribute in the LDAP entry is not this server, Messaging Server then tries to route the message to a remote server. You can specify whether and how the server rewrites the envelope recipient address before routing the message to the remote server.

Note: Messaging Server rewrites the envelope address only if the intended recipient is found in LDAP and is not local. (The server rewrites the envelope recipient address in the SMTP protocol dialog.)

You can specify one or more of the following rewrite methods. By default, these methods are disabled. Messaging Server tries each enabled method in the order listed until it is able to compose a new address using the indicated attributes from the account's LDAP entry:

- Use the `mailRoutingAddress` attribute
- Combine the `uid` with the `mailHost` attribute
- Combine the local part of the address with the `mailHost` attribute

If Messaging Server is unable to compose a new address (because the necessary attributes are not present in the user's LDAP entry or because no rewrite methods are selected), the server will not rewrite the envelope address. However, if the "search by truncated domain" method was attempted in Step 2, the server uses the truncated domain instead of the original domain.

mailRoutingAddress Attribute

This rewrite method uses the `mailRoutingAddress` attribute, which specifies a specific mail routing address. This method is most useful for LDAP entries that represent mail accounts on non-Netscape mail servers or gateway systems.

If you enable this rewrite method, you must modify the user's LDAP entry to include the `mailRoutingAddress` attribute. You can set this attribute only by using LDAP tools such as `ldapmodify`. For example:

```
mailRoutingAddress: joesmith@judge.airius.com
```

The `mailRoutingAddress` attribute differs from the `mailForwardingAddress` attribute as follows:

- `mailRoutingAddress` determines how to route the message to the server that handles this account (the mail account that is represented by this LDAP entry)
- `mailForwardingAddress` determines how a mail account forwards its mail to some other account

Combine uid and mailHost Attributes

This rewrite method combines the `uid` attribute and the `mailHost` attribute found in the LDAP directory.

For example, mail arrives on one server for `Joe_Smith@airius.com`. The server determines that this mail belongs to `jsmith` whose mail account is on `judge.airius.com`. The server rewrites the envelope address to `jsmith@judge.airius.com` then relays the message to `judge.airius.com`.

This method works best if the "search by user ID" method is employed on the next server.

Note: Some sites prefer that only explicit addresses (those specified by the `mail` and `mailAlternateAddress` attributes) are valid email addresses for users. You should not use this method if the local policy does not consider `uid` a valid email address.

Combine Local Part and mailHost Attribute

This rewrite method combines the local part of the original address with the `mailHost` attribute value to create the new address.

For example, `Customer_Service@airius.com` becomes `Customer_Service@judge.airius.com`. This method is useful to support entities, such as mail groups, that do not have a `uid`.

Some SMTP installations prefer that addresses routed internally within the network be host-specific. This feature is most likely to work properly if the "truncated domain" search method is in use on the next server, or if all user accounts have a matching email address explicitly specified.

Do not use this method unless, for each address of each user (as specified with the `mail` and `mailAlternateAddress` attributes), changing the domain part to a specific host does not create ambiguity about the message recipient. For example, suppose the messaging server `mail5.airius.com` has three different users: `joe@division1.airius.com`, `joe@division2.airius.com`, and `joe@airius.com`. In this scenario, `joe@airius.com` would receive mail addressed to the other users, if the "local part at mailHost" rewrite method is used on the sending server and the "truncated domain" search method is used on `mail5.airius.com`.

Note: This method is not used if the "custom domain" search method was used to resolve the address.

About Mailing List Expansion and Delivery

When a message is sent to a mailing list, Messaging Server determines whether this list is handled locally by checking the `mailHost` attribute defined for the list. If there is no `mailHost` attribute defined, or if the `mailHost` attribute indicates the local server, Message Server assumes the mailing list is local and performs the steps outlined in this section.

Note that attributes for mailing lists, including the restrictions and actions described throughout this section, are defined using the Netscape Console Users and Groups interface. For more information about mailing list attributes, see Chapter 4, "Managing Mail Users and Mailing Lists."

Step 1. Checking Restrictions. Messaging Server checks for the following restrictions before delivering the message to the list:

- Allowed sender restriction. Messaging Server checks to see if the sender is allowed to send messages to this list.
- Allowed sender domain restriction. Messaging Server checks to see if the message sender's domain is allowed to send messages to this list.
- Maximum message size restriction. Messaging Server checks to see if the message size is within the limits set up for this list.

- Password restriction. Messaging Server checks to see if password restriction is in effect for senders to this list. If the password is present—in an Approved header or in the first line of the message—Messaging Server strips the password from the message before delivering it to the list.

If Messaging Server must reject a message, it goes to step 2; otherwise it proceeds to step 3.

Step 2. Rejecting a Message. If Messaging Server must reject a message, it first checks to see if any of the following actions are specified for the list.

- Send message to moderator(s). Messaging Server sends the message to the moderator by adding the moderator(s) to the envelope recipient list and by adding a new header, "Moderation-from:".

- Send a reply. Messaging Server sends a reply to the sender (using the "Reply-To:" field if present; using the "From:" field if the "Reply-To:" field is not present).

  (You can configure the text for this reply using the Groups Actions tab in Netscape Console. If no text is specified, Messaging Server sends a default reply "Your message to this group has been rejected." You can also specify whether Messaging Server should send the original message as an attachment to the reply.)

Step 3. Delivering the Message. If the message passes all restrictions configured for the list, Messaging Server:

1. Checks to see if the mail list is moderated. If yes, Messaging Server checks to see if the message is being forwarded by the moderator (that is, the sender is a moderator). If the message has been forwarded (there is an attached message), Messaging Server strips the moderator's header from the message before delivering the message to the list.

2. Performs any header rewrites as necessary. For example, the `Resent-From` header is added.

   If error messages concerning delivery to the list should be sent to a particular email address (see the "Errors-To" field in the Groups Settings Tab), Messaging Server uses this address in the `Return-Path:` header. This ensures that bounced messages will return to the address configured in the "Errors-To" field and not to the original sender.

At this stage, Messaging Server splits the envelope into individual recipients and list recipients. By splitting the envelope, Messaging Server ensures that modifications for mailing list delivery are not applied to other recipients of the message. Messaging Server might further split the envelope for list recipients—particularly if the list contains many members. Each envelope split is considered a separate "control" envelope.

3. Messaging Server adds list members to the message recipient list. Members are added according to the type of membership:

   - Member with email-only membership. These members are added as email addresses. If the address is found in LDAP and is local to this server, then Messaging Server handles the recipient. If the recipient is not found in LDAP or is not local to this server, Messaging Server routes the message to the appropriate server according to the `mailHost` attribute or the domain.

   - Dynamic members. These members meet search criteria specified as dynamic criteria for email-only membership (see the Groups Email-Only Members tab) or are part of a dynamic group (see the Members Dynamic Group tab). Messaging Server checks the `mailHost` attribute for the entry to determine whether mail for the dynamic member can be handled locally or should be routed to a remote server.

   - Unique members are members defined as part of a static group (see Groups Static Group tab). Messaging Server checks to see if all members are in the LDAP directory. If any members cannot be found in the directory, the member is considered an unknown recipient.

More About Mailing Lists

This section provides further information about how Messaging Server delivers mail to mailing list members.

Duplicate checks. Messaging Server performs duplicate checks to see if any message recipients, including list members, are unique members.

Duplicate checks can impact server performance—especially for large lists containing 1000 or more members. You can specify that Messaging Server not perform duplicate checks by modifying the list attribute

`mgrpNoDuplicateChecks`. You can modify this attribute by using the `ldapmodify` utility. For more information on this utility, see the *Netscape Directory Server Administrator's Guide*.

Nested Groups. Lists can contain other lists as members.

- For static groups, nested groups are expanded with the original group.
- Nesting of dynamic groups can cause degradation in group handling performance if membership is not handled correctly. Consequently, Messaging Server does not, by default, automatically nest dynamic groups. If you want to nest dynamic groups, you can set the configuration parameter `service.smtp.smtp-router.nestedgroups` to `yes`.
- A nested group will still enforce all its restrictions even when nested within another group. Consequently, spammers cannot avoid group restrictions by nesting groups within groups.
- Nested groups are handled by recursion. Consequently, several layers of nested groups can affect server performance. Netscape recommends no more than 3 or 4 levels of nested groups.

Recipients per control envelope. If a message sent to a list is routed to another MTA, Messaging Server limits the number of recipients sent in each control envelope. The default number is 500 recipients per control envelope.

Error notification. If error notification (Errors-To field) is configured for the group, to easily track members without mail accounts, delivery status notification (DSN) extensions are added to each member.

About the Domain Name System (DNS)

This section provides a brief overview of the Domain Name System (DNS). For complete details about DNS, see the book, *DNS and Bind, 2nd Edition* by Paul Albitz and Cricket Liu, published by O'Reilly. DNS is the naming system that allows computers to find each other on the Internet. Every computer needs a DNS name to communicate on the Internet.

The DNS is managed by DNS servers that store information about domains and individual host systems. Each unit of data in the DNS distributed database is indexed by a name. These names are essentially just paths in a large inverted tree, called the domain name space.

A Fully Qualified Domain Name (FQDN) is the unique name that identifies a specific Internet location. FQDNs consist of two or more components, separated by dots. Each section is a string of letters or numbers without spaces, usually a recognizable word or abbreviation. The order of the sections in an FQDN is significant.

As you move from left to right, each section represents a more general level in the DNS hierarchy. As an example, `server1.airius.com`, would refer to a target system `server1` at an organization called `airius` which is a subdomain of the top-level domain `com`, which designates it as a company.

A DNS server helps a messaging server convert the Internet domain name in an email address into an IP (Internet Protocol) address. Once the messaging server knows the other computer's IP address, the messaging server is able to contact the other computer and forward messages to it. The DNS server provides information such as the following to the messaging server:

- Identification of messaging server hosts (for example, which messaging servers handle incoming mail for this domain?)
- Address resolution (for example, what is the IP address for the name `server1.airius.com`?)
- Reverse address resolution (for example, what is the name for the IP address 195.95.92.6?)

The DNS server uses the records that have been set up within it to tell querying computers how to route mail to the local messaging server host.

DNS Records

DNS servers contain several record types. This section focuses on three record types, described in Table 9.3. These record types should be entered in the primary DNS server for a domain. (For information about other record types, see the book, *DNS and Bind*.)

Table 9.3 DNS records

Record Type	Function
A record	Map host name to IP address
MX record	Map domain name to host name of server that handles incoming mail for the domain
PTR record	Map IP address to host name

You can use A records or a combination of A records and MX records to resolve domain names to IP addresses. To reduce the load on the DNS system, domains that receive a significant amount of mail should have at least one MX record.

Using A Records

You can use an A record at the local DNS server to resolve the local server's FQDN to its IP address. For example, the following record indicates that the IP address of `server1.airius.com` is `195.95.92.6`:

```
server1.airius.com. IN A 195.95.92.6
```

Using MX Records

You can use an MX or *mail exchange* record to route messages for one domain to one or more hosts running a messaging server. For messaging services, it is recommended that you use MX records in addition to A records. All messaging servers look for MX records first in their search to identify an external mail host.

In addition, MX records establish priority. Priority rankings enable you to balance the load of incoming messages and to identify backup mail servers by inserting additional records in your DNS server.

Example 1. In the following example, inbound messages addressed to `airius.com` (for example j`oe@airius.com`) are routed to the `server1.airius.com` host; which is assumed to be a messaging server. (Note that the example would not affect inbound messages addressed to *`host`*`.airius.com`.)

```
airius.com IN MX 10 server1.airius.com
```

The record contains two fields: a priority field (`10`) and a host field (`server1.airius.com`). The priority field specifies the priority of this mail exchange (lower numbers have a higher priority). The host field is the name of a mail exchange host.

Example 2. The next example provides a backup destination for times when the main destination is unavailable. The first line indicates that mail routed to `airius.com` should be routed to `server1.airius.com`. The second line indicates that if `server1.airius.com` is unavailable for any reason, messages are sent to an alternate server, named `backupserver.airius.com`:

```
airius.com IN MX 10 server1.airius.com
airius.com IN MX 20 backupserver.airius.com
```

Example 3. The next example balances the load between two servers, `server1.airius.com` and `server2.airius.com`. If both servers are unavailable, mail is routed to the alternate server, `backupserver.airius.com`:

```
airius.com IN MX 10 server1.airius.com
airius.com IN MX 10 server2.airius.com
airius.com IN MX 20 backupserver.airius.com
```

Note: The exchange host name in any MX record must have at least one A record that maps the host name to an IP address. Considering example 3, the exchange hosts `server1.airius.com`, `server2.airius.com`, and `backupserver.airius.com` must have A records, such as the following (note that the domain `airius.com` does not require an A record):

```
server1.airius.com. IN A 195.95.92.6
server2.airius.com. IN A 195.95.92.7
backupserver.airius.com IN A 195.95.92.8
```

Using PTR Records

PTR or pointer records map IP addresses to names for purposes of reverse address resolution. The following example creates a PTR record for the IP address `195.95.92.6`:

```
6.92.95.195.in-addr.arpa. IN PTR server1.airius.com
7.92.95.195.in-addr.arpa. IN PTR server2.airius.com
8.92.95.195.in-addr.arpa. IN PTR backupserver.airius.com
```

Chapter

10

Monitoring and Maintaining Your Server

After you install and configure Netscape Messaging Server, you need to perform various tasks to monitor and maintain your server. Many of these tasks are performed automatically by the server. For example, the `stored` utility performs daily maintenance tasks for the server, such as erasing messages stored on disk according to expiration policies you specify. This chapter contains the following sections:

- Overview
- Performing Daily Tasks
- Starting and Stopping Services
- Monitoring and Controlling Disk Usage
- Monitoring Server Response Time
- Performing Recovery Tasks
- Factors Affecting Messaging Server Performance
- System Monitoring Tools
- Using SNMP on Unix Platforms

Overview

In most cases, a well-planned, well-configured server will perform from day to day and from month to month without requiring intervention from an administrator. As an administrator, however, it is your job to monitor the server for exception conditions that require action on your part to keep the server running smoothly. Tasks you should perform include:

- Monitoring disk usage
- Monitoring server performance and response times
- Managing exception conditions when and if they occur
- Reconfiguring, when necessary, to accommodate new users or new conditions

This chapter focuses on Messaging Server 4.1 configuration and maintenance. However, you will also need to monitor the system on which the server resides. A well-configured server cannot perform well on a poorly-tuned system. For example, you should monitor the following conditions:

- CPU performance
- Disk I/O
- Memory usage
- Network performance

This chapter does not provide details about system monitoring and performance. However, it does provide information about tools you can use to monitor system performance; see "System Monitoring Tools" on page 296.

Netscape Messaging Server provides several command-line utilities for monitoring and maintaining your server, as described in Table 10.1. For complete syntax reference and usage guidelines for these utilities, see Appendix A, "Command-line Utilities."

Table 10.1 Command-line utilities

Category	Command-Line Utility
Management	`configutil, imscripter, mboxutil, NscpMsg, processq`
Recovery	`deliver, reconstruct`
Background and daily tasks	`stored`
Monitoring and reporting	`counterutil, hashdir, mailq, quota, readership`

Performing Daily Tasks

Probably the most important tasks you should perform on a daily basis are checking `postmaster` mail and monitoring the log files.

Checking postmaster Mail

Messaging Server has a predefined administrative mailing list set up for `postmaster` email. Any users who are part of this mailing list will automatically receive mail addressed to `postmaster`.

The rules for `postmaster` mail are defined in RFC822, which requires every email site to accept mail addressed to a user or mailing list named `postmaster` and that mail sent to this address be delivered to a real person. All messages sent to `postmaster@host.domain` are sent to a postmaster account or mailing list.

Typically, the `postmaster` address is where users should send email about their mail service. As `postmaster`, you might receive mail from local users about server response time, from other server administrators who are encountering problems sending mail to your server, and so on. *You should check postmaster mail daily.*

You can also configure the server to send certain error messages to the `postmaster` address. For example, when the MTA cannot route or deliver a message, you can be notified via email sent to the `postmaster` address. You

can also send exception condition warnings (low disk space, poor server response) to `postmaster`. For more information about sending messages to `postmaster`, see "Specifying Error Handling" on page 106 and "Alarm Attributes" on page 437.

Monitoring and Maintaining the Log Files

Netscape Messaging Server creates a separate set of log files for each of the major protocols, or services, it supports: SMTP, IMAP, POP, and HTTP. You should monitor the log files on a routine basis—especially if you are having problems with the server. For information about searching and viewing log files, see "Searching and Viewing Logs" on page 318.

Be aware that logging can impact server performance. The more verbose the logging you specify, the more disk space your log files will occupy for a given amount of time. You should define effective but realistic log rotation, expiration, and backup policies for your server. For information about defining logging policies for your server, see "Defining and Setting Logging Options" on page 313.

Setting Up the stored Utility

The `stored` utility performs automatic monitoring and maintenance tasks for the server, such as:

- Background and daily messaging tasks
- Deadlock detection and rollback of deadlocked database transactions
- Cleanup of temporary files on startup
- Implementation of aging policies
- Periodic monitoring of server state, disk space, service response times, and so on
- Issuing of alarms if necessary

The `stored` utility automatically performs cleanup and expiration operations once a day at midnight. For more information about `stored`, see "stored" on page 433 and "Using the stored Utility" on page 165.

Starting and Stopping Services

If a server process crashes, other processes will hang as they wait for locks held by the server process that crashed. Therefore, if any of the server processes crash, you should stop all processes, then restart all processes. This includes the POP, IMAP, HTTP, and SMTP processes, as well as the `stored` (message store) process, and any utilities that modify the message store, such as `mboxutil`, `quota`, `deliver`, `reconstruct`, `readership`, or `upgrade`.

You can stop and start services by using the Netscape Console interface. You can also stop and start services as follows: On Windows NT, by using the Services Control Manager; on Unix platforms, by using the `NscpMsg` utility. For more information, see "NscpMsg" on page 423.

Monitoring and Controlling Disk Usage

The server requires adequate disk space for processing and storing messages. You must never let the server run out of disk space. Consequently, you must monitor disk usage and take appropriate actions if available disk space becomes too low.

The message queue contains messages that are in the process of being delivered or that cannot be routed or delivered immediately. The message store contains the messages delivered to local users and the contents of all their folders. Both the queue and the store must have adequate disk space for the messages they contain. For example, if disk space reserved for the store falls too low, the server might start queueing messages destined for the store. If disk space reserved for the queue falls too low, the server might start rejecting messages with a temporary failure condition code. For general information about the message queue, see "Message Queue Concepts" on page 114. For general information about the message store, see Chapter 5, "Managing the Message Store."

Monitoring Disk Usage

You can monitor disk usage by configuring the disk space alarm attributes described in Table 10.2. You configure these attributes by using the `configutil` utility. You can specify how often the system should monitor disk space and under what circumstances the system should send a warning.

Table 10.2 Disk space alarm attributes

Disk Space Attributes	Default Value
`alarm.diskavail.msgalarmstatinterval`	3600 seconds
`alarm.diskavail.msgalarmthreshold`	10%
`alarm.diskavail.msgalarmwarninginterval`	24 hours

For example, if you want the system to monitor disk space every 600 seconds, specify the following command:

```
configutil -o alarm.disavail.msgalarmstatinterval -v 600
```

If you want to receive a warning whenever available disk space falls below 20 %, specify the following command:

```
configutil -o alarm.diskavail.msgalarmthreshold -v 20
```

For more information about setting alarm attributes, see “Alarm Attributes” on page 437.

Controlling Disk Usage

You can control disk usage by setting user disk quotas, specifying aging policies for messages stored on disk, limiting the size of messages the server will accept, and processing the message queue at frequent intervals. If these methods are not sufficient or are not acceptable to your users (or are not practical), you might want to consider adding disks to your system. For more information, see “Specifying Alternate Paths for Physical Queues” on page 118 and “Configuring Message Store Partitions” on page 160.

User Disk Quotas

If disk space is limited, you might want to set user disk quotas for your system. Disk quotas allow you to limit users to a fixed mailbox size. If a user exceeds the limit, the user can no longer retrieve mail until he or she deletes existing messages to free disk space. For information about specifying message quotas, see "Configuring User Disk Quotas" on page 156.

You can use the `quota` utility to view reports and optionally fix mailbox quota usage. This utility generates a report listing quotas, giving their limits and usage. For more information about the `quota` utility, see Appendix A, "Command-line Utilities."

Aging Policies

Another method for controlling disk space is to specify aging policies for messages in the message store. You can control how long messages are stored in one or more mailboxes. If you set aging policies, you should educate your users about these policies because the server will not send a warning message before it starts deleting messages from the store.

You can specify constraints for the following:

- Number of messages in the mailbox
- Total size of the mailbox
- Number of days that messages remain in the mailbox
- Number of days that messages exceeding a given size remain in the mailbox

For more information, see "Specifying Aging Policies" on page 162.

Message Size Limits

You might want to limit the size of messages that your server will accept. If you limit message size, the server will automatically reject any messages that exceed the maximum message size. By limiting the size of messages, you'll save queue disk space and message store disk space. For more information about how to limit message size, see "Limiting Message Size (SIZE)" on page 103.

Reserved Disk Space

You can specify a minimum amount of disk space that will remain unused for the message queue. If the minimum threshold is reached, the server will temporarily reject all messages until disk space is freed. The server returns an error (452) notifying the client of a temporary disk space shortage and asking the client to resend the message at a later time. For information about how to reserve free disk space, see "Reserving Free Disk Space for the Message Queue" on page 100.

Session Cache

You must ensure that you have adequate disk space for the session cache. If you are running out of disk space when starting server processes, check your temporary directory (by default, this directory is located at *msg-instance*`/tmp`). By default, for each entry in an SSL server session cache file, 512 bytes are allocated on disk. To ensure adequate disk space, you should allocate 512 bytes per session ID.

You can also try setting the number of session entries to a lower number (the default number is 10,000). You can set this number as follows:

```
configutil -o service.imap.sslcachesize -v number
```

where *number* represents the number of session entries. Keep in mind that setting this value too low can adversely affect server performance.

If you want to move the temporary directory to another disk with more space, you can specify a path for the temporary directory as follows:

```
configutil -o local.tempdir -v path
```

Monitoring Server Response Time

In general, server response time is measured by the number of messages per second and the number of client connections per second your server can handle. There are many factors that affect server response time: number of users, peak traffic times, hardware and software configuration, network bandwidth, and so on.

Netscape Messaging Server provides a set of server response attributes you can use to monitor server response time for IMAP, POP, HTTP, and SMTP services. These attributes are listed in Table 10.3. Response time is measured for how long it takes to make a connection to a service and to receive the service greeting. If response time exceeds a specified number of seconds, you will be notified via email. You set values for the server response attributes by using the `configutil` utility. For more information about the `configutil` utility and the `msgalarmproc` attributes, see Appendix A, "Command-line Utilities."

Table 10.3 Server response attributes

Server Response Attributes	Default Value
`alarm.serverresponse.msgalarmstatinterval`	600 seconds
`alarm.serverresponse.msgalarmthreshold`	10 seconds
`alarm.serverresponse.msgalarmwarninginterval`	24 hours

To improve server response time, you can:

- Run `stored` at off hours—not during peak hours for your business.
- Defer queue processing to off hours.
- Reduce the size of the queue.
- Distribute the queue across multiple physical disks.
- Distribute the message store across multiple physical disks.
- Limit the size of user inboxes.

For more information about improving server response time, see "Factors Affecting Messaging Server Performance" on page 286.

Performing Recovery Tasks

If one or more mailboxes becomes corrupt, you can use the `reconstruct` utility to rebuild the mailboxes or the mailboxes database and repair any inconsistencies. For more information, see "Repairing Mailboxes and the Mailboxes Database" on page 166.

For information about backing up and restoring the message store, contact your Netscape technical support person.

Factors Affecting Messaging Server Performance

This section describes factors that affect Messaging Server 4.1 performance and contains tips to help you enhance the performance of Netscape Messaging Server 4.1. These tips are intended as general guidelines and suggestions only; the actual performance of your messaging server depends on many factors, including CPU power, disk space, usage patterns, network bandwidth, and so on.

Note: The tips in this document are for Netscape Messaging Server 4.1 only and might or might not apply to other versions of Netscape Messaging Server.

Factors that affect Netscape Messaging Server performance include the following:

- Number of Users per Disk
- Configuration of POP, IMAP, and HTTP Services
- Configuration of SMTP Services
- Configuration of Logging Services
- Size of Mailboxes
- Distribution of the Store and Queue Directories
- MTA Thread Settings
- Applications Co-Resident with Messaging Server
- Activity of Administration Server
- Activity of Directory Server
- Location of Messaging Server and Directory Server
- Number of Address Lookups per Message
- Ratio of Local Delivery to Outbound Sends
- Use of RAID Technology
- Memory, Disk, and CPU Requirements

Number of Users per Disk

As the number of users per disk increases, the I/O to that disk increases non-linearly due to the algorithms used by the OS to cache the directory index (it's faster to search memory than to search a hard drive). To improve disk access, you should distribute users across available disks.

To ease management of multiple disks, you can use RAID (Redundant Array of Inexpensive Disks) technology to distribute data across a series of physical disks. For more information, see "Use of RAID Technology" on page 295.

Configuration of POP, IMAP, and HTTP Services

How you configure your POP, IMAP, and HTTP services affects Messaging Server performance. You can configure the following:

- The number of POP, IMAP, and HTTP processes
- The number of POP, IMAP, and HTTP connections per process
- The number of threads per POP, IMAP, and HTTP process

For details on configuring POP, IMAP, and HTTP services, see Chapter 2, "Configuring POP, IMAP, and HTTP Services."

Number of POP, IMAP, and HTTP Processes. If you have a multiprocessor machine, you might want to configure multiple POP, IMAP, and HTTP processes to allow more connections to your server and perhaps less thread contention. There is a performance overhead, however, in allocating tasks among multiple processes and in switching from one process to another.

Number of POP, IMAP, and HTTP Connections Per Process. IMAP and HTTP connections are generally very efficient compared to POP connections. Each POP reconnection requires re-authentication of the user. In contrast, an IMAP connection requires only a single authentication because the connection remains open for the duration of the IMAP session (login to logout). An HTTP connection is short, but the user need not reauthenticate for each connection because multiple connections are allowed for each HTTP session (login to logout).

Scalability is most adversely affected, therefore, by the number of POP users "pinging" the server to see if they have new mail. Netscape Messaging Servers can service up to hundreds of user requests per second. Consequently, 10,000 POP users checking mail once a day is easier to support than 1000 POP users checking mail every second during the day.

Connected processes do occupy memory on the server system, however. Because IMAP connections are persistent, the more IMAP users there are connected to the server, the fewer resources there are for new connections. Depending on your platform, you will need at least one IMAP process per 5,000 simultaneous IMAP connections

Number of Threads per POP, IMAP, or HTTP Process. Having more simultaneously executing threads means that more client requests can be handled without delay, so that a greater number of clients can be serviced quickly. However, there is a performance overhead to dispatching among threads, so there is a practical limit to the number of threads the server can make use of.

Configuration of SMTP Services

This section summarizes SMTP configuration options that affect system performance. Some of these options can improve server performance; other options provide valuable features but can impact server performance adversely. If you are having problems with server performance for whatever reasons, you might want to consider whether you need the features provided by some of these options. SMTP configuration options that can affect server performance include:

- Limiting Dial-up Connections
- Limiting the Size of Messages
- Verifying Recipient Addresses
- Performing Reverse IP Address Lookups
- Specifying Alternate Search Methods
- Specifying "From" Address Rewrite Style
- Using the Plug-in API

Limiting Dial-up Connections

You can improve server performance by limiting the number of dial-up connections to your server. If both client (in this case another MTA) and server support the ETRN command, when the client connects to the server to send a message, it can also initiate processing of the deferred queue for the client's domain. This feature is useful for sites that are set up as secondary mail exchange (MX) hosts for other sites that only have a dial-up connection to the Internet. By enabling this command, you permit dial-up servers to request delivery of their mail. For more information, see "Enabling Requests for Deferred Queue Processing (ETRN)" on page 103.

Limiting the Size of Messages

The size of messages is a factor when considering the transmit time to actually "move" the message from the client to the server and the disk space available for storing messages. Users may negatively impact performance by sending extremely large documents over the network. You can limit the size of messages your server will accept. For more information, see "Limiting Message Size (SIZE)" on page 103.

Verifying Recipient Addresses

You can specify an option to verify recipient addresses when a client connects to the server. Specifying this option has slight performance impact because the server must perform an LDAP lookup for each recipient while connected to the client. The benefit is that the server can detect bad recipient names in the envelope address and return an error to the client before the client sends the body of the message. The client can fix the address before sending the message text. For more information, see "Verifying Recipient Addresses" on page 97.

Performing Reverse IP Address Lookups

If you enable this option, Messaging Server will use DNS to map the client's IP address to the associated host name. The host name is then used in the process table, the log files, and in "Received" lines in message headers. This option requires DNS lookups, however, so enabling this option can impact performance adversely if your server handles a large volume of messages. For more information, see "Performing Reverse IP Address Lookups" on page 98.

Specifying Alternate Search Methods

If you enable alternate search methods, you can expand the list of possible recipient matches. Be aware that enabling this option can impact server performance because of the increase in LDAP lookups. For more information, see "Specifying Alternate Search Methods" on page 110.

Specifying "From" Address Rewrite Style

Rewriting the "From:" address increases the odds that replies to outgoing messages are processed correctly. However, this feature does incur some performance overhead. For more information on this option, see "Specifying From Address Rewrite Style" on page 109.

Using the Plug-in API

If you are using the Messaging Server plug-in API, some CPU and memory resources will necessarily be diverted from the usual Messaging Server processing. However, you can use the plug-in API to provide valuable extensions to the Messaging Server capabilities. For more information, see Chapter 7, "Working with SMTP Plug-Ins."

Configuration of Logging Services

Logging options can be expensive in terms of server performance and response time. You should carefully consider your logging requirements and log only those messages that you need to successfully monitor and maintain your server. For more information on logging policies, see "Defining and Setting Logging Options" on page 313.

Size of Mailboxes

Users with mailboxes that contain an extremely large number of messages might encounter slow response times from the server. If so, you might want to set up user disk quotas or aging policies for user mailboxes. For more information about disk quotas and aging policies, see Chapter 5, "Managing the Message Store."

You might also want to consider distributing mailboxes across disks. Keep in mind that you cannot, however, distribute a single mailbox across partitions. See "Distributing the Store Across Disks" on page 292.

Distribution of the Store and Queue Directories

The `store` directory contains the user mailboxes. The `queue` directory is a temporary directory where all mail manipulation and rewriting take place. To improve disk access, the `store` directory and the `queue` directory should reside on separate disks. If you have a fast RAID setup (see "Use of RAID Technology" on page 295), you should keep the store and the queue on separate disks. You should also consider:

- Mounting the Queue Directory on a Fast File System and Fast Disk
- Distributing the Queue Across Disks
- Distributing the Store Across Disks

To ease management of multiple disks, you can use RAID (Redundant Array of Inexpensive Disks) technology to distribute data across a series of physical disks. The disks appear as one logical volume, so disk management is simplified. For more information, see "Use of RAID Technology" on page 295.

Mounting the Queue Directory on a Fast File System and Fast Disk

Because the message queue directory is the most heavily used part of the system, you should mount the queue directory on a very fast file system, such as the Veritas file system (VxFS). Mounting the queue directory on a fast file system can significantly increase performance for SMTP accept rates. By maintaining the queue directory in its own file system, you can also monitor message queue performance separately from message store performance.

You should also consider disk performance. For example, newer I/O buses like Ultra2 SCSI and Fibre Channel can transfer large quantities of data rapidly—up to 80 MB per second. The faster the communication speed between disk and CPU, the better Messaging Server will perform.

Distributing the Queue Across Disks

If you have more than one disk available, you might want to distribute the queue across disks. By distributing the queue, you can reduce the load associated with delivering a message because the server can perform concurrent I/O operations. You can also use multiple queue directories to reduce the overhead associated with large number of files accumulating in a single queue. For more information about specifying alternate queues, see "Specifying Alternate Paths for Physical Queues" on page 118.

Distributing the Store Across Disks

For improved mailbox access performance, you can distribute the message store across multiple disks. When distributing the message store, you should limit the number of mailboxes on any one disk.

Distributing mailboxes across disks typically does not change the SMTP access rate, but it will dramatically improve message delivery time. The number of mailboxes you allocate per disk depends on the following factors:

- Disk capacity
- Disk space allocated to each user

Note: Messaging Server 4.1 stores one copy of each message per file system. Therefore, if you have 5 partitions, each partition will contain a copy of the message. All other copies are created using file system hard links.

For more information about distributing the message store, see "Configuring Message Store Partitions" in Chapter 5.

MTA Thread Settings

You can use the `configutil` utility to modify the thread settings for various MTA components:

```
service.smtp.account-handler.minruncount
service.smtp.autoreply-handler.minruncount
service.smtp.error-handler.minruncount
service.smtp.mailbox-deliver.minruncount
service.smtp.prog-deliver.minruncount
service.smtp.smtp-accept.minruncount
service.smtp.smtp-deliver.minruncount
```

```
service.smtp.smtp-router.minruncount
service.smtp.unix-deliver.minruncount
```

For example, you might want to improve the SMTP acceptance rate and allow for more simultaneous inbound connections by increasing the number of threads allocated to the `service.smtp.smtp-accept` component as follows:

```
configutil -o service.smtp.smtp-accept.minruncount -v 75
```

As another example, you might want to improve the mailbox delivery rate by increasing the number of threads allocated to the `service.smtp.mailbox-deliver` component as follows:

```
configutil -o service.smtp.mailbox-deliver.minruncount -v 5
```

Be aware that threads use available memory and might generate more resource contention. So, depending on the hardware configuration of the server machine, too many threads can impede server performance. You should carefully measure the results of any tuning to make sure you are helping—not hurting—performance.

Applications Co-Resident with Messaging Server

Be aware that other applications running on the same system as Netscape Messaging Server will compete for the same system resources (memory, disk space, and so on) as Messaging Server. Depending on your messaging requirements, you might want to dedicate one or more machines to the messaging services.

Activity of Administration Server

If Netscape Administration Server is being used while Netscape Messaging Server is running, there will be less memory available for general use.

Activity of Directory Server

If Messaging Server is competing with numerous other LDAP clients (for example, other Netscape servers or a synchronization process with another LDAP based directory) for access to Directory Server, Messaging Server performance will be affected. Under large stress loads, you should replicate the directory services to more machines to increase accessibility of the service. You should also dedicate a directory machine to one or more Messaging Servers.

Location of Messaging Server and Directory Server

In general, for improved performance, you should keep Messaging Server and Directory Server on separate machines. Placing these servers on separate machines will prevent contention over system resources (CPU, RAM, and disk space).

If you do decide to place the servers on separate machines, you must also consider the network bandwidth between the two machines. For example, a 100 MB dedicated network should provide good performance. If possible, you should consider using full duplex on a switched/dedicated network to eliminate collisions.

Number of Address Lookups per Message

Be aware that if the average message is destined for a large number of recipients, there is a corresponding increase in the number of required LDAP lookups required to resolve the address. You might want to limit access to aliases that contain a large number of users; for example, you might not want all users to have access to the `companyall` alias.

Ratio of Local Delivery to Outbound Sends

Most customers have a higher number of messages received to messages sent per user (on the order of 4 to 1 or higher). Netscape messaging solutions are optimized for exactly these kind of environments. Because sending a message on the server is more “expensive” than receiving, any deviation from the expected ratio (for example, 1:1 sending and receiving) will affect the expected performance and scalability of your Messaging Server.

Use of RAID Technology

To ease management of multiple disks, you can use RAID (Redundant Array of Inexpensive Disks) technology to distribute data across a series of physical disks. With RAID technology, multiple disks appear as one logical volume, so disk management is simplified. You can have multiple logical volumes each consisting of several physical disks.

There are several levels of RAID technology. You should consider implementing a RAID-0+1 configuration. RAID 0 provides fast access to data through a technology called striping. A stripe is a disk segment varying in size from one sector up to several megabytes. Disk I/O is distributed across all stripes. RAID 1 implements disk mirroring for fault tolerance.

You can configure RAID technology by using dedicated RAID hardware or by using software RAID techniques such as those offered by the Veritas Volume Manager (VxVM).

Memory, Disk, and CPU Requirements

As stated before, a well-configured server cannot perform well on a poorly-tuned system. During the initial planning stage, you need to estimate your memory, disk, CPU, and network bandwidth requirements for your business environment. These estimates depend on many factors, including how many users the server supports, average number of messages per user, geographic location of users, and so on.

Of course, your estimates will depend on usage patterns and mail configuration. For example, if you are a host providing resources to residential consumers, your estimates will differ from the estimates required for corporate use. In general, residential consumers use much less resources than corporate consumers, so you would allocate less memory, disk space, and processing power for each residential user.

If your original estimates did not plan for growth, if business conditions change, or if performance requirements change, you might need to reconfigure your hardware. For example, you might need to add disks to the system, add processors, and so on. Although deployment planning is beyond the scope of this section, you can use the tools described in "System Monitoring Tools" on page 296 to monitor how your system is performing.

System Monitoring Tools

The following table lists system tools you can use to monitor your server environment. These tools are available on various Unix platforms. For more information about these tools, see the man pages delivered with your Unix system.

Table 10.4 General Unix tools

Tool	Description
`iostat`	Provides information about disk I/O and CPU usage.
`lsof`	Provides information about open file descriptors. (Available in source from :ftp://vic.cc.purdue.edu/pub/tools/unix.)
`lslk`	Provides information about file system locks. (Available in source from :ftp://vic.cc.purdue.edu/pub/tools/unix.)
`netstat`	Provides statistics about network functions.
`nslookup`	Allows you to query DNS servers for information about hosts and domains; for example you can print a list of hosts in a particular domain; also provides an IP address-to-host name mapping function (and vice versa).
`ping`	Allows you to query the status of a remote host or network gateway.

Table 10.4 General Unix tools (Continued)

Tool	Description
`sar`	Unix SysV performance monitoring tool. Useful for gathering system information over a longer period of time to use in long term planning, for example.
`tcpdump`	Public domain tools used in debugging and to monitor network traffic.
`top`	Provides quick, easy monitoring of processes and CPU activities. (This is a public domain tool that works on most Unix platforms.)
`trace`	Similar to `truss` on Solaris. Sometimes included by the vendor; otherwise, you can download this tool from an Internet site.
`traceroute`	Determines the path a packet takes throughout the Internet to reach its final destination.
`vmstat`	Provides statistics about process, virtual memory, disk, trap, and CPU activity.

Table 10.5 System monitoring tools - Sun Solaris

Tool	Description
`lockstat`	Provides information on OS and application locking. Available on Solaris 2.6 only.
`mpstat`	Provides statistics about each processor on the system
`pmap`	Provides breakdown on how much memory a process is using so you can see how much is shared and how much is private. (Located in `/usr/proc/bin`.)
`proctool`	Monitors processes and threads. (Available from Sun's web site.)
`snoop`	Monitors network traffic; indispensable when debugging low-level packets.

Table 10.5 System monitoring tools - Sun Solaris

Tool	Description
SymbEL/Virtual Adrian	A very powerful system monitoring toolkit. Provides the functionality of the above listed tools and more. Can be used to tune the `ncsize` and `ufs_ninod`e parameters and even has a mode to tune the operating system automatically.
`truss`	Provides information about which system calls a process makes.

Table 10.6 System monitoring tools - HP-UX

Tool	Description
`glance`	Provides detailed system information about open file descriptors, locks, threads, and so on.
`gpm`	Provides detailed system information about open file descriptors, locks, threads, and so on.
`tusc`	A system call trapper. Might not be available on all systems.
`sysdef`	Provides information about kernel parameters.
`landiag`	A tool for monitoring network statistics.
`sam`	System Administration Manager. A tool for general system administration.

Table 10.7 System monitoring tools - SGI Irix

Tool	Description
`dkstat`	Similar to `iostat`. Provides information about disk I/O and CPU use.
`gmemusage` (Irix 6.x)	X windows tool for viewing information about virtual memory.
`netstat -C`	Provides real-time, full-screen data.

Table 10.7 System monitoring tools - SGI Irix

Tool	Description
`osview`	Provides full-screen information; combines functionality of `vmstat`, `mpstat`, and `netstat`.
`par`	Similar to `truss` on Solaris. Provides information about system calls made by a process.
Performance Copilot	An SGI add-in package.

Using SNMP on Unix Platforms

This section describes SNMP on Unix platforms. If you are using SNMP on Windows NT, see your Windows NT documentation for information about SNMP.

Netscape Messaging Server supports the Simple Network Management Protocol (SNMP), version 1, and provides controls for configuring its SNMP subagent. Using SNMP and network management software, such as HP OpenView, you can monitor Messaging Server in real time as you do any other network device.

Under SNMP, data travels between a managed device and a network management station (NMS). A managed device is any device that runs SNMP; a server is a managed device. From the network management station, you can monitor the network remotely and exchange data between servers about network activity.

In Messaging Server, SNMP uses two agents to transfer information between the network management station and the managed device, the master agent and the subagent.

- The master agent handles all communication between servers, through their subagents, and the network management station. The master agent is installed with the Administration Server, and is controlled through its user interface. For more information, see *Managing Servers with Netscape Console*.

- The subagent for each server has access to its operation information, which the subagent sends to the master agent when requested. The Messaging Server subagent is installed with Messaging Server and is controlled by the SMTP Settings form. For more information, see "The Messaging Server Subagent" on page 301.

Each server stores operation information in the form of variables, or managed objects, which the network management station can query. The definitions for the server's managed objects are stored in the Management Information Base (MIB). For information about the Messaging Server MIB, see Appendix C, "SNMP MIB."

Note: For more information about SNMP capabilities in Netscape Server products, see *Managing Servers with Netscape Console*.

For detailed information about the SNMP protocol, see RFC 1155 and RFC 1157.

Communication Between the NMS and the Managed Device

To exchange network information, SNMP uses protocol data units (PDUs), which contain information about the variables stored on the server. The network management station (NMS) and the server, or managed device, exchange PDUs in either of two ways:

- In network management station-initiated communication, the management station queries the server for data. This accounts for most communication between the management station and a server. Messaging Server 4.1 supports querying server data only.

  In a network management station-initiated session, the management station identifies which servers and variables to monitor by examining the MIB. Then, the management station sends a PDU to the master agent, which passes it to the subagent of the server. The subagent returns the data to the master agent, which sends it back in PDU form to the management station.

- In managed-device-initiated communication, SNMP uses traps to send information about network events, such as shutdown and startup.

In a managed-device-initiated session, when a monitored event occurs, the subagent sends a trap to the master agent, which forwards it to the network management station. For information about traps, see "MIB Traps" on page 474.

Note: To send SNMP traps to the network management station, you must set the correct community and trap destination through the Administration Server. For more information, see *Managing Servers with Netscape Console*.

The Messaging Server Subagent

The Messaging Server's SNMP subagent, called `ns-mailagt`, gathers data about Messaging Server's activities. You can configure some parts of the subagent through Netscape Console.

Before you can enable the Messaging Server subagent, you must start the master agent through the Administration Server user interface. The master agent handles communication between servers, through their subagents, and the network management station. The subagents of all installed Netscape servers communicate with the same master agent. For information about the master agent, see *Managing Servers with Netscape Console*.

Once the master agent is available, you can start the subagent. The subagent does several things:

- It reads its configuration information, which is stored in the Directory Server. This is the information you can set by using Netscape Console.
- It initializes the Messaging Server MIB, which contains variables that store network information.
- It informs the master agent about the part of the MIB subtree that it monitors, and therefore what information it can provide. In this case, the Messaging Server subagent monitors, and can provide, only Messaging Server MIB data.

When the network management station sends a query, the master agent sends the request to the subagent through the SNMP Multiplexing (SMUX) protocol. The subagent queries the variables in the MIB and reports back to the master agent.

Periodically, the subagent checks the status of Messaging Server. If it detects that the server has shut down, restarted, or failed to respond, it sends a trap to the network management station through the master agent. When the Administration Server terminates the subagent, the master agent unregisters the MIB, while the subagent performs any cleanup or logging and then exits.

For information about the Messaging Server MIB, see Appendix C, "SNMP MIB." For information about the SMUX protocol, see RFC 1227.

Configuring SNMP

You use the Messaging Server SNMP tab, which is part of Netscape Console, to set some options for the subagent, enable server statistics collection, and start or stop the subagent. After you modify the subagent, you must restart it before your changes can take effect.

To view and configure information subagent options, go to the SNMP tab.

1. In the Messaging Server Console, select the server for which you are setting subagent options.
2. Under Server Group, double-click the Configuration tab.
3. In the window on the right side of the page, click the SNMP tab.

Using this tab, you can perform the following tasks:

- Configuring the Subagent
- Starting and Stopping the Subagent

To verify the changes you make using the SNMP tab, see "Verifying SNMP Configuration Changes" on page 305.

Configuring the Subagent

To configure the SNMP subagent on your system and enable statistics reporting to the network management station, you use the SNMP tab, which is part of Netscape Console.

Note: Before you can modify the subagent, the Messaging Server installation you want to configure and the master agent must be running. The master agent is enabled through the Administration Server's SNMP Master Agent Control form. For more information, see *Managing Servers with Netscape Console*.

Follow these steps to configure the subagent.

1. Go to the SNMP tab. For directions, see "Configuring SNMP" on page 302.

2. At the top of the SNMP Settings form, configure the fields as follows:

   Master agent hostname. In this field, enter the name of the host where the SNMP master agent resides. This must be a machine name, not an IP address. Default: `localhost`.

   Master agent port number. In his field, specify the port number the subagent uses to communicate with the master agent. Default: 199.

   Organization name. In this field, enter the name of the organization using Messaging Server. Usually this is a department or company name.

   Server Description. In this field, enter a text description that uniquely describes this Messaging Server installation, for example, the Messaging Server for Marketing.

   Contact person info. In this field, specify the person to contact about anything related to Messaging Server. Usually this is the server administrator. You can enter the name or email address of the contact, or both.

   Server Location. In this field, specify the location of this installation of Messaging Server. Usually this is a street address.

   The information in these fields is stored in Netscape Directory Server for this installation.

3. Click Save.

   A message appears reminding you to restart the subagent so that the settings can take effect. *Before you can restart the subagent*, you must enable statistics collection.

4. Go on to "Enabling Statistics Collection."

Enabling Statistics Collection

The subagent does not report SNMP statistics to the network management station unless you enable statistics collection on the SNMP Settings form, which is part of Netscape Console. If statistics collection is not enabled, the subagent cannot be started.

Important: If the network management station has problems getting Messaging Server's SNMP statistics, check the server log information, which is located in *`serverRoot`*`/mail-`*`instanceName`*`/log/default`, for information.

If the SNMP data collection process (`snmpcoll`), is not running, check the Administration Server Console to see whether the SNMP enable flag is on. In Messaging Server 3.x, this process starts when you start Messaging Server, whether SNMP is enabled or not. In Messaging Server 4.x, this flag applies to both the SNMP agent and `snmpcoll`. For more information, see *Managing Servers with Netscape Console*.

If you disable the startup server, this collection process is also disabled.

Follow these steps to enable data collection, after completing the steps described in "Configuring the Subagent" on page 302.

1. Check the Enable Statistics Collection box.

   If you remove the check, the subagent cannot be enabled.

2. Restart the subagent by clicking the Start button.

   Your configuration information is stored in Directory Server, the subagent starts, and statistics collection begins.

3. If you want to verify your changes to this form, see "Verifying SNMP Configuration Changes" on page 305.

Starting and Stopping the Subagent

To start, stop, or restart the SNMP subagent, select one of three options at the bottom of the SNMP Settings form.

To start, stop, or restart the subagent:

1. Go to the SNMP tab. For directions, see "Configuring SNMP" on page 302.
2. At the bottom of the form, click one of these buttons:

   Start. Messaging Server attempts to start the subagent.

   Stop. Messaging Server attempts to stop the subagent, if it is currently running.

   Restart. Messaging Server attempts to stop and then restart the subagent.

Note: If the SNMP subagent fails to start or stop, check the SNMP subagent log, which is located in *serverRoot*/mail-*instanceName*/log/default, for information.

Verifying SNMP Configuration Changes

You can verify the changes you make using the SNMP Settings form by entering a Messaging command in a command window.

Follow these steps to verify your changes to the subagent.

1. Open a Unix command window.
2. Go to the Messaging Server `bin` directory.
3. Enter the Messaging command `configutil`. For example:

   ```
   configutil | more
   ```

   You see all Messaging Server configuration statistics.
4. Scroll to the statistics whose names start with SNMP and find the settings you can configure using the SNMP Settings form:

   ```
   SNMP.master port | 199
   SNMP.contact
   SNMP.description
   SNMP.location
   ```

Chapter

11

Logging and Log Analysis

Netscape Messaging Server can create log files that record events related to its administration, to communications using any of the protocols (IMAP, POP, SMTP, and HTTP) that the server supports, and to other processes employed by the server. By examining the log files, you can monitor many aspects of the server's operation.

You can customize the policies for creating and managing the Messaging Server log files. This chapter describes the types and structure of log files, and discusses how to administer and how to view the log files.

This chapter has the following sections:

- Log Characteristics
- Defining and Setting Logging Options
- Searching and Viewing Logs
- Analyzing Logs with Third-Party Tools
- Selected Event-Message Formats

Log Characteristics

Messaging Server logging is flexible and customizable. You can specify settings that affect which and how many events are logged, and you can use those settings and other characteristics to refine searches for logged events when you are analyzing log files.

Services That Are Logged

Messaging Server creates a separate set of log files for each of the major protocols, or services, it supports. You can customize and view each type of log file individually. Table 11.1 lists the services that can be logged, and describes the log files for each service.

Table 11.1 Logged services

Service	Log-file description
Admin	Contains logged events related to communication between Netscape Console and Messaging Server (mostly through several CGI processes), by way of its Administration Server
SMTP	Contains logged events related to SMTP activity of this server
IMAP	Contains logged events related to IMAP4 activity of this server
POP	Contains logged events related to POP3 activity of this server
HTTP	Contains logged events related to HTTP activity of this server
Default	Contains logged events related to other activity of this server, such as command-line utilities and other processes

Levels of Logging

The level, or priority, of logging defines how detailed, or verbose, the logging activity is to be. A higher priority level means less detail; it means that only events of high priority (high severity) are logged. A lower level means greater detail; it means that more kinds of events are recorded in the log file.

You can set the logging level separately for each service, and you can use logging level to filter searches for log events. Table 11.2 describes the available levels.

Table 11.2 Levels of Logging

Level	Description
Critical	The minimum logging detail. An event is written to the log whenever a severe problem or critical condition occurs—such as when the server cannot access a mailbox or a library needed for it to run.
Error	An event is written to the log whenever an error condition occurs—such as when a connection attempt to a client or another server fails.
Warning	An event is written to the log whenever a warning condition occurs—such as when the server cannot understand a communication sent to it by a client.
Notice	An event is written to the log whenever a notice (a normal but significant condition) occurs—such as when a user login fails or when a session closes.
Informational	An event is written to the log with every significant action that takes place—such as when a user successfully logs on or off or creates or renames a mailbox.
Debugging	The most verbose logging. Useful only for debugging purposes. Events are written to the log at individual steps within each process or task, to pinpoint problems.

Note: These Messaging-Server logging levels are a subset of those defined by the Unix `syslog` facility.

Note: The more verbose the logging you specify, the more disk space your log files will occupy; for guidelines, see "Defining and Setting Logging Options" on page 313.

When you select a particular logging level, events corresponding to that level and to all higher (less verbose) levels are logged. The default level of logging is Notice.

Facilities as Categories of Logged Events

Within each supported service or protocol, Messaging Server further categorizes logged events by the facility, or functional area, in which they occur. Every logged event contains the name of the facility that generated it. These categories aid in filtering events during searches. Table 11.3 lists the facilities that Messaging Server recognizes for logging purposes.

Table 11.3 Facilities in which log events occur

Facility	Description
General	Undifferentiated actions related to this protocol or service
LDAP	Actions related to Messaging Server accessing the LDAP directory database
Network	Actions related to network connections (socket errors fall into this category)
Account	Actions related to user accounts (user logins fall into this category)
Protocol	Protocol-level actions related to protocol-specific commands (errors returned by IMAP or POP functions fall into this category)
Stats	Actions related to the gathering of server statistics
Store	Low-level actions related to accessing the message store (read/write errors fall into this category)

For examples of using facility categories as filters in log searches, see "Searching and Viewing Logs" on page 318.

Filename Conventions for Log Files

All log files created by Messaging Server use identical naming conventions. Each log file has a filename of the form:

service.*sequenceNum*.*timeStamp*

where the components of the filename have these definitions:

Table 11.4 Log filename conventions

Component	Definition
service	The protocol or service being logged (see Table 11.1).
sequenceNum	An integer that specifies the order of creation of this log file compared to others in the log-file directory. Log files with higher sequence numbers are more recent than those with lower numbers. Sequence numbers do not roll over; they increase monotonically for the life of the server (beginning at server installation).
timeStamp	A large integer that specifies the date and time of file creation. (Its value is expressed in standard Unix time: the number of seconds since midnight January 1, 1970.)

For example, a log file named `imap.63.915107696` would be the 63rd log file created in the directory of IMAP log files, created at 12:34:56 PM on December 31, 1998.

The combination of open-ended sequence numbering with a timestamp gives you more flexibility in rotating, expiring, and selecting files for analyzing. For more specific suggestions, see “Defining and Setting Logging Options” on page 313.

Content Format for Log Files

All log files created by Messaging Server have identical content formats. Log files are multiline text files, in which each line describes one logged event. All event descriptions, for each of the supported services, have the general format:

dateTime hostName processName[*pid*]: *facility logLevel*: *eventMessage*

in which the components of the event description have these definitions:

Table 11.5 Log file components

Component	Definition
dateTime	The date and time at which the event was logged, expressed in *dd/mon/yyyy hh*:*mm*:*ss* format, with a time-zone field expressed as +/-*hhmm* from GMT. For example: :02/Jan/1999:13:08:21 -0700
hostName	The name of the host machine on which the server is running: for example, showshoe. **Note:** If there is more than one instance of Messaging Server on the host, you can use the process ID (pid) to separate logged events of one instance from another.
processName	The name of the process that generated the event: for example, cgi_store.
pid	The process ID of the process that generated the event: for example, 18753.
facility	The facility category that the event belongs to: for example, General (see Table 11.3 on page 310).
logLevel	The level of logging that the event represents: for example, Notice (see Table 11.2 on page 309).
eventMessage	An event-specific explanatory message that may be of any length: for example, Log created (894305624). For descriptions of the formats of some event messages, see "Selected Event-Message Formats" on page 321.

Note: This format of event descriptions is identical to that defined by the Unix syslog facility, except that the date/time format is different and the format includes two additional components (*facility* and *logLevel*).

Here are three examples of logged events as viewed using Netscape Console:

```
02/May/1998:17:37:32 -0700 showshoe cgi_store[18753]: General Notice:
  Log created (894155852)

04/May/1998:11:07:44 -0400 xyzmail cgi_service[343]: General Error:
  function=getserverhello|port=2500|error=failed to connect

03/Dec/1998:06:54:32 +0200 AiriusPost imapd[232]: Account Notice:
  close [127.0.0.1] [unauthenticated] 1998/12/3 6:54:32 0:00:00 0 115 0
```

When viewing a log file in the Log Viewer window, you can limit the events displayed by searching for any specific component in an event, such as a specific logging level or facility, or a specific process ID. For more information, see "Searching and Viewing Logs" on page 318.

Log-File Directories

Every logged service is assigned a single directory, in which its log files are stored. All IMAP log files are stored together, as are all POP log files, and log files of any other service. You define the location of each directory, and you also define how many log files of what maximum size are permitted to exist in the directory.

Make sure that your storage capacity is sufficient for all your log files. Log data can be voluminous, especially at lower (more verbose) logging levels.

It is important also to define your logging level, log rotation, log expiration, and server-backup policies appropriately so that all of your log-file directories are backed up and none of them become overloaded; otherwise, you may lose information. See "Defining and Setting Logging Options" (next).

Defining and Setting Logging Options

You can define the logging configurations for Messaging Server that best serve your administration needs. This section discusses issues that may help you decide on the best configurations and policies, and it explains how to implement them.

Flexible Logging Architecture

The naming scheme for log files (*service.sequenceNum.timeStamp*) helps you to design a flexible log-rotation and backup policy. The fact that events for different services are written to different files makes it easier for you to isolate problems quickly. Also, because the sequence number in a filename is ever-increasing and the timestamp is always unique, later log files do not simply overwrite earlier ones after a limited set of sequence numbers is exhausted. Instead, older log files are overwritten or deleted only when the more flexible limits of age, number of files, or total storage are reached.

Messaging Server supports automatic rotation of log files, which simplifies administration and facilitates backups. You are not required to manually retire the current log file and create a new one to hold subsequent logged events. You can back up all but the current log file in a directory at any time, without stopping the server or manually notifying the server to start a new log file.

In setting up your logging policies, you can set options (for each service) that control limits on total log storage, maximum number of log files, individual file size, maximum file age, and rate of log-file rotation.

Planning the Options You Want

Keep in mind that you must set several limits, more than one of which might cause the rotation or deletion of a log file. Whichever limit is reached first is the controlling one. For example, if your maximum log-file size is 3.5 MB, and you specify that a new log be created every day, you may actually get log files created faster than one per day if log data builds up faster than 3.5 MB every 24 hours. Then, if your maximum number of log files is 10 and your maximum age is 8 days, you may never reach the age limit on log files because the faster log rotation may mean that 10 files will have been created in less than 8 days.

The following default values, provided for Messaging Server administration logs, may be a reasonable starting point for planning:

Maximum number of log files in directory: 10
Maximum log-file size: 2 MB
Total maximum size permitted for all log files: 20 MB
Minimum free disk space permitted: 5 MB

Log rollover time: 1 day
Maximum age before expiration: 7 days
Level of logging: Notice

You can see that this configuration assumes that server-administration log data is predicted to accumulate at about 2 MB per day, backups are weekly, and the total space allotted for storage of admin logs is at least 25 MB. (These settings may be insufficient if the logging level is more verbose.)

For SMTP, POP, IMAP or HTTP logs, the same values might be a reasonable start. If all services have approximately the same log-storage requirements as the defaults shown here, you might expect to initially plan for about 150 MB of total log-storage capacity. (Note that this is meant only as a general indication of storage requirements; your actual requirements may be significantly different.)

Setting Logging Options

You can use Netscape Console to set options that control the logging configuration for each Messaging Server service.

The optimal settings for these options depend on the rate at which log data accumulates. It may take between 4,000 and 10,000 log entries to occupy 1 MB of storage. At the more verbose levels of logging (such as Notice), a moderately busy server may generate hundreds of megabytes of log data per week. Here is one approach you can follow:

1. In Netscape Console, open the Messaging Server whose log file options you want to set.

2. Click the Configuration tab, open the Log Files folder in the left pane, and select the log files of a service (such as IMAP, SMTP, HTTP, or Admin).

3. From the "Levels of detail" drop-down list, choose a logging level.

   Set a level of logging that is consistent with your storage limits—that is, a level that you estimate will cause log-data accumulation at approximately the rate you used to estimate the storage limit.

4. In the "Directory path for log files" field, enter the name of the directory to hold your log files.

5. In the "File size for each log" field, enter your maximum log-file size.

   Define the log file size so that searching performance is not impacted. Also, coordinate it with your rotation schedule and your total storage limit. Given the rate at which log entries accumulate, you might set a maximum that is slightly larger than what you expect to accumulate by the time a rotation automatically occurs. And your maximum file size times your maximum number of files might be roughly equivalent to your total storage limit.

   Example: If your IMAP log rotation is daily, your expected accumulation of IMAP log data is 3 MB per day, and your total storage limit for IMAP logs is 25 MB, you might set a maximum IMAP log-file size of 3.5 MB. (In this example, you could still lose some log data if it accumulated so rapidly that all log files hit maximum size and the maximum number of log files were reached.)

6. In the "Create new log every" field, enter a number for the log-rotation schedule.

7. In the "Number of logs per directory" and the "When a log is older than" fields, enter the maximum number of log files and a maximum age to coordinate with your backup schedule.

   Example: If server backups are weekly and you rotate IMAP log files daily, you might specify a maximum number of IMAP log files of about 10 (to account for faster rotation if the individual log-size limit is exceeded), and a maximum age of 7 or 8 days.

8. In the "When total log size exceeds" field, enter the total storage limit you want.

   Pick a total storage limit that is within your hardware capacity and that coordinates with the backup schedule you have planned for the server. Estimate the rate at which you anticipate that log data will accumulate, add a factor of safety, and define your total storage limit so that it is not exceeded over the period between server backups.

   Example: If you expect to accumulate an average of 3 MB of IMAP log-file data per day, and server backups are weekly, you might specify on the order of 25 - 30 MB as the storage limit for IMAP logs (assuming that your disk storage capacity is sufficient).

9. In the "When free disk space is less than" field, enter the minimum amount of free disk space you want to reserve.

For safety, pick a minimum amount free disk space that you will permit on the volume that holds the log files. That way, if factors other than log-file size cause the volume to fill up, old log files will be deleted before a failure occurs from attempting to write log data to a full disk.

Command Line

You can also set logging options at the command line as follows.

To set the logging level:

```
configutil -o logfile.service.loglevel -v level
```

where *service* is `admin`, `smtp`, `pop`, `imap`, or `http` and *loglevel* is `Nolog`, `Critical`, `Error, Warning`, `Notice`, `Information`, or `Debug.`

To specify a directory path for log files:

```
configutil -o logfile.service.logdir -v dirpath
```

To specify a maximum file size for each log:

```
configutil -o logfile.service.maxlogfilesize -v size
```

where *size* specifies a number of bytes.

To specify a log rotation schedule:

```
configutil -o logfile.service.rollovertime -v number
```

where *number* specifies a number of seconds.

To specify a maximum number of log files per directory:

```
configutil -o logfile.service.maxlogfiles -v number
```

To specify a storage limit:

```
configutil -o logfile.service.maxlogsize -v number
```

where *number* specifies a number in bytes.

To specify the a minimum amount of free disk space you want to reserve:

```
configutil -o logfile.service.minfreediskspace -v number
```

where *number* specifies a number in bytes.

To specify an age for logs at which they will expire:

```
configutil -o logfile.service.expirytime -v number
```

where *number* specifies a number in seconds.

Searching and Viewing Logs

Netscape Console provides a basic interface for viewing Messaging Server log data. It allows for selecting individual log files and for performing flexible filtered searches of log entries within those files.

For a given service (such as SMTP), log files are listed in chronological order. Once you have chosen a log file to search, you can narrow the search for individual events by specifying search parameters.

Search Parameters

These are the search parameters you can specify for viewing log data:

- **A time period**. You can specify the beginning and end of a specific time period to retrieve events from, or you can specify a number of days (before the present) to search. You might typically specify a range to look at logged events leading up to a server crash or other occurrence whose time you know of. Alternatively, you might specify a day range to look at only today's events in the current log file.
- **A level of logging**. You can specify the logging level (see "Levels of Logging" on page 309). You might select a specific level to uncover a specific problem; for example, Critical to see why the server went down, or Error to locate failed protocol calls.
- **A facility.** You can specify the facility (see "Facilities as Categories of Logged Events" on page 310). You might select a specific facility if you know the functional area that contains the problem; for example, Store if you believe a server crash involved a disk error, or Protocol if the problem lies in an IMAP protocol command error.

- **A text search pattern.** You can provide a text search pattern to further narrow the search. You can include any component of the event (see "Content Format for Log Files" on page 312) that can be expressed in a wildcard-type search, such as event time, process name, process ID, and any part of the event message (such as remote host name, function name, error number, and so on) that you know defines the event or events you want to retrieve.

  Your search pattern can include the following special and wildcard characters:

  `*` Any set of characters (example: `*.com`)
  `?` Any single character (example: `199?`)
  `[`*`nnn`*`]` Any character in the set *nnn* (example: `[aeiou]`)
  `[^`*`nnn`*`]` Any character not in the set *nnn* (example: `[^aeiou]`)
  `[`*`n-m`*`]` Any character in the range *n-m* (example: `[A-Z]`)
  `[^`*`n-m`*`]` Any character not in the range *n-m* (example: `[^0-9]`)
  `\` Escape character: place before `*`, `?`, `[`, or `]` to use them as literals

Note: Searches are case-sensitive.

Examples of combining logging level and facility in viewing logs might include the following:

- Specifying Account facility (and Notice level) to display failed logins, which may be useful when investigating potential security breaches
- Specifying Network facility (and all logging levels) to investigate connection problems
- Specifying all facilities (and Critical logging level) to look for basic problems in the functioning of the server

Specifying a Search and Viewing Results

Follow these steps to search for logged events with specific characteristics belonging to a given service:

1. In Netscape Console, open the Messaging Server whose log files you want to inspect.
2. Follow either of these steps to display the Log Files Content tab for a given logged service:

- Click the Tasks tab, then click "View *service* logs", where *service* is the name of the logged service (such as "IMAP service", "SMTP service", or "administration").
- Click the Configuration tab, then open the Log Files folder in the left pane and select the log files of a service (such as IMAP, SMTP, or Admin). Then click the Content tab in the right pane.

3. The Content tab for that logged service is displayed.
4. In the Log filename field, select the log file you want to examine.
5. Click the View selected log button to open the Log Viewer window.
6. In the Log Viewer window, specify your desired search parameters (described in the previous section, "Search Parameters").
7. Click Update to perform the search and display the results in the Log entry field.

Analyzing Logs with Third-Party Tools

For log analyses and report generation beyond the display capabilities of Netscape Console, you need to use other tools. You can manipulate log files on your own with text editors or standard system tools.

With a scriptable text editor supporting regular-expression parsing, you can potentially search for and extract log entries based on any of the criteria discussed in this chapter, and possibly sort the results or even generate sums or other statistics.

In Unix environments you might also be able to modify and use existing report-generation tools that were developed to manipulate Unix `syslog` files. If you wish to use a public-domain `syslog` manipulation tool, remember that you may need to modify it to account for the different date/time format and for the two extra components (*`facility`* and *`logLevel`*) that appear in Messaging Server log entries but not in `syslog` entries.

Selected Event-Message Formats

The event message of each log entry is in a format specific to the type of event being logged: that is, each service defines what content appears in any of its event messages. Many event messages are simple and self-evident; others are more complex.

To help you search for and interpret common log entries related to message transfer, this section describes the format of logged events written by three modules of the SMTP service: SMTP-Accept, SMTP-Deliver, and Mailbox-Deliver.

The log-entry elements described here are all parts of the *eventMessage* component of the log entry, where the entire entry has the format:

dateTime hostName processName[*pid*]: *facility logLevel*: *eventMessage*

For descriptions of the other components, see "Content Format for Log Files" on page 312.

SMTP-Accept Log Format

The event message for an SMTP-Accept log entry has the format

moduleName:*envelopeID*:*mailFrom*:[*peerAddr*ess]:*peerHost*:*msgID*:*msgSize*:
numRecipients:*recipientList*

where the components of the event message have the following definitions:

Table 11.6 SMTP-Accept event message elements

Component	Definition
moduleName	The name of the SMTP module that logged the event (SMTP-Accept)
envelopeID	The ID assigned to the message by Messaging Server (unique to each received message)
mailFrom	The sender's address, from the message envelope
peerAddress	The IP address of the connecting server
peerHost	The host name (or IP address, if no lookup is performed) of the connecting server

Table 11.6 SMTP-Accept event message elements (Continued)

Component	Definition
msgID	The ID of the message, written by the sending client into the message header
msgSize	The size of the message, in bytes
numRecipients	The number of recipients
recipientList	The address of each recipient

Here is an example of an SMTP-Accept log entry:

```
[08/Sep/1998:19:04:24 -0700] dizzy smtpd[8379]: General Notice:
   SMTP-Accept:0EYZV320.6U1:<aswe32dasdf@netscape.com>:[127.0.0.1]:
   127.0.0.1:<pkeni@netscape.com>:272:1:<dizzy2@dizzy.mcom.com>
```

SMTP-Deliver Log Format

The event message for an SMTP-Deliver log entry has the format:

moduleName:*envelopeID*:*mailFrom*:*status*:*destHost*:*msgID*:*msgSize*:
numRecipients:*recipientList*

in which the components of the event message have the following definitions:

Table 11.7 SMTP-Deliver event message formats

Component	Definition
moduleName	The name of the SMTP module that logged the event (`SMTP-Deliver`)
envelopeID	The ID assigned to the message by Messaging Server (unique to each received message)
mailFrom	The sender's address, from the message envelope
status	The delivery status of the message (`Delivered` or `Deferred`)
destHost	The host name of the destination server
msgID	The ID of the message, written by the sending client into the message header

Table 11.7 SMTP-Deliver event message formats (Continued)

Component	Definition
msgSize	The size of the message, in bytes
numRecipients	The number of recipients
recipientList	The address of each recipient

Here is an example of an SMTP-Deliver log entry:

```
[08/Sep/1998:19:04:02 -0700] dizzy smtpd[8379]: General Notice:
   SMTP-Deliver:0EYZV2Q0.8C0:<aasdfasdfds@netscape.com>:Delivered:
   c3po.netscape.com:<pkeni@netscape.com>:337:1:<pkeni@netscape.com>
```

Mailbox-Deliver Log Format

The event message for a Mailbox-Deliver log entry has the format:

moduleName:*envelopeID*:*msgSize*:*msgID*:*userID*

where the components of the event message have the following definitions:

Table 11.8 Mailbox-Deliver event message elements

Component	Definition
moduleName	The name of the SMTP module that logged the event (`Mailbox-Deliver`)
envelopeID	The ID assigned to the message by Messaging Server (unique to each received message)
msgSize	The size of the message, in bytes
msgID	The ID of the message, written by the sending client into the message header
userID	The account name of the recipient to whom the message was delivered

Here is an example of a Mailbox-Deliver log entry:

```
[31/Jul/1998:16:50:56 -0700] slug smtpd[19530]: General Notice:
   Mailbox-Deliver:0EWZGWV0.02Z:17943:<12345678.123@nowhere>:slug464
```

Chapter

12

Program Delivery

This appendix explains how to set up Netscape Messaging Server to deliver incoming messages to external programs. Because program delivery has significant system security implications, administrators should carefully review and thoroughly understand the security implications before enabling program delivery.

This appendix discusses the following topics:

- About Program Delivery
- Security Considerations
- Enabling the Program Delivery Module
- Using Program Delivery to Handle Incoming Mail
- Program Delivery in Unix Environments
- Program Delivery in NT Environments

About Program Delivery

This section gives general overview information about the Netscape Messaging Server program delivery feature.

In this section, the term *program* refers to:

- **Unix**. Any executable file (including scripts).
- **NT**. Any Windows application. For example, any file with the filename extension of `.exe`, `.com`, or `.cmd`.

By default, incoming messages are put in the inbox of the mail account the message is addressed to. Accounts can be configured to perform various operations with the messages it receives, for example putting incoming messages in particular mail folders, forwarding them somewhere else, or generating an automatic response.

To accommodate the needs of advanced users who want more sophisticated control over the handling of their mail, Netscape Messaging Server offers the ability to deliver mail to external programs that can carry out these additional tasks. This is called *program delivery*. For example:

- **Program delivery can be used to help sort mail.** If you receive a great deal of mail, you might consider using a mail filter that sorts and delivers incoming mail to one or more folders. An automatic filter can usually sort messages based on the sender or topic of the message. With this type of program delivery, messages are delivered to the filtering program as they arrive.
- **Program delivery can also be used as a mail file server.** Some sites have a lot of information that they wish to make publicly available. The most common way to share files on the Internet is to make them available through the File Transfer Protocol (FTP) or the World Wide Web (WWW). But not everyone has FTP or web access. You can make files available to those who only have mail with a file server that selects and mails documents in response to mail requests.

  A request sent to a typical mail file server consists of one or more commands such as this:

  ```
  SEND /documents/internet/rfc/rtc822.txt
  ```

When you or a user specifies program delivery as an account option, as described in “Using Program Delivery to Handle Incoming Mail” on page 331, one or more programs are run whenever mail addressed to that account is received. Messaging Server starts the program and the mail is handed over to it. The program then performs whatever it is designed to do with incoming messages.

As described in the following sections, an administrator must first enable program delivery on Messaging Server. Once program delivery is enabled, users can select one or more programs to be run when messages addressed to their account are received.

Program Delivery and Mailbox Delivery

Program delivery is independent of, and separate from, delivery of messages to the account's mailbox. An account can use one or both. For example, if both POP/IMAP delivery and program delivery are selected, then incoming messages are delivered to the mailbox and also processed by program delivery.

Program Delivery Failures

If an incoming message is addressed to an account using program delivery, and for some reason program delivery fails, the incoming message is returned to the sender and a "delivery failed" type error message is generated. This might occur, for example, if a particular program is designated to handle incoming messages but the program delivery module cannot find that program because it has been moved or deleted from the directory where program delivery expects to find it.

Because program delivery and mailbox delivery are two separate and distinct operations, messages will be bounced back to senders if one fails even though the other succeeds. For example, if an account is using both program delivery and mailbox delivery, and program delivery fails, an incoming message will be bounced back to the sender with an "undeliverable" notice even though a copy of the message may be successfully delivered to the account's mailbox.

Security Considerations

Because users can specify that program delivery automatically execute one or more programs in response to incoming messages, program delivery could compromise system security if not properly administered. For this reason, program delivery is disabled by default and must be explicitly turned on by an administrator as explained in "Enabling the Program Delivery Module" on page 330.

Trusted Programs and Directory

A *trusted* program is one that is assumed to function properly when used with program delivery. Before designating a program as *trusted*, you should carefully inspect it to make sure that program delivery can automatically run it without risking system problems or reducing network security.

The program delivery module looks for trusted programs in a special directory known as the *trusted directory*. Any program or executable file stored in the trusted directory is assumed to be a trusted program. In other words, you designate a program as trusted by storing it in the trusted directory.

The location of the trusted directory cannot be changed. The location of the trusted directory is:

- **Unix**: *server-root*/msg-*instance*/smtp-bin/delivery
- **NT**: *server-root*\msg-*instance*\smtp-bin\delivery

As an administrator, you must ensure that each trusted program is well understood and known to be safe before placing it in the trusted directory. Make sure that every program stored in the trusted directory does nothing that will compromise security. When examining programs for security problems, keep in mind that system security involves more than just keeping messages and data out of unauthorized hands. An innocent mistake in a poorly written or configured program can cause serious system problems.

When running in secure mode, the program delivery module ignores any path specified in the user's account and only runs programs stored in the trusted directory. This allows an administrator to determine the exact executable files the program delivery feature will run.

For example, if Windows NT users want to set up a program delivery application that notifies them when new mail arrives, they can specify it for their account as:

```
\bin\new_mail.exe
```

or simply as

```
new_mail.exe
```

Regardless of how users specify `new_mail.exe`, the program delivery module will only execute the trusted version of `new_mail.exe` that is stored in the trusted program directory. If there is no version of `new_mail.exe` in the trusted directory, program delivery exits with an error message to the mail administrator.

Trusted Directory and Operating Modes

- **NT**. In NT environments, program delivery will only run in secure mode. This means that it will only run programs stored in the trusted directory.
- **Unix**. In Unix environments, program delivery can run in one of two operating modes: *secure* and *non-secure*. In secure mode, program delivery will only run programs stored in the trusted directory. In non-secure mode, program delivery can run programs stored anywhere on the network. For details, see "Secure and Non-secure Modes (Unix)" on page 338.

Guarding the Trusted Directory

By default, only administrators with `root` access (Unix), or administrator privileges (NT), can add or change programs in the trusted directory. Netscape recommends that this protection be maintained, and that the permissions for this directory never be relaxed to allow anyone else to add or modify programs in the trusted directory.

In regards to program delivery, the most important aspect of security is preventing unauthorized access to the trusted directory. Because program delivery assumes that any program stored in the trusted directory is secure and safe, it is essential that unauthorized persons are prevented from adding or modifying files in the trusted directory.

Scripts and Batch Files

Unix Environments

In Unix environments, scripts and batch files can be used by program delivery, but extra care should be taken to ensure that they are safe.

Scripts and batch files can run programs that are not stored in the trusted directory. If you use a script or batch file for program delivery, and it calls commands or programs that have not been inspected for safety and stored in the trusted directory, you run the risk of someone substituting or changing that command or program to detrimental effect.

Programs that interpret their input as a sequence of commands to execute (such as `sh`, `tcsh`, or `perl`) should not be used as trusted programs. However, some scripts that run under such programs can be considered safe after careful inspection. For example, it is risky to set up `perl` as a trusted program, but a carefully inspected `perl` script might be safe to use.

NT Environments

In Windows NT environments, Netscape recommends that you do not use scripts or batch files for program delivery.

Enabling the Program Delivery Module

For security reasons, program delivery is disabled by default and must be explicitly turned on by an administrator.

- **Disabled**. If the trusted directory is empty, the program delivery module is disabled and no one can use programs to process incoming mail.
- **Enabled**. If one or more files are stored in the trusted directory, the program module is enabled. Programs can be designated to process incoming mail if properly set up as described in “Setting Up Program Delivery (Unix)” on page 340 and “Setting Up Program Delivery (NT)” on page 344.

In both Unix and Windows NT environments, you enable the program delivery module by placing one or more programs in the trusted directory.

In Unix environments, there are two additional ways of enabling the program delivery module other than placing an executable file in the trusted directory:

- Store a link to an executable file in the trusted directory.
- Store a non-executable file named `INSECURE-PROGRAM-DELIVERIES` in the trusted directory. Note, however, that the presence of this file causes program delivery to run in non-secure mode which significantly increases security risks. For details, see "Secure and Non-secure Modes (Unix)" on page 338.

Using Program Delivery to Handle Incoming Mail

This section describes how to designate one or more software programs to process incoming messages for an account.

To designate programs to handle incoming mail, the program delivery module must first be set up and enabled as described in "Setting Up Program Delivery (Unix)" on page 340 and "Setting Up Program Delivery (NT)" on page 344.

Once program delivery is set up and enabled:

- An administrator can designate one or more programs to handle incoming mail for any account.
- An account owner can designate programs to process incoming mail for the account.

Administrators

Administrators can specify program delivery for any account. This can be done through the Create User tab at the time the mail account is created, or through the Edit User tab for an account that already exists.

1. Before establishing program delivery for a user:

   Make sure program delivery has been enabled as explained in "Enabling the Program Delivery Module" on page 330.

   Make sure the programs to be used for this account have been inspected for safety and placed in the trusted directory. (In Unix environments, you can choose to run program delivery in non-secure mode. In that case, the programs do not have to be stored in the trusted directory.)

2. Go to the Create User or Edit User tab.

3. Choose Mail from the menu and click on the Delivery tab.

4. Check the box labeled "Program delivery." The Properties button is activated.

   POP/IMAP delivery. If the "Enable POP/IMAP delivery" box is *also* checked, mail will continue to be delivered to the mailbox regardless of program delivery. In other words, if both the "Program delivery" and the "POP/IMAP delivery" boxes are checked, incoming mail will be processed by program delivery and *also* delivered to the mailbox.

   Unix delivery. The "Unix delivery" box has nothing to do with program delivery. Do not check this box simply because you are operating in a Unix environment. For information on when and why to check the "Unix delivery" box, see "Configuring Delivery Options" on page 130 in Chapter 4.

5. Click the Properties button next to the Program Delivery option. The Program Delivery dialog box is displayed.

6. In the Program Delivery dialog box, enter the command (program) that is to process incoming messages for this account.

   Unix secure mode. When running program delivery in secure mode, you need only enter the command name, you do not need to enter a path. For example, to run the program named `mymail` stored in the trusted directory, you enter `mymail`.

Unix non-secure mode. When running program delivery in non-secure mode, you must enter an absolute path for the program to run. (Program delivery does not make any use of paths in the account owner's environment.) For example, to run the program named `mymail` stored in the `/usr/bin` directory, you enter `/usr/bin/mymail`.

NT. You must enter the filename exactly as the filename exists in the trusted directory (including the filename extension). You do not need to enter a path. For example, to run the `mymail.exe` stored in the trusted directory you enter `mymail.exe`. program delivery.

7. To run multiple programs, enter each program on a separate line by itself. Programs will be run in the order you specify. For example, to first run the `new_mail.exe` program, and then the `sort_mail.cmd` program, simply enter:

```
new_mail.exe
sort_mail.cmd
```

8. Now click on OK.

Note: Messaging Server will allow you to enter the name of a program that has not been placed in the trusted directory (or a program with an incorrect path if running in Unix non-secure mode). But program delivery will fail when it tries to use that program. At that point a "delivery failed" type error message will be sent to the mail administrator and the incoming message bounced back to the sender with an "undeliverable" type notice.

To stop using program delivery to handle incoming messages for this account, simply delete the programs from the dialog box and uncheck the Program Delivery box.

To change the programs that program delivery runs for this account, simply add, delete, or change the programs listed in the dialog box.

Users and Account Owners

End users can designate program delivery for their accounts through the end user Server Account Management forms. To designate one or more programs to handle incoming mail for their accounts, end users should follow these steps:

1. Check with the appropriate administrator to:

   - Make sure program delivery has been enabled for Messaging Server.
   - Find out if the program (or programs) they want to use have been placed in the trusted directory. (In Unix environments, this may not be necessary if you have set up program delivery to use any program, not just those stored in the trusted directory.)
   - Obtain the machine name and port number of the Administration Server.

2. Go to the Delivery Options tab of the Server Account Manager.

3. In the Extra Processing pane of the Delivery Options tab, check the box labeled "Filter all incoming messages through one or more programs."

   Note that program delivery is separate from, and independent of, mailbox delivery. If either the "Your POP3/IMAP mailbox" or "Your UNIX mailbox" boxes are *also* checked, mail will continue to be delivered to the end users mailbox regardless of program delivery. In other words, if both the "Filter all..." and the "POP3/IMAP mailbox" boxes are checked, incoming mail will be processed by program delivery and *also* be delivered to the user's mailbox.

4. In the dialog box, enter the command (program) that will process incoming messages for this account:

   Unix secure mode. By default, program delivery runs in secure mode in Unix environments. Check with the administrator to confirm that it is running in secure-mode. Under secure-mode, users need only enter the command name; they do not need to enter a path. For example, to run the program named `mymail` stored in the trusted directory, enter `mymail`.

Unix non-secure mode. If program delivery is running in non-secure mode, users must enter an absolute path locating the program to be run. (Program delivery does not make any use of the user path as stored in the user environment.) For example, to run the program named `mymail` stored in the `/usr/bin` directory, enter `/usr/bin/mymail`.

NT. In NT environments, the user must enter the filename exactly as the filename exists in the trusted directory (including the filename extension). Users do not need to enter a path. For example, to run the `mymail.exe` stored in the trusted directory, enter `mymail.exe`.

To run multiple programs, enter each program on a separate line by itself. Programs will be run in the order you specify. For example, to first run the `new_mail.exe` program, and then the `sort_mail.cmd` program, simply enter:

```
new_mail.exe
sort_mail.cmd
```

5. Click Change.

Note: Messaging Server will allow users to enter the name of a program that has not been placed in the trusted directory (or a program with an incorrect path if running in Unix non-secure mode). But program delivery will fail when it tries to use that program. At that point an error message is sent to the mail administrator and the incoming message bounced back to the sender with an "undeliverable" type notice.

To stop using program delivery to handle incoming messages for this account, simply uncheck the "Filter all..." box and delete the names of the program, or programs, listed in the dialog box. Then click Change.

To change the programs that program delivery runs for your mail, simply add, delete, or change the programs listed in the dialog box; then click Change.

Program Delivery in Unix Environments

This section discusses the following topics:

- Program Delivery and Unix
- How Program Delivery Works (Unix)
- Secure and Non-secure Modes (Unix)

- Setting Up Program Delivery (Unix)
- Suspending Program Delivery (Unix)
- Disabling Program Delivery (Unix)

Program Delivery and Unix

The following factors should be considered when using program delivery in Unix environments:

- **Restricted environment:** Many programs (especially shells such as `/bin/sh` and others) use information from environment variables to modify their behavior. For security reasons, the only environment variables passed to an external program are `TZ` (time zone information), `AGENT=` Messaging Server (for compatibility with sendmail), and sometimes `PATH`.

- **Setuid-root program:** Some programs need more permissions than those of the user who executes them. Such programs acquire root permissions when they are run. Setuid programs can be identified by an *s* as the user-execute permission in the output of `ls -l`, as shown here:

  ```
  -r-s--x--- 1 root mta 70064 Feb 17 10:32 sort_my_mail
  ```

- **Valid shell:** Before running a program, the login shell of the user the message is addressed to is checked against the list of valid shells found in the `/etc/shells` file. The `/etc/shells` file is simply a list of shells, one per line, that can be used to log into the system. If this file is missing or empty, the user's shell is checked against the following default list:

  ```
  /bin/sh
  /usr/bin/sh
  /bin/csh
  /usr/bin/csh
  /bin/ksh
  /usr/bin/ksh
  ```

Messaging Server therefore won't run commands for users who aren't normally allowed to log in and type the commands themselves.

How Program Delivery Works (Unix)

When a program is run in response to an incoming message, that program is run under the user ID of the owner of the account the message is addressed to. For example, if a message is addressed to the `salesdata` account, the program is run under the user ID of the owner of the `salesdata` account. Note, however, that for security reasons, program delivery will not run programs under the `root` user ID. For additional information on running programs for the `root` account, see "Specifying the User ID for Root (Unix)" on page 342.

The following algorithm is used to handle incoming mail when program delivery has been specified as a delivery option for incoming mail:

1. Messaging Server sets up a restricted environment consisting of only the variables `TZ` and `AGENT`.

2. Messaging Server permanently gives up root permissions by changing to those of the controlling user (using `setuid(2)`). The controlling user is the owner of the account the incoming message is addressed to. If the account is owned by `root`, the controlling user is the designated user as described in "Specifying the User ID for Root (Unix)" on page 342. Messaging Server changes to the controlling user's home directory if possible (it remains in `/tmp` if a failure occurs).

3. Messaging Server performs two checks:

   - It checks that the program to be run is located in the trusted directory. If the program is not in the trusted directory, it checks the trusted directory for the presence of a file named `INSECURE-PROGRAM-DELIVERIES`. If that file is present, it runs the program as specified by the user with an absolute path. If neither the program nor the `INSECURE-PROGRAM-DELIVERIES` file is present in the trusted directory, program delivery aborts and an error message is sent to the administrator.

   - It makes sure there are no special characters in the command. The special characters it checks for are `$ ^ & ( ) | ' ; < > CR` and `LF`. So, for example, you won't be able to run two programs connected by a

pipe. If one of these disallowed characters are present, program delivery fails and the incoming message is bounced back to the sender with an "undeliverable" type notice.

4. Messaging Server starts the program.
5. Messaging Server feeds the message to the program.
6. If the user has designated multiple programs, each program is run in the sequence the user specified.

If the program exits abnormally or produces any output, an error message is generated.

Secure and Non-secure Modes (Unix)

For general security-related information that applies to both the Unix and NT versions of program delivery, see "Security Considerations" on page 327.

The program delivery module in Netscape Messaging Server can operate in one of two security modes:

- **Secure mode**. In secure mode, program delivery will only run the executable files (programs) that are stored in the trusted directory. For details, see "Secure Mode" on page 339.
- **Non-secure mode**. In non-secure mode, program delivery will run any executable file (program) stored anywhere on the network. For details, see "Non-secure Mode" on page 339.

In this context, an *executable* file is any Unix file with execute permission or a link (hard or soft) to an executable file. In other words, program delivery treats links to programs as if they were the programs themselves.

Netscape recommends that you run program delivery in secure mode. Secure mode allows you as an administrator to specify that program delivery only run those executables that you have examined for security problems and placed in the trusted directory.

Netscape recommends against running program delivery in non-secure mode. In this mode any program anywhere on the network can be used by program delivery and there is no way to ensure that those programs are safe.

Secure Mode

Program delivery runs in secure mode by default.

When running in secure mode, the program delivery module ignores any path specified in the user's account and runs the version of the program that is stored in the trusted directory. This allows the administrator to specify the exact executable files that the program delivery feature will run. If the program is not stored in the trusted directory, program delivery exits with an error message.

For example, if users want to set up a program named `sort_mail` to sort new mail into folders as it arrives, they can specify it in their account as:

```
/usr/local/bin/sort_mail
```

or as

```
sort_mail
```

Regardless of whether or not the user specifies a path, if program delivery is running in secure mode (as Netscape recommends) program delivery will only execute the version of sort_mail in the trusted directory.

Non-secure Mode

When running in non-secure mode, program delivery will run programs stored anywhere on the network, not just those stored in the trusted directory.

Non-secure mode allows any user to have program delivery run any program, or any version of any program. If users can create (or modify) programs for program delivery to automatically run in response to an incoming message, there is no way to ensure that those programs are safe. **Therefore, Netscape cautions that running in non-secure mode endangers system and network security.**

To run program delivery in non-secure mode, simply place a file named `INSECURE-PROGRAM-DELIVERIES` in the trusted directory. The contents of this file do not matter and it does not have to be executable. The name of the file is case-sensitive and must be exactly as shown.

To create this file:

```
touch INSECURE-PROGRAM-DELIVERIES
```

If a file named `INSECURE-PROGRAM-DELIVERIES` is present in the trusted directory, program delivery runs the program identified by the absolute path specified by the user (or administrator). In other words, when program delivery is running in non-secure mode, programs must be qualified by an absolute path in the program delivery dialog box as explained in "Using Program Delivery to Handle Incoming Mail" on page 331.

In non-secure mode, program delivery relies entirely on the specified path. Even if there is a version of the program in the trusted directory, unless the absolute path points to that directory, program delivery will not run it. If no path is specified, or the path is incorrect, program delivery fails and an error message is generated. When program delivery fails, the incoming message is returned to the sender with an "undeliverable" type notice.

To stop running program delivery in non-secure mode and return to secure mode, simply remove the `INSECURE-PROGRAM-DELIVERIES` file from the trusted directory.

Running Programs as root

For security reasons, program delivery will not run programs as `root`. In order to use program delivery for an account owned by `root`, you must designate an alternate user ID to run programs for mail addressed to an account owned by `root`. For details, see "Specifying the User ID for Root (Unix)" on page 342.

Setting Up Program Delivery (Unix)

For security reasons, the program delivery module is disabled by default and must be explicitly activated by an administrator who is logged in on Messaging Server as `root`.

The administrator must perform the following procedures to set up program delivery:

1. Enable the program delivery module as described in "Enabling the Program Delivery Module" on page 330.

2. Select the programs that program delivery is going to work with, and make sure that they are safe to run.

3. Install the inspected programs in the trusted directory as described in "Installing Trusted Programs (Unix)" on page 341. (Or if you want to run program delivery in non-secure mode, place a file named `INSECURE-PROGRAM-DELIVERIES` in the trusted directory.)

4. Specify the shells that can be used with program delivery as described in "Setting up the List of Valid Shells (Unix)" on page 341.

5. If program delivery is going to be used for accounts owned by `root`, designate an alternate user ID under which to run programs as described in "Specifying the User ID for Root (Unix)" on page 342.

Once these steps have been completed and program delivery is set up, program delivery can be used to handle incoming mail as described in "Using Program Delivery to Handle Incoming Mail" on page 331.

Installing Trusted Programs (Unix)

Before installing a program in the trusted directory, first inspect it to make sure it is safe for program delivery to automatically run in response to an incoming message.

Then move or copy the inspected program into the trusted directory.

You can use a link in the trusted directory to a program stored somewhere else, but using a link may weaken security. By default, only administrators with `root` privileges can modify or replace a program in the trusted directory, but if you link to a program stored in a directory that grants broader access privileges, you run the risk of someone substituting a poorly written, corrupt, or unauthorized version of the program.

Setting up the List of Valid Shells (Unix)

If you want to allow users with login shells other than `sh`, `csh`, or `ksh` to use the program delivery feature, you need to set up `/etc/shells`.

If you're creating the `/etc/shells` file for the first time, you need to include entries for any of the six default shells that you want to allow.

Here is an example of a `/etc/shells` file:

```
/bin/csh
/bin/sh
/bin/ksh
/usr/bin/sh
/bin/tcsh
/usr/bin/csh
/usr/bin/ksh
/usr/bin/tcsh
```

Specifying the User ID for Root (Unix)

Program delivery will not run programs as `root`. If you are setting up program delivery for a mail account owned by `root`, you must specify an alternate user ID under which to run programs.

To specify the user ID for accounts owned by `root`, follow these steps:

1. Create a special user ID for running programs for mail accounts owned by `root`. For example, a user named `progdel`. Limit the permissions on this account to just those needed to run the programs.

2. Go to the SMTP System tab and enter the user ID for accounts owned by `root` in the Program Delivery pane. For example, enter `progdel` in the Safe user ID for running programs box. For details, see Chapter 3, "Configuring SMTP Services."

3. (Optional.) If you wish, you can also specify a group ID for running programs for accounts owned by `root`.

(Note that the pane labeled Unix delivery, has nothing to do with the program delivery module being described in this appendix.)

Suspending Program Delivery (Unix)

You can temporarily suspend all program deliveries by placing a file named `SUSPEND-PROGRAM-DELIVERIES` in the trusted directory. The contents of this file do not matter and it does not have to be executable. The name of the file is case-sensitive and must be exactly as shown.

When program delivery is suspended, incoming messages are not bounced back to the sender, instead they simply queue up waiting for program delivery to be resumed. **Therefore, administrators are cautioned not to suspend program delivery for long periods of time.**

To resume program delivery, simply remove the `SUSPEND-PROGRAM-DELIVERIES` file from the trusted directory.

Disabling Program Delivery (Unix)

To disable program delivery, simple remove all files from the trusted directory.

If program delivery is disabled, messages addressed to accounts that have specified program delivery as a delivery option are bounced back to the sender. This occurs even if the account has also enabled POP/IMAP mailbox delivery. In other words, if both program mailbox delivery are set up for an account, and all files are removed from the trusted directory, one copy of an incoming message will be placed in the account's mailbox and the message will also be bounced back to the sender with an "undeliverable" type notice.

Program Delivery in NT Environments

This section discusses the following topics:

- How Program Delivery Works (NT)
- Setting Up Program Delivery (NT)
- Suspending Program Delivery (NT)
- Disabling Program Delivery (NT)

How Program Delivery Works (NT)

When a program is run in response to an incoming message, that program is run under the server account specified at installation time. You can also use the SMTP System tab to specify an account to run programs.

The following algorithm is used to handle incoming mail when the program delivery has be specified as a delivery option for incoming mail:

1. Before running a program, the program delivery module performs two checks:

   - It makes sure that the program to be run is stored in the trusted directory. If there is no version of the program in the trusted directory, program delivery exits with an error message to the mail administrator and the incoming message is returned to the sender with an "undeliverable" type notice.

   - It makes sure there are no special characters in the specified command. The special characters is looks for are `$ ^ & ( ) | \ ; < > CR` and `LF`. So, for example, program delivery will not accept two programs connected by a pipe. If one of these disallowed characters are present, program delivery fails and the incoming message is bounced back to the sender with an "undeliverable" type notice.

2. Messaging Server runs the trusted program.

3. Messaging Server feeds the message to the running program.

4. If the user has designated multiple programs, each program is run in the sequence the user specified.

If the program exits abnormally or produces any output, an error message is generated and the incoming message is bounced back to the sender with an "undeliverable" type notice.

Setting Up Program Delivery (NT)

For security reasons, the program delivery module is disabled by default and must be explicitly activated by an administrator who is logged on Messaging Server with administrator privileges.

The administrator must perform the following procedures to set up program delivery:

1. Enable the program delivery module as described in "Enabling the Program Delivery Module" on page 330.
2. Select the programs that the program delivery module is going to work with and make sure that they are safe to run.
3. Install the inspected programs in the trusted directory as described in "Installing Trusted Programs (NT)" on page 345.

Once these steps have been completed and program delivery is set up, users can pick the trusted programs that they want program delivery to run when they receive messages. For information on how users select program delivery, see "Using Program Delivery to Handle Incoming Mail" on page 331.

Installing Trusted Programs (NT)

You must install in the trusted directory the trusted programs that you want to make available to the program delivery module. First inspect each program to make sure it is safe for program delivery to automatically run in response to an incoming message. Then move or copy the inspected program into the trusted directory.

For example, to enable the program delivery module to use a filter program named `mail-filter.exe`, follow these steps:

1. Make sure that `mail-filter.exe` is safe to run
2. cd *`server-root`*`\msg-`*`instance`*`\smtp-bin\delivery`
3. `copy \bin\mail-filter.exe mail-filter.exe`

Suspending Program Delivery (NT)

You can temporarily suspend all program deliveries by placing a file named `SUSPEND-PROGRAM-DELIVERIES` in the trusted directory. The contents of this file do not matter and it does not have to be executable. The name of the file is case-sensitive and must be exactly as shown.

When program delivery is suspended, incoming messages are not bounced back to the sender, instead they simply queue up waiting for program delivery to be resumed. **Therefore, administrators are cautioned not to suspend program delivery for long periods of time.**

To resume program delivery, simply remove the `SUSPEND-PROGRAM-DELIVERIES` file from the trusted directory.

Disabling Program Delivery (NT)

To disable program delivery, simply remove all files from the trusted directory.

If program delivery is disabled, messages addressed to accounts that have specified program delivery as a delivery option are bounced back to the sender. This occurs even if the account has also enabled POP/IMAP mailbox delivery. In other words, if both program mailbox delivery are setup for an account, and all files are removed from the trusted directory, one copy of an incoming message will be placed in the account's mailbox and the message will also be bounced back to the sender with an "undeliverable" type notice.

Chapter

13

Messaging Multiplexor

This chapter describes Netscape Messaging Multiplexor and provides installation and configuration instructions. This chapter contains the following sections:

- About Messaging Multiplexor
- Multiplexor Configuration
- Installing and Configuring Multiplexor (Unix)
- Installing and Configuring Multiplexor (NT)
- Running Multiplexor
- Uninstalling Multiplexor

About Messaging Multiplexor

Netscape Messaging Multiplexor is a specialized messaging server that acts as a single point of connection to multiple messaging servers. With the Multiplexor, large-scale messaging-service providers can distribute POP and IMAP user mailboxes across many machines to increase messaging capacity. All users connect to the single Multiplexor server, which redirects each connection to the appropriate messaging server.

If you provide electronic mail service to many users, you can install and configure Messaging Multiplexor so that an entire array of messaging servers will appear to your mail users to be a single host.

Messaging Multiplexor is provided as part of Netscape Messaging Server. You can install Messaging Multiplexor when first installing Messaging Server or other Netscape servers, or at a later time.

Netscape Messaging Multiplexor supports:

- Both unencrypted and encrypted (SSL) communications with mail clients.
- Client certificate-based authentication, described in "Certificate-Based Client Authentication" on page 352.
- User pre-authentication, described in "User Pre-Authentication" on page 353.
- Virtual domains that listen on different IP addresses and automatically append domain names to user IDs, described in "Virtual Domains" on page 354.
- Multiple installations of Multiplexor on different machines (one installation per machine).
- Multiple instances of Multiplexor on a server machine, described in "Multiple Multiplexor Instances" on page 356. Multiple instances can be used for alternate configurations such as SSL or the listen port that cannot be handled through virtual domains.
- Enhanced LDAP searching.

Multiplexor Benefits

Message stores on heavily used messaging servers can grow quite large. Spreading user mailboxes and user connections across multiple servers can therefore improve capacity and performance. In addition, it may be more cost-effective to use several small server machines than one large, high-capacity, multiprocessor machine.

If the size of your mail-server installation requires the use of multiple messaging servers, your organization can benefit in several ways from using the Messaging Multiplexor. The indirect connection between users and their message stores, coupled with the ease of reconfiguration of user accounts among messaging servers allows for the following benefits:

- **Simplified user management**

  Because all users connect to one server (or two, if you have separate Multiplexors for POP and IMAP), you can preconfigure email clients and distribute uniform login information to all users. This simplifies your administrative tasks and reduces the possibility of distributing erroneous login information.

  For especially high-load situations, you can run multiple Multiplexor servers and manage connections to them by DNS round robin or by using a load-balancing program, such as LocalDirector from Cisco Systems.

  Because Multiplexor uses information stored in the LDAP directory to locate each user's messaging server, moving a user to a new server is simple for the system administrator and transparent to the user. The administrator can move a user's mailbox from one messaging server to another, and then update the user's entry in the directory. The user's mail address, mailbox access, and other client preferences need not change.

- **Improved performance**

  If a message store grows prohibitively large for a single machine, you can balance the load by moving some of the message store to another machine.

  You can assign different classes of users to different machines. For example, you can choose to locate premium users on a larger and more powerful machine.

  Multiplexor performs some buffering, so that slow client connections (through a modem, for example) do not slow down the messaging server.

- **Decreased cost**

  Because you can efficiently manage multiple messaging servers with Messaging Multiplexor, you can decrease overall costs by purchasing several small server machines that together cost less than one very large machine.

- **Better scalability**

  With Messaging Multiplexor, your configuration can expand easily. You can incrementally add machines as your performance or storage-capacity needs grow, without replacing your existing investment.

- **Minimum user downtime**

  Using Messaging Multiplexor to spread a large user base over many small store machines isolates user downtime. When an individual server fails, only its users are affected.

- **Increased security**

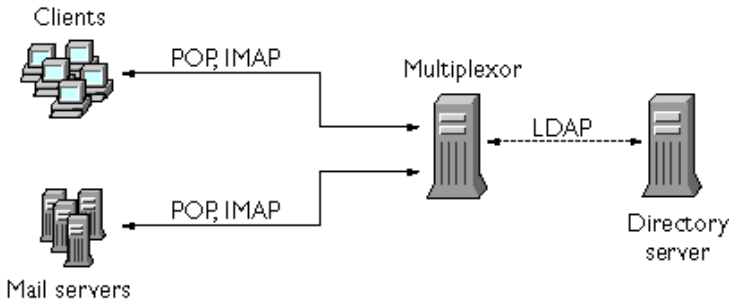
  You can use the server machine on which Messaging Multiplexor is installed as a firewall machine. By routing all client connections through the Multiplexor machine, you can restrict access to the internal message store machines by outside computers. Messaging Multiplexor supports both unencrypted and encrypted communications with clients.

How Multiplexor Works

Messaging Multiplexor is a multithreaded server that facilitates distributing mail users across multiple server machines. Multiplexor handles incoming client connections destined for other server machines (the machines on which user mailboxes reside). Clients connect to Multiplexor itself, which then redirects the session to the server with the correct mailbox. This capability allows Internet service providers and other large installations to spread message stores across multiple machines (to increase capacity) while providing the appearance of a single mail host for users (to increase efficiency) and for external clients (to increase security).

Figure 13.1 shows how servers and clients relate to each other in a Messaging Multiplexor installation.

Figure 13.1 How the Messaging Multiplexor interacts with clients and servers



All POP and IMAP clients work with Messaging Multiplexor. Messaging Multiplexor accepts connections, performs LDAP directory lookups, and routes the connections appropriately. As is typical with other mail-server installations, each user is assigned a specific address and mailbox on a specific messaging server. However, all connections are routed through Multiplexor.

In more detail, these are the steps involved in establishing a user connection:

1. A user's client connects to Multiplexor, which accepts preliminary authentication information (user name).
2. Multiplexor queries an LDAP directory server to determine which messaging server contains that user's mailbox.
3. Multiplexor connects to the proper messaging server, replays authentication, then acts as a pass-through pipe for the duration of the connection.

Encryption (SSL) Option

The Netscape Multiplexor supports both unencrypted and encrypted (SSL) communications between the IMAP server and mail clients.

In SSL mode, Multiplexor listens by default on port 993. You can specify a different port if you wish. The IMAP Multiplexor SSL supports STARTTLS which allows Multiplexor to promote non-SSL connections to SSL.

To enable SSL encryption for your IMAP service:

- When using the `mmp-setup` script, answer yes to the "`Should the IMAP MMP do SSL`" prompt. Then specify *`port`*, *`secmodfile`*, *`certfile`*, *`keyfile`*, *`keypass`*, *`ciphers`*, and *`certname`* parameters. See Table 13.9 on page 391 for details about configuration prompts.
- When specifying Multiplexor configuration options from the command line, use the `-s` parameters (*`secmodfile`*, *`certfile`*, *`keyfile`*, *`keypass`*, *`ciphers`*, and *`certname`*) parameters. See Table 13.4 on page 367 for details.

Certificate-Based Client Authentication

Multiplexor can use `certmap` to match a client's certificate to the correct user in the user-group LDAP database.

In order to use certificate-based client authentication, you must also enable SSL encryption as described in "Encryption (SSL) Option" on page 351.

You also have to configure a store administrator. You can use the mail administrator, but Netscape recommends that you create a unique user ID, such as `mmpstore` for this purpose so that you can set permissions as needed.

Note that Multiplexor does not support certmap plug-ins. Instead, Multiplexor accepts enhanced `DNComps` and `FilterComps` property value entries in the `certmap.conf` file. These enhanced format entries use the form:

`mapname``:DNComps` *`FROMATTR=TOATTR`*
`mapname``:FilterComps` *`FROMATTR=TOATTR`*

So that a *`FROMATTR`* value in a certificate's subjectDN can be used to form an LDAP query with the *`TOATTR=value`* element. For example, a certificate with a subjectDN of `"cn=Pilar Lorca, ou=pilar o=airius.com"` could be mapped to an LDAP query of `"(uid=pilar)"` with the line:

`mapname``:FilterComps ou=uid`

To enable certificate-based authentication for your IMAP service:

1. Decide on the user ID you intend to use as store administrator.

   While you can use the mail administrator for this purpose, Netscape recommends that you create a unique user ID for store administrator. For example, `mmpstore`.

2. Make sure that SSL encryption is (or will be) enabled as described in "Encryption (SSL) Option" on page 351.

3. Configure Multiplexor to use certificate-based client authentication:

   - When using the `mmp-setup` script, answer yes to the "`Do you have a certmap.conf (Should MMP do client-cert auth)`" prompt. Then specify the location of the `certmap.conf` file, the store administrator user ID, and password. See Table 13.9 on page 391 for details.

   - When specifying Multiplexor configuration options from the command line, use the `-cm` *`cmfile`*, `-d` *`storeadmin`*, and `-W` *`password`* options. See Table 13.4 on page 367 for details about these configuration prompts.

User Pre-Authentication

Multiplexor has the option of pre-authenticating users by binding to the directory as the incoming user and logging the result.

Note: Enabling the pre-authentication option will reduce server performance.

The log entries are in the format:

`date time` `(sid 0x%p) user` *`name`* `pre-authenticated - client IP` *`address`*

Where date is in the format `yyyymmdd`, time is in the format `hhmmss`, `sid` is the session object, the `user` *`name`* includes the virtual domain (if any), and the IP address is in dot-quad format.

Virtual Domains

Multiplexor supports the 4.0 format virtual domain file syntax.

Virtual domains listen on different IP addresses and automatically append domain names to user IDs. They can also be used to specify alternate configurations.

Multiplexor can map IP addresses to domain names for searching an LDAP directory and for logging in to the store server. When a connection is accepted from a client, if the server's IP address is in the virtual domain mapping file, the domain is appended to the user ID and used for the LDAP search and for subsequent replay of authentication. This capability is useful for hosting multiple domains with overlapping user ID name spaces.

To enable virtual domains:

- When installing in Unix environments, answer yes to the "`Should this MMP do virtual domain mapping?`" prompt. Then specify the location (name) of the virtual domain mapping file and the domain delimiters it uses. See Table 13.7 on page 379 for details.
- When specifying Multiplexor with the Server Setup program in Windows NT environments, answer yes to the "`Use Virtual Domain Mapping`" prompt. See Table 13.9 on page 391 for details.
- When specifying Multiplexor configuration options from the command line in Windows NT environments, use the `-vd` *`file`* option followed by the `-vdd` *`str`* option. See Table 13.4 on page 367 for details.

Each entry of a virtual domain file has the following syntax:

```
vdmap name IPaddr
name:property value
```

Where *`name`* is whatever name you choose to use, *`IPaddr`* is in dot-quad format, and *`property`* and *`value`* pairs configure the virtual domain as described in Table 13.1. When set, virtual domain properties override global configuration parameters.

Table 13.1 describes the properties you can specify for a virtual domain. (See Table 13.3 for a description of configuration variables you can specify for Multiplexor.)

Table 13.1 Virtual domain properties

Property	Description
`BindDN` `BindPass`	User-group LDAP credentials. This property is ignored if the `LdapUrl` property is not set. In most cases it is not necessary to specify this property.
`CanonicalVirtual DomainDelim`	The delimiter used by the Multiplexor to separate the user ID from the appended virtual domain when talking to the message store server and LDAP server.
`CertMapFile`	If certificate-based authentication is enabled, the `certmap.conf` file for this domain.
`LdapUrl`	The URL for user-group LDAP information using the format: `ldap[s]://HOST[:PORT]/BASEDN`
`MailHostAttrs`	A comma-separated list of LDAP attributes to be treated as the users' mailhost. Multiplexor tries to connect to each server returned by the search in turn.
`PreAuth`	If set to "Yes," pre-authentication is enabled. Note that enabling pre-authentication reduces server performance.
`StoreAdmin` `StoreAdminPass`	The store administrator credentials (user ID and password). For proxy-authentication to the store server when the client is authenticated to the Multiplexor via an SSL client certificate.
`UidSearch`	A `printf`-style format string with which to construct a user-group LDAP query for the user's mailhost. These are the valid escapes for `UidSearch`: • `%s` - For `userd+virtualdomain` • `%U` - For `userid` only • `%V` - For virtual domain only • `%C` - Client IP address • `%S` - Server IP address • `%D` - Client cert DN
`VDomain`	The virtual domain name to append to incoming user names (for sites that have virtual domains enabled). If omitted, no domain name is appended to the replayed credentials.

Table 13.1 Virtual domain properties (Continued)

Property	Description
`VirtualDomainDelim`	A string listing all of the virtual domain delimiters that the Multiplexor accepts. This is just a string of characters not separated by spaces or commas. Any character in the string is treated as a delimiter.
`VirtualDomainFile`	The name of the file containing your virtual domain mapping. If a filename is entered for this variable, virtual domains are enabled. See Table 13.1 for a description of configuration variables that apply to virtual domains.

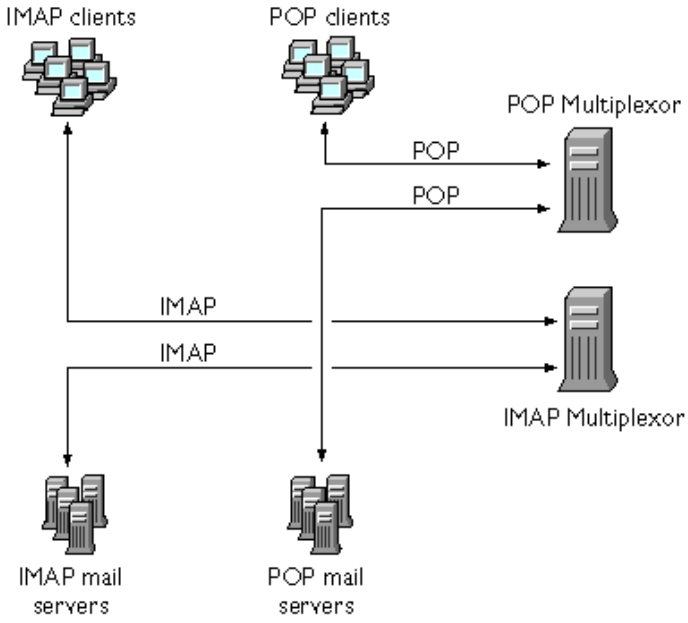
Multiple Multiplexor Instances

You can create multiple instances of Multiplexor, all of which must be on the same server. In other words, you can have multiple installations of Multiplexor on different servers, and on any given machine you can have multiple instances.

Using multiple instances of Multiplexor allows you to create alternate configurations, such as SSL or the listen port, that cannot be handled through virtual domains.

You can configure a single instance of Multiplexor to support both POP and IMAP protocols (as shown in Figure 13.1), or you can create separate Multiplexor instances for each protocol, as shown in Figure 13.2 . By splitting messaging services across different machines, you can tune the resources on each computer for maximum performance.

Figure 13.2 Separate Multiplexors and messaging servers for POP and IMAP support



For instructions on creating multiple instances of Multiplexor, see Installing and Configuring Multiplexor (Unix) or Installing and Configuring Multiplexor (NT).

Multiplexor Configuration

You control how Multiplexor operates by setting configuration parameters.

There are two ways to set configuration parameters:

- **Configuration environment variables** are parameters that are permanently stored and applied each time Multiplexor is run. There are several methods of specifying configuration environment variables as described in Table 13.2.
- **Command line configuration options** are parameters that are specified on the command line at the time Multiplexor is run. A configuration parameter specified on the command line at the time Multiplexor is run

overrides any corresponding value specified as an environment variable. For example, if you have specified a value for the `capability` parameter as an environment variable, but you enter a different `capability` value from the command line, the command line value will be used by that Multiplexor session.

Note that the only difference between environment variable and command-line option parameters is how you choose to set them. Any configuration parameter can be set as either an environment variable or a command-line option.

- **In Unix environments**, environment variables for each Multiplexor instance are stored in configuration files named `ImapMMP.config` and `PopMMP.config`. These files are stored in the directory *`MMPRoot/MMP_instanceName`*.
- **In NT environments**, environment variables for each Multiplexor instance are stored in the Windows registry.

Table 13.2 describes different ways to set environment variables. Table 13.3 on page 360 describes the configuration parameters you can set for Multiplexor.

Table 13.2 Methods of setting Multiplexor configuration parameters

Unix Environment Configuration Methods	NT Environment Configuration Methods
Run the Server Setup program to install and configure the first Multiplexor instance as described in "Multiplexor Installation (Unix)" on page 374 which takes you through the configuration prompts described in Table 13.7 on page 379. You can also use this method to configure subsequent Multiplexor instances.	Run the Server Setup program as to install and configure the first Multiplexor instance described in "Multiplexor Installation (NT)" on page 387 which takes you through the configuration prompts described in Table 13.9 on page 391. You can also use this method to configure subsequent Multiplexor instances.
Once Multiplexor has been installed with the Server Setup program, you can configure Multiplexor instances with the `mmp-setup` script as described in "Multiplexor Installation (Unix)" on page 374 which takes you through the configuration prompts described in Table 13.7 on page 379.	(The `mmp-setup` script is not available on NT platforms.)

Table 13.2 Methods of setting Multiplexor configuration parameters (Continued)

Unix Environment Configuration Methods	NT Environment Configuration Methods
An alternate method of configuring Multiplexor instances is to run the `PopProxy` or `ImapProxy` commands with the `install` and command-line options. This displays configuration syntax on your screen which you can paste into Multiplexor configuration files.	An alternate method of configuring Multiplexor instances is to run the `PopProxy` or `ImapProxy` commands with the `install` option and use command-line options to specify configuration variables as described in "Command-Line Configuration Options" on page 366 and Table 13.4 on page 367.
You can also directly edit the configuration parameters in the Multiplexor configuration files as described in "Directly Setting Configuration Variables (Unix)" on page 383.	You can also directly edit the configuration parameters in the Windows NT Registry as described in "Directly Specifying Configuration Variables (NT)" on page 395.

Note: You cannot run the `mmp-setup` script or `PopProxy` or `ImapProxy` commands, or directly edit variables, until your first instance of Multiplexor has been set up through the Server Setup program.

Multiplexor Configuration Parameters

You control how Multiplexor operates by specifying various configuration parameters, either as environment variables or as command-line options.

Table 13.3 on page 360 describes the parameters you can set. (Table 13.1 on page 355 describes the parameters that you can set for virtual domains.)

Note: The names and values of configuration parameters are case-sensitive.

Table 13.3 Multiplexor configuration parameters

Variable	Description
`BacksidePort`	Port on which to connect to message store server. This parameter lets you run a multiplexor and a store server on the same machine, with the store server on a different port. You might want to do this if you want a flat configuration—that is, if you want to run Multiplexors on all machines. For information about specifying ports, see "Configuration Prompts (NT)" on page 391. Default = `110` for POP3; `143` for IMAP (the standard ports) (select `n` on Unix installer to choose defaults)
`Banner`	Banner replacement string. Multiplexor will use the string you specify instead of its default banner for its greeting line. Default = `"Netscape Messaging Multiplexor ready"` (select `n` on Unix installer to choose default)
`BaseDN`	BaseDN is where Multiplexor begins its search in the LDAP database. Some LDAP servers may require that a client (in this case Messaging Multiplexor) be authenticated before it can search the database for certain information. If your server has ACLs that require some level of authentication for getting a user's mail information, set this parameter. The `BaseDN` must specify an entry to which the bind distinguished name (`binddn`) has access privileges for operations on the directory database. Default: `o=mcom.com` (Change this to your base DN.)
`BindDN`	LDAP bind DN to use to authenticate to the LDAP server. Default = `Anonymous`
`BindPass`	LDAP bind password. There is no default.

Table 13.3 Multiplexor configuration parameters (Continued)

Variable	Description
`CanonicalVirtual DomainDelim`	Canonical virtual domain delimiter. The character used by Multiplexor to separate the user ID from the appended virtual domain when talking to the message store server and LDAP server. The default is `+`, so user IDs passed to LDAP and the message store servers have the form `userid+virtual.domain`. Default = `"+"` (default string passed to directory is "*`userid+virtual.domain`*")
`Capability`	Capability replacement string. Multiplexor will use the string you specify for `Capability` instead of its default (own) capability to tell IMAP clients what it (or the servers behind it) can do. This variable has no effect in POP3. If you are using Netscape servers and want to use the Manage Mail Account feature, you must specify this `Capability` configuration parameter to change Multiplexor's capability. A suggested string to support Manage Mail Account is: `IMAP4 IMAP4rev1 AUTH=LOGIN AUTH=PLAIN X-NETSCAPE` Default = `IMAP4 IMAP4rev1 AUTH=LOGIN AUTH=PLAIN` (select `n` on Unix installer to choose default)
`CertMapFile`	The name of the `certmap.conf` file. Default = `certmap.conf`
`LDAPHost`	LDAP server and port to use for user information. The host machine name and port of the LDAP server that contains your user database (among other things). You must set up the LDAP host before the Multiplexor can work properly. Default = *`Localhost:IPPORT_LDAP`* (port 389)

Table 13.3 Multiplexor configuration parameters (Continued)

Variable	Description
`ListenPort`	The port on which to listen for incoming client connections. An IP address may be specified for an optional bind address, for multi-homed hosts (hosts that have more than one IP address). For information about specifying ports, see "Choosing a Port Number (Unix)" on page 379 or "Configuration Prompts (NT)" on page 391. Default = `110` for POP3; `143` for IMAP (the standard ports) (select `n` on Unix installer to choose defaults)
`LogDir`	The directory in which the Multiplexor creates log files. If you specify a directory that does not exist, no log file is created. Log file names have the following format: *`MMP_yyyymmdd`*`.log` Default = current directory (the directory that contains Multiplexor)
`LogLevel`	The logging verbosity level—the amount of information written into log files. You can specify a number from 0 through 10, with 10 representing the highest level of verbosity. At higher levels, more events are logged. The higher the level, the more information in the log. Default = `1`
`MailHostAttrs`	Comma-separated list of LDAP attributes identifying the user's mail host. Multiplexor tries to connect to each server returned by the search in the order specified by the list. Default = `mailHost`

Table 13.3 Multiplexor configuration parameters (Continued)

Variable	Description
`NumThreads`	The maximum number of worker threads to allocate. If the machine has multiple CPUs, running the Multiplexor with worker threads will improve performance. The optimal number of work threads is the number of processors on the machine. For example if your machine has two CPUs, specify 2. If this is a single-processor machine, specify 0 for optimal performance. Default = `0 (the` main thread does all the work)
`PreAuth`	Enables Global Roaming preauthentication. With preauthentication, clients authenticate to Multiplexor and Multiplexor relays authentication information to the message store. If set to "Yes," pre-authentication is enabled. Note that enabling pre-authentication reduces server performance. Default = off
`ServerDownAlert`	IMAP only. String returned to client in an IMAP ALERT message when Multiplexor cannot connect to a user's store server. Default = `"Your IMAP server appears to be temporarily out of service."`
`SpoofMessageFile`	The file to use for POP3 inbox spoofing. Multiplexor can imitate a base-functionality POP3 server in case Multiplexor can't connect to a client's store machine. In such a situation, Multiplexor creates an inbox for the user and places this one message into it. The format of the message contained in this file should conform to RFC 822 (including the final `'.'`). Default = no spoof message
`SSLBacksidePort`	Port on which to connect to message store server using SSL.

Table 13.3 Multiplexor configuration parameters (Continued)

Variable	Description
`SSLCertFile`	Server certificate database file location (defined when you obtained a certificate for this server). Multiplexor needs a server certificate to offer to clients in the handshake phase of SSL. The location specified here should be absolute, not relative to the Multiplexor installation directory. Default = `cert7.db`
`SSLCertName`	Name of this server's SSL server certificate (defined when you obtained the certificate). Multiplexor uses this string to identify the certificate in its certificate database (*`certfile`*).

`SSLCipherSecs`	A colon-separated list of ciphers (or the string "all") representing the cipher algorithms that this server can use to encrypt SSL sessions. The client and server agree to one of them when a session is established. The available cipher specifications are:
	
	`SSL_RSA_WITH_RC4_128_ MD5`
	`SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA`
	`SSL_RSA_WITH_3DES_EDE_CBC_SHA`
	`SSL_RSA_FIPS_WITH_DES_CBC_SHA`
	`SSL_RSA_WITH_DES_CBC_SHA`
	`SSL_RSA_EXPORT_WITH_RC4_40_MD5`
	`SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5`
	`SSL_RSA_WITH_NULL_MD5`
	
	Default = `all`

Table 13.3 Multiplexor configuration parameters (Continued)

Variable	Description
SSLEnable	Whether or not to enable SSL. If set to "True" or "Yes", Multiplexor will listen on both normal and SSL ports. If SSL is enabled, all of the following variables must be set. You can specify an empty parameter with empty quotes (""). SSLSecmodFile SSLCertFile SSLKeyFile SSLKeyPasswd SSLCipherSpecs SSLCertNickname Default = n (SSL not enabled)
SSLKeyFile	Key database file location (defined when you obtained a certificate for this server). Multiplexor needs a private key corresponding to its SSL server certificate. The location specified here should be absolute, not relative to the Multiplexor installation directory. Default = key3.d
SSLKeyPasswd	Password that protects access to the private key file. The password may be null if the key is not password-protected. Default = no password protection
SSLListenPort	Port on which to listen for incoming SSL connections. Default = 993 for IMAP (the standard SSL IMAP port) (select n on Unix installer to choose default)
SSLSecmodFile	Security module database file location (usually null). If you have hardware accelerators for SSL ciphers, this file describes them to the Multiplexor. Default = no secmod database (select n on Unix installer to choose default)

Table 13.3 Multiplexor configuration parameters (Continued)

Variable	Description
`StoreAdmin`	The user name of the store administrator for proxy authentication during SSL. Default = Null Recommended = `mmpstore`
`StoreAdminPass`	Password of store admin for proxy-authentication during SSL. There is no default.
`VirtualDomainDelim`	String of acceptable virtual domain delimiters. Any character in this string will be treated as a domain delimiter in a user ID received by Multiplexor. (Multiplexor searches user IDs from the end.)
`VirtualDomainFile`	The name of the file containing your virtual domain mapping. If a filename is entered for this variable, virtual domains are enabled. See Table 13.1 on page 355 for a description of configuration variables that apply to virtual domains.
`UidSearch`	A `printf`-style format string with which to construct user-group LDAP query for the user's mailhost when virtual domains are enabled.

Command-Line Configuration Options

Once an instance of Multiplexor has been installed using the Server Setup program, you can run Multiplexor directly from the command line using either the `PopProxy` or `ImapProxy` commands as described in "Running Multiplexor" on page 395.

Note: In NT environments, the preferred method of running Multiplexor is from the Service Control Manager as described in "Running Multiplexor (NT)" on page 397. Normally, you would run Multiplexor from the command line only if you wanted to override one or more configuration options by specifying different options on the command line.

If you choose to run Multiplexor from a command line, configuration options you specify on the command line itself take precedence over any corresponding configuration environment variables contained in Unix configuration files or the Windows registry. For example, if there is a `UidSearch` environment variable set, but you specify a different `UidSearch` value from the command line, the one you specify on the command line is used for that Multiplexor session. When starting Multiplexor from the command line, configuration options that you do not specify on the command line are taken from environment variables if they exist, or set to default values if there is no existing configuration variable.

To run Multiplexor for POP service with command-line configuration options, use this syntax:

```
PopProxy [install] [options]
```

To run Multiplexor for IMAP service with command-line configuration options, use this syntax:

```
ImapProxy [install] [options]
```

Where the configuration *`options`* you can set are described in Table 13.4. For the environment variables that the command-line options specify, see Table 13.3 on page 360.

Table 13.4 Multiplexor command-line option descriptions

Option	Description
`-a` *`str`*	IMAP only. IMAP ALERT message. Specifies the message returned to IMAP clients when the Multiplexor cannot connect to a user's store server. Configuration parameter = `ServerDownAlert`
`-b` *`basedn`*	The distinguished name (DN) to use as the search base for LDAP user queries. Configuration parameter = `BaseDN`
`-ba` *`str`*	Banner replacement string. Multiplexor uses *`str`* instead of its default banner for its greeting line. Configuration parameter = `Banner`
`-c` *`port`*	The port on which to connect to the message store server. Configuration parameter = `BacksidePort`
`-cm` *`cmfile`*	Specifies the `certmap.conf` file. Configuration parameter = `CertMapFile`

Table 13.4 Multiplexor command-line option descriptions (Continued)

Option	Description
`-d` *`storeadmin`*	The user name of the store administrator for proxy authentication during SSL. Configuration parameter = `StoreAdmin` Recommended = `mmpstore`
`-D` *`binddn`*	The distinguished name used by Multiplexor to authenticate to an LDAP server when performing an operation. Configuration parameter = `BindDN`
`h` *`host`*`[:`*`port`*`]`	The host machine name and port of the LDAP server that contains your user database (among other things). Configuration parameter = `LDAPHost`
`install`	In Unix environments, the `install` option simply prints out the configuration syntax you specify on the command line. In NT environments, the `install` option changes the configuration variables you specify in the Windows registry. See "Using the install Option" on page 371 for details. **Note:** If you use the `install` option, Multiplexor does not run as a server.
`-m` *`file`*	The file to use for POP3 inbox spoofing. Configuration parameter = `SpoofMessageFile`
`-ma` *`str`*	Comma-separated list of LDAP attributes identifying the user's mail host. Multiplexor tries to connect to each server returned by the search in the order specified by the list. Configuration parameter = `MailHostAttrs`
`-n` *`instance`*	The name of the instance that is displayed in service control manager. If the Multiplexor instance of this name already exists, the configuration options you specify on the command line will be applied to that instance. If no instance of this name exists, a new one will be created.
`-o` *`dir`*	The directory in which Multiplexor creates log files. If you specify a directory that does not exist, no log file is created. Log file names have the following format: *`MMP_yyyymmdd`*`.log` Configuration parameter = `LogDir`

Table 13.4 Multiplexor command-line option descriptions (Continued)

Option	Description
`-p [IP:]`*`port`*	The port on which to listen for incoming client connections. `IP` is an optional bind address for a machine that is multihomed (has more than one IP address). For information about specifying ports, see "Choosing a Port Number (Unix)" on page 379 or "Configuration Prompts (NT)" on page 391. Configuration parameter = `ListenPort`
`-pre`	Enables Global Roaming preauthentication; clients can authenticate to Multiplexor; Multiplexor relays authentication information to the message store. Configuration parameter = `PreAuth`
`-ps` *`port`*	The port on which to listen for incoming client SSL connections. Configuration parameter = `SSLListenPort`
`-s` *`SSLSecmodFile SSLCertFile SSLKeyFile SSLkeypass SSLCipherSecs SSLCertNickname`*	Specify whether to enable SSL connections with clients. If you enable SSL, Multiplexor listens on both normal and SSL ports. If you do enable SSL by using the `-s` option in the command line, you must set the six required SSL parameters in the following order, on the command line: `-s` *`SSLSecmodFile SSLCertFile SSLKeyFile SSLkeypass SSLCipherSecs SSLCertNickname`* See Table 13.5 on page 370 for descriptions of these six parameters, and Table 13.3 on page 360 for a description of the variables that they set. You can specify an empty parameter with empty quotes (`""`).
`-t` *`num`*	The maximum number of worker threads to allocate to the machine on which Multiplexor runs. Configuration parameter = `NumThreads`
`-us` *`str`*	String for user ID search. When performing LDAP directory searches, Multiplexor uses *`str`* instead of the default. Configuration parameter = `UidSearch`

Table 13.4 Multiplexor command-line option descriptions (Continued)

Option	Description
`-v num`	The logging verbosity level—the amount of information written into log files. You can specify a number from 0 through 10, with 10 representing the highest level of verbosity. At higher levels, more events are logged. Configuration parameter = `LogLevel`
`-vd file`	Virtual domain file location. Configuration parameter = `VirtualDomainFile`
`-vdd str`	Virtual domain delimiter list. A string containing the acceptable delimiter characters for virtual domains. Configuration parameter = `VirtualDomainDelim`
`-vddc char`	Canonical virtual domain delimiter. Configuration parameter = `CanonicalVirtualDomainDelim`
`-w pass`	The password associated with the bind distinguished name, for authenticating to an LDAP server. Configuration parameter = `BindPassword`
`-W pass`	The password of the store administrator for proxy authentication during SSL. Configuration parameter = `StoreAdminPass`
`-x str`	Capability replacement string. Configuration parameter = `Capability`

Table 13.5 SSL command line parameters for -s command

Option	Description
`SSLSecmodFile`	Security module database file location (usually null). Configuration parameter = `SSLSecmodFile`
`SSLCertFile`	Server certificate database file location (defined when you obtained a certificate for this server). Configuration parameter = `SSLCertFile`
`SSLKeyFile`	Key database file location (defined when you obtained a certificate for this server). Configuration parameter = `SSLKeyFile`

Table 13.5 SSL command line parameters for -s command (Continued)

Option	Description
`SSLKeyPasswd`	Password that protects access to the private key file. The password may be null if the key is not password-protected. Configuration parameter = `SSLKeyPasswd`
`SSLCipherSpecs`	A colon-separated list of ciphers (or the string "`all`") representing the cipher algorithms that this server can use to encrypt SSL sessions. See Table 13.3 on page 360 for the available cipher specifications. Configuration parameter = `SSLCipherSpecs`
`SSLCertNickname`	Name of this server's SSL server certificate (defined when you obtained the certificate). Configuration parameter = `SSLCertNickname`

Using the install Option

Note that using the `install` option aborts loading Multiplexor. In other words, if you run `PopProxy` or `ImapProxy` with the `install` option, Multiplexor does not start.

When running Multiplexor from the command line, you can use the `install` option:

- **In NT environments**, if you run `PopProxy` or `ImapProxy` with the `install` option, any options you specify on the command line are added to or changed in the NT registry. In other words, options you specify on the command line become environment variables. The next time Multiplexor is started, it will take those variables from the registry. Note that if a parameter is specified in the registry, and you do not include it on the command line, it remains as it was in the registry.

- **In Unix environments**, the `install` option does *not* make any changes to any of the configuration files. In other words, the `install` option does not create any environment variables. Instead, the configuration options you specified on the command line are printed out, and you can copy those into a configuration file.

Installing and Configuring Multiplexor (Unix)

Messaging Multiplexor is available as part of Netscape Messaging Server. You can install Multiplexor at the same time as you install Messaging Server, or you can install it later using the Server Setup program. Either way, you first need to prepare the system to support Multiplexor.

Before You Install (Unix)

Before you install Messaging Multiplexor on a Unix machine, perform the following tasks:

1. Choose the machine on which you will install Multiplexor. Netscape recommends against installing Multiplexor on a system that is also running Messaging Server or Directory Server. It is best to use a separate machine for Multiplexor.

2. Check that the system meets all the hardware and software requirements for using Netscape Messaging Server. For more information about installation requirements, see the *Messaging Server Installation Guide*.

3. On the machine that Messaging Multiplexor is to be installed on, create a new user to be used exclusively by Multiplexor. This new user must belong to a group. Suggested names for the user are `nsmmp` ("Netscape Messaging Multiplexor") or `nsmail`.

4. Set up the LDAP directory server and its host machine for use with Messaging Server, if they are not already set up. For more information, see the Directory Server documentation.

5. If you already have an older version of the Multiplexor installed and want to replace it, you must remove the old version of Multiplexor before you can install the new one. To remove Multiplexor, remove the `mmp` server root directory and the `/etc/netscape.mmp.conf` file.

HP-UX. If you're using Messaging Multiplexor on HP-UX, you must install the operating-system patches that are required to run Messaging Server 4.1. For more information, see the *Messaging Server Installation Guide*.

Also for HP-UX, you should increase the values of the configurable kernel parameters `maxfiles`, `maxfiles_lim`, and `nfiles` to a much larger number. For example, with the settings shown below, Multiplexor can support approximately 8,000 simultaneous sessions. (The more RAM you have in your machine, the higher these values can be.)

```
maxfiles 16384
maxfiles_lim 16384
nfiles 32768
```

See your platform documentation for information on how to set these parameters.

Multiplexor Files (Unix)

In Unix environments, Multiplexor executable files are stored in the Multiplexor installation directory (*`MMPRoot`* in this document), and two subdirectories of *`MMPRoot`*: *`MMPRoot`*`/bin` and *`MMPRoot`*`/lib`. At installation time you specify the directory that you want to use as *`MMPRoot`*. For example, `/usr/netscape/server4/bin/mmp`.

When you install Multiplexor, you create one or more Multiplexor instances. Different instances can use different configuration variables. Each instance has its own directory that contains the configuration files for that instance. Instance directories are created as subdirectories of *`MMPRoot`*. For example, *`MMPRoot`*`/mplex1`.

Table 13.6 lists the principal files that make up a Messaging Multiplexor installation in a Unix environment.

Table 13.6 Messaging Multiplexor files (Unix)

File	Description
`PopProxy`, `ImapProxy`	The executable Messaging Multiplexor programs for IMAP and POP services, respectively (installed in directory *`MMPRoot`*`/bin`).
`ImapMMP.config`, `PopMMP.config`	Configuration files specifying environment variables used for IMAP and POP services, respectively (installed in directory *`MMPRoot/MMP_instanceName`*).
`ImapMMP`, `PopMMP`	Shell scripts that set environment variables and execute Multiplexor for IMAP and POP services, respectively (installed in directory *`MMPRoot/MMP_instanceName`*). May be included in the `init` directory to automatically start up Multiplexor.
`mmp-setup`	The Multiplexor Installer (installed in directory *`MMPRoot`*).
`libldapv30` (or `libldap32v30`), `libnspr21`, `libplc21`, `libplds21`	Shared libraries used by Messaging Multiplexor (installed in directory *`MMPRoot`*`/lib`).
`$CONFIG_FILE`	Identifies the location of *`MMPRoot`* (installed in `/etc/netscape.mmp.conf`).

Multiplexor Installation (Unix)

In Unix environments, there are two ways to install Messaging Multiplexor on a machine:

- When you install Messaging Server or other Netscape components, the Server Setup program gives you the option of choosing to install Messaging Multiplexor at the same time. You can run the Server Setup program to

install Multiplexor at any time. See "Creating a Multiplexor Instance (Unix)" on page 375 for details. (For instructions on using the Server Setup program, see the *Messaging Server Installation Guide*.)

- Once Multiplexor has been installed on a machine, you can also use the Multiplexor Installer to create additional Multiplexor instances by running the `mmp-setup` script on that machine. See "Creating Additional Instances (Unix)" on page 377 for details.

Note: Netscape recommends that Multiplexor not be installed on a machine that is also running either Messaging Server or Directory Server.

Creating a Multiplexor Instance (Unix)

You must use the Server Setup program to create your first Multiplexor instance. This program sets up and then uses the Netscape Messaging Multiplexor Installer. (Note that Multiplexor installation is not the default; you must select it as part of the Messaging Server suite.) For subsequent installations of additional instances, you can call the Multiplexor Installer directly by running the `mmp-setup` script.

When the Multiplexor Installer starts (either from the Server Setup program or from running `mmp-setup`), follow these steps to install and configure the Multiplexor:

1. When prompted for the user name that Multiplexor should run as, enter the user name that you created for exclusive use of Multiplexor as explained in "Before You Install (Unix)" on page 372.

   The default value is `nobody`. Change this to the user name that you created.

2. At the next prompt, enter the installation directory (called *`MMPRoot/`* in this document), the directory path into which you want Multiplexor to be installed.

   To accept the default (`/usr/netscape/suitespot4/mmp`), press Enter. In this example the `/mmp` directory is the *`MMPRoot`*.

   The installation program creates the directories for the Multiplexor installation and installs the files. (If you are creating a second or subsequent instance of Multiplexor on this machine, the installation program uses the existing installed files.)

The installer program starts to create an instance of Multiplexor from those files. (If there is a previously installed instance of Multiplexor, you can at this point choose to either configure the existing instance or create a new one.)

3. If you choose to configure a new instance of Multiplexor, enter the name you want to give it.

   To accept the default name (the host name of the machine you are installing Multiplexor onto) press Enter. The installation program finishes creating the new instance, installing it into the subdirectory *MMPRoot/*`MMP-`*instanceName/* where *instanceName* is the name you specified.

4. Enter one of the following numbers to specify which kind of mail service you want this instance to support:

   `1` - Configure this instance for IMAP mail service
   `2` - Configure this instance for POP3 mail service
   `3` - Configure this instance for both IMAP4 and POP3 mail service

5. You are then stepped through a series of configuration prompts that allow you to specify environment variables that will control how this instance of Multiplexor operates. See "Configuration Prompts (Unix)" on page 379 for a description of these prompts.

6. The installer shows you a summary of the information you have entered. If the information is correct, type `y`. If you need to make changes, type `n`, in which case the install program takes you back through the prompts again.

   After you have approved the configuration parameters and the Multiplexor Installer has implemented them, the program displays the following information:

   - The location of shell scripts you can use to start Multiplexor. Depending on your system's capabilities, you may be able to put these scripts in your `init` directory to start Multiplexor automatically at system startup. Alternatively, you can run the scripts from the command line by typing `ImapMMP.sh start` or `PopMMP.sh start`. These shell scripts set environment variables that control Multiplexor configuration and then execute the Multiplexor program.

   - The location of the Netscape Messaging Multiplexor Installer (`mmp-setup`). When you use the Netscape Servers installation program to install the Messaging Multiplexor, the Multiplexor Installer installs itself

(at the top level of the installation directory, under *MMPRoot/*), so that you can later execute it directly to modify the configuration of this instance.

- Instructions for removing Messaging Multiplexor, in case you ever need to. To remove all instances of Multiplexor from the host, remove the entire installation directory (*MMPRoot/*) and the Multiplexor configuration file (`netscape.mp.conf`, in the directory `/etc/`). To remove only a single instance of Multiplexor, remove just the subdirectory *MMPRoot*/MMP_*instanceName/*.

Creating Additional Instances (Unix)

Use the Multiplexor Installer to create a new instance after an initial installation.

To run the Multiplexor Installer, follow these steps:

1. Go to the directory that contains the Multiplexor Installer.

   The program is installed at the top of the installation directory, under *MMPRoot/*.

2. From the command line, type `mmp-setup`.

3. The Multiplexor Installer asks whether you want to change an existing instance or create a new one. If you choose to create a new instance, the installer takes you through the installation process, as described in "Creating a Multiplexor Instance (Unix)" on page 375.

Modifying an Instance (Unix)

Use the Multiplexor Installer to modify the configuration of a previously installed instance of the Multiplexor.

To run the Multiplexor Installer, follow these steps:

1. Go to the directory that contains the Multiplexor Installer. The program is installed at the top of the installation directory, under *MMPRoot/*.

2. From the command line, type `mmp-setup`.

3. The Multiplexor Installer asks whether you want to change an existing instance or create a new one. To change an existing instance, select the name of that instance. The installer takes you back through the prompts for configuring the Multiplexor so you can make the changes you want. For information about each parameter, see "Multiplexor Configuration" on page 357.

Multiplexor Configuration (Unix)

Multiplexor is controlled by setting the configuration parameters that are described in Table 13.3 on page 360. See Table 13.2 on page 358 for different methods you can use to set configuration parameters.

This section describes how to set Multiplexor configuration variables by:

- Using the configuration prompts invoked by running the Server Setup program or the `mmp-setup` script. See "Configuration Prompts (Unix)" on page 379.
- Directly editing the Multiplexor configuration files. See "Directly Setting Configuration Variables (Unix)" on page 383.

You can also set Multiplexor configuration variables by running the `PopProxy` and `ImapProxy` commands with the `install` option as described in "Command-Line Configuration Options" on page 366, then copying the screen output into the configuration files.

Listing Options (Unix)

To display a list of the current configuration parameters, you can execute Multiplexor for either POP or IMAP by running either `PopProxy` or `ImapProxy` from the command line, using the `-h` option with no attributes:

```
PopProxy -h
ImapProxy -h
```

Choosing a Port Number (Unix)

Keep the following in mind when entering a port number:

- Port numbers can be any number from 1 to 65535.
- If you choose a port number 1024 or lower, you must start all processes as root (superuser).
- Make sure the port you choose isn't already in use or reserved for another service. Look at the file `/etc/services` on the Multiplexor machine to make sure you don't assign a port number that is used by another service.

Configuration Prompts (Unix)

Table 13.7 lists the Multiplexor configuration prompts and associated environment variables in Unix environments. See Table 13.3 on page 360 for environment variable descriptions.

Table 13.7 Multiplexor configuration prompts (Unix)

Installer prompt	Environment variable
`LDAP Host:`	`LDAPHost`
`Base DN:`	`BaseDN`
`[IMAP4/POP3] MMP LogDir:`	`LogDir`
`Log Level:`	`LogLevel`
`Should the MMP bind to LDAP as someone in particular (y/n):` If you answer yes, the following two questions are asked:	`LDAPAuth`
`What user should the MMP authenticate as:`	`LDAPAuth`
`What's the password for "BindDN":`	`BindPassword`
`Number of Threads:`	`NumThreads`

Table 13.7 Multiplexor configuration prompts (Unix) (Continued)

Installer prompt	Environment variable

`Should the MMP listen on a non-default port (y/n):`
(See "Choosing a Port Number (Unix)" on page 379 for additional information about choosing a port.)
If you answer yes, the following question is asked:
`Which port:`
	`ListenPort`

`Do your main [IMAP4/POP3] servers listen on non-default ports (y/n):`
If you answer yes, the following question is asked:
`Which port:`
	`BacksidePort`

`Would you like to override the IMAP4 MMP's CAPABILITY response (y/n):`
If you answer yes, the following question is asked:
`Please type in the new CAPABILITY, and hit return.`
	`Capability`

`Would you like to override the MMP's banner (y/n):`
If you answer yes, the following prompt is displayed:
`Please type in the new banner, and hit return.`
	`Banner`

`Would you like to override the MMP's LDAP search string (y/n):`
If you answer yes, the following prompt is displayed:
`Please type in the new search string, and hit return.`
	`UidSearch`

Table 13.7 Multiplexor configuration prompts (Unix) (Continued)

Installer prompt	Environment variable

`Would you like MMP to use a particular attribute for auth replay (y/n):`
If you answer yes, the following prompt is displayed:
`Please type in attribute, and hit return.`
	`UidAttr`

`Would you like the MMP to use a non-default attribute for the mailhost? (y/n):`
If you answer yes, the following prompt is displayed:
`Please type in the list of attributes, and hit return.`
(Comma-delimited list.)
	`MailHostAttrs`

`Should this MMP do virtual domain mapping(y/n):`
If you answer yes, the following prompt is displayed:
`Please type the location of the mapping file, and hit return:`
	`VirtualDomainFile`

`Do you want to specify a list of virtual domain delimiters? (y/n)` If you answer yes, the following prompts are displayed:	`VirtualDomainDelim`
`Please type in the new delimiter and hit return:`	`VirtualDomainDelim`
`Do you want to specify a canonical virtual domain delimiter? (y/n)`	`CanonicalVirtual DomainDelim`
`Please type in the new delimiter and hit return:`	`CanonicalVirtual DomainDelim`

Table 13.7 Multiplexor configuration prompts (Unix) (Continued)

Installer prompt	Environment variable

`Would you like to provide a "spoof" message for POP3 (y/n):`
This prompt is only displayed for POP Multiplexors.
If you answer yes, the following prompt is displayed:
`Please type in the location of the file to be used, and hit return:`
	`SpoofMessageFile`

`Would you like to override the Server Down Alert for the MMP? (y/n)`
If you answer yes, the following prompt is displayed:
`Please type in the alert string and hit return:`
	`ServerDownAlert`

`Should the IMAP4 MMP do SSL (y/n):` (This prompt is only displayed for IMAP Multiplexors.) If you answer yes, the following five prompts are displayed:	`SSLEnable`
1 - `Should the MMP listen for SSL on a non-default port (y/n)` If you answer yes, you are asked: `Which port?`	`ListenPort`
2 - `Do your main servers listen on non-default SSL ports? (y/n):` If you answer yes, you are asked: `Which port?`	`SSLBacksidePort`

3 - `Do you have a secmod database (y/n):`
`If you answer yes, you are asked:`
`Please type the location of the secmod database:`
`Please type the location of the certificate database:`
`Please type the location of the key database:`
	`SSLSecmodFile`
	`SSLSecmodFile`
	`SSLCertFile`
	`SSLKeyFile`

Table 13.7 Multiplexor configuration prompts (Unix) (Continued)

Installer prompt	Environment variable
4 - `Is the key file password protected (y/n)` If you answer yes, you are asked:	`SSLKeyPasswd`
`Please type the key file password:`	`SSLKeyPasswd`
`Please type the names of the desired cipher specs, separated by colons, or the word "all":`	`SSLCipherSpecs`
`Please type the name of the certificate in the database:`	`SSLCertNicknname`
5 - `Do you have a certmap.conf (Should MMP do client-cert auth) (y/n):` If you answer yes, the following prompts are displayed:	`CertMapFile`
`Please type in the location of the certmap.conf file:`	`CertMapFile`
`Who should the MMP authenticate as for the proxy-auth?`	`StoreAdmin`
`What's the password for StoreAdmin?`	`StoreAdminPass`
`Should the MMP pre-authenticate clients? (y/n)`	`PreAuth`

Directly Setting Configuration Variables (Unix)

On Unix, Multiplexor configuration parameters are stored in the configuration files `ImapMMP.config` and `PopMMP.config`. These files are stored in the directory *`MMPRoot/MMP_instanceName`*. You can set configuration parameters by directly modifying the parameter specifications in these files.

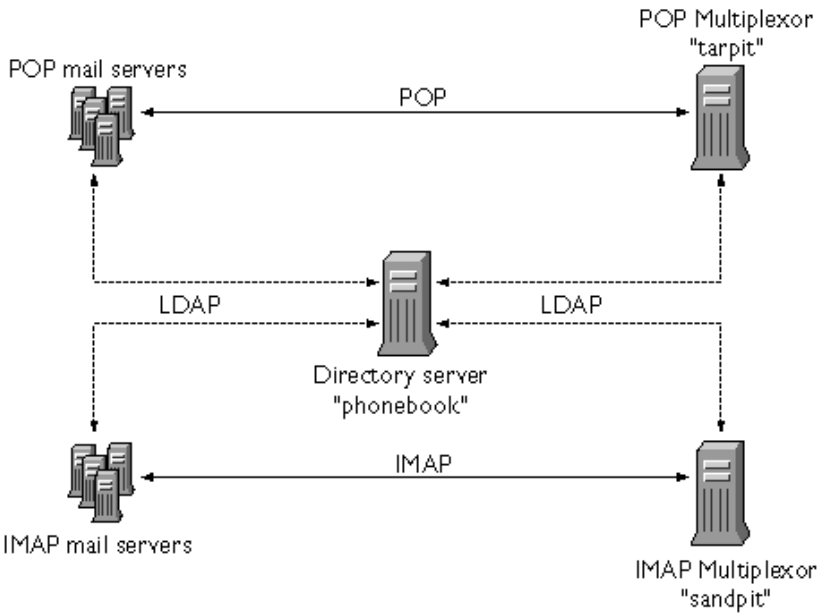
For example, on Unix you can specify the `Capability` configuration parameter by adding it as an environment variable to the `ImapMMP.config` file. For example, you would add this line:

```
Capability="IMAP IMAP4rev1 AUTH=LOGIN AUTH=PLAIN X-NETSCAPE"
```

Example Messaging Topology

In the example shown in Figure 13.3 , the fictional Airius Corporation has two Multiplexors (on separate machines), each supporting several Netscape Messaging Servers. POP and IMAP user mailboxes are split across the Messaging Server machines, with each server dedicated exclusively to POP or exclusively to IMAP. (You can restrict client access to POP services alone by removing the IMAP-server binary; likewise, you can restrict client access to IMAP services alone by removing the POP-server binary.) Each Multiplexor also supports only POP or only IMAP. The LDAP directory service is on a separate, dedicated machine.

Figure 13.3 Example Multiplexor Topology



IMAP Configuration Example

The IMAP Multiplexor in Figure 13.3 is installed on `sandpit`, a machine with 2 processors. This Multiplexor is listening to the standard port for IMAP connections (143). Multiplexor communicates with the LDAP server on the host `phonebook` for user mailbox information, and it routes the connection to the appropriate IMAP server. It overrides the IMAP capability string, provides a virtual domain file, and supports SSL communications.

This is its `ImapMMP.config` configuration file:

```
LDAPHost=phonebook
BaseDN="o=Airius.com"
LogDir="/usr/netscape/suitespot/mmp/MMP_sandpit/log"
LogLevel=5
BindDN="cn=Directory Manager"
BindPassword="password"
NumThreads=2
ListenPort=143
BacksidePort=143
Capability="IMAP4 IMAP4rev1 AUTH=LOGIN AUTH=PLAIN X-NETSCAPE"
VirtualDomainFile="/usr/netscape/suitespot/mmp/vdfile.txt"
SSLEnable=y
SSLListenPort=993
SSLSecmodFile="/usr/netscape/suitespot/mmp/secmod.db"
SSLCertFile="/usr/netscape/suitespot/mmp/cert7.db"
SSLKeyFile="/usr/netscape/suitespot/mmp/key3.db"
SSLKeyPasswd=""
SSLCipherSpecs="all"
SSLCertNickname="Airius.com Server Cert"
```

POP Configuration Example

The POP Multiplexor example in Figure 13.3 is installed on `tarpit`, a machine with 4 processors. This Multiplexor is listening to the standard port for POP connections (110). Multiplexor communicates with the LDAP server on the host `phonebook` for user mailbox information, and it routes the connection to the appropriate POP server. It also provides a spoof message file.

This is its `PopMMP.config` configuration file:

```
LDAPHost=phonebook
BaseDN="o=Airius.com"
LogDir="/usr/netscape/suitespot/mmp/MMP_tarpit/log"
LogLevel=10
BindDN="cn=Directory Manager"
BindPassword="password"
NumThreads=4
ListenPort=
BacksidePort=
SpoofMessageFile="/usr/netscape/suitespot/mmp/pop3spoof.txt"
SSLEnable=n
```

Installing and Configuring Multiplexor (NT)

Messaging Multiplexor is available as part of Netscape Messaging Server. You can install Multiplexor at the same time as you install Messaging Server, or you can install it later using the Server Setup program. Either way, you first need to prepare the system to support Multiplexor.

Before You Install (NT)

Before you install Messaging Multiplexor on a Windows NT machine, perform the following tasks:

1. Choose the machine on which you will install Multiplexor. Netscape recommends against installing Multiplexor on a system that is also running Messaging Server or Directory Server. It is best to use a separate machine for Multiplexor.

2. Check that the system meets all the hardware and software requirements for using Netscape Messaging Server. For more information about installation requirements, see the *Messaging Server Installation Guide*.

3. On the machine that Messaging Multiplexor is to be installed on, create a new user to be used exclusively by Multiplexor. This new user must belong to a group. Suggested names for the user are `nsmmp` ("Netscape Messaging Multiplexor") or `nsmail`.

4. Set up the LDAP directory server and its host machine if it's not already set up.

5. If you already have an older version of Multiplexor installed and want to replace it, you must remove the old Multiplexor before you can install the new one. To remove it, open a command prompt window, change to the directory containing Multiplexor, and type the following commands:

```
c:\MultiPlex\ImapProxy uninstall
c:\MultiPlex\PopProxy uninstall
```

Multiplexor Files (NT)

Multiplexor installation and executable files are initially installed in *server-root*\bin\mmp. (In this document, *server-root* refers to the Messaging Server root directory, for example, C:\Netscape\Server4.)

When you install Multiplexor you also create one or more Multiplexor instances. It is the instances that are actually run. When a Multiplexor instance is installed, an instance directory is created immediately under the *server-root* using the naming convention of: mmp-*instancename*. All executable files needed to run that instance are then copied into the instance directory from *server-root*\bin\mmp. The instance is then run from the instance directory. (In other words, the files in *server-root*\bin\mmp are never run; they are just copied into instance directories.)

For example, if you create an instance and give it the name mplex1, its files are stored in the directory: *server-root*\mmp-mplex1. And the mplex1 instance is run from the files in its instance directory.

Table 13.8 lists the principal files that make up a Messaging Multiplexor installation in a Windows NT environment.

Table 13.8 Multiplexor files (NT)

File	Description
PopProxy.exe, ImapProxy.exe	The executable Messaging Multiplexor programs for POP and IMAP services, respectively.
libldap32v30.dll, libnspr21.dll, libplc21.dll, libplds21.dll, libsh.dll	Shared libraries used by Messaging Multiplexor.

Multiplexor Installation (NT)

In NT environments, there are two different ways to install Messaging Multiplexor on a machine:

- When you install Messaging Server or other Netscape components, the Server Setup program gives you the option of choosing to install Messaging Multiplexor at the same time. You can also use the Server Setup program to

install a Multiplexor instance at a later time. See "Creating a Multiplexor Instance (NT)" on page 388 for details. (For instructions on using the Server Setup program, see the *Messaging Server Installation Guide*.)

- Once Multiplexor has been installed on a machine, you can create additional Multiplexor instances by running the `PopProxy` (for a POP instance) or `ImapProxy` (for an IMAP instance) programs. See "Creating Additional Instances (NT)" on page 389 for details.

Note: Netscape recommends that Multiplexor not be installed on a machine that is also running either Messaging Server or Directory Server.

Creating a Multiplexor Instance (NT)

You must use the Server Setup program to create your first Multiplexor instance. (For subsequent installations of additional instances, you can re-run the Server Setup program or use the `ImapProxy` or `PopProxy` commands.)

Follow these steps to install and configure your first Messaging Multiplexor on a Windows NT system:

1. Run the Server Setup program (`setup.exe`), as described in the *Messaging Server Installation Guide*.

   When you run the Server Setup program, you are asked to choose between three types of installation: Express, Typical, and Custom. If you choose Express, you are asked to specify a minimum number of variables and all other variables are automatically set to the default. If you choose Custom, you are asked to specify all values. If you choose Typical the number of variables you are asked to specify is greater than with an Express installation, but fewer than with a Custom installation.

2. When the Select Components screen is displayed, select Netscape Messaging Suites and click the Change button.

   The Select Sub-Components window is displayed.

3. Unselect the server itself and select Netscape Messaging Multiplexor.

   The Messaging Multiplexor Upgrade window is displayed.

4. Select Create a new multiplexor instance.

   The Messaging Multiplexor Instance window is displayed.

Note: The "Upgrade all existing multiplexor instances" selection in the Messaging Multiplexor Upgrade window is for repairing or upgrading Multiplexor software, not for re-configuring an instance. If you choose this selection, all currently running Multiplexor instances are halted, the software files are recopied into each instance's directory, and those instances that were running at the time you chose to upgrade are restarted.

5. Enter information about your installation and configuration in response to the windows and prompts as they are displayed.

   The windows and prompts you see will vary depending on the kind of installation you chose (Express, Typical, or Custom) and the choices you make as you proceed. See "Configuration Prompts (NT)" on page 391 for a description of these windows and prompts.

6. When you see the summary of the information you've entered, make sure the information is correct, and then click Next.

   The program files are now copied to your hard disk.

7. The setup program then runs each of the Multiplexors (first POP, then IMAP) in a special configuration mode to configure them according to the information you have entered.

Once the installation is complete, open the Services Control Panel to start the new Multiplexor services. The new services for POP and IMAP appear as "Netscape POP3 MMP" and "Netscape IMAP4 MMP," respectively, with the instance name in parentheses.

Creating Additional Instances (NT)

To install an additional instance of Multiplexor on a machine that already has an installed instance, you can either:

- Re-run the Netscape Servers installation program as described in "Creating a Multiplexor Instance (NT)" on page 388. Enter a new instance name in the Messaging Multiplexor Instance window.
- Create a directory for the new instance under the *`server-root`* directory (*`server-root`*`\mmp-`*`instancename`*). Copy all of the files in *`server-root`*`\bin\mmp` to the new instance directory. Then run the `PopProxy` (for a POP instance) or `ImapProxy` (for an IMAP instance) program from

the command line with the `install` option. See "Command-Line Configuration Options" on page 366 for additional details. Note that each instance name must be unique.

Modifying an Instance (NT)

To make permanent configuration changes to a Multiplexor instance, you can:

- Uninstall and then re-install Multiplexor.
- Go to the directory containing the instance you want to modify and run the `PopProxy` or `ImapProxy` programs from the command line with the `install` option as described in "Command-Line Configuration Options" on page 366.
- Modify the information in the NT Registry as described in "Directly Specifying Configuration Variables (NT)" on page 395.

Multiplexor Configuration (NT)

Multiplexor is controlled by setting the configuration parameters that are described in Table 13.3 on page 360. See Table 13.2 on page 358 for the different methods you can use to set these configuration parameters.

In a Windows NT environment, Multiplexor configuration environment variables are stored in the NT Registry. When the Multiplexor program executes, it receives its configuration information from the environment variables in the Registry. As an alternative, you can also directly specify configuration options when running Multiplexor from the command line. When specified on the command line, a given configuration option overrides any value stored for that parameter in the Registry.

This section describes how to set Multiplexor configuration variables by:

- Using the configuration prompts invoked by running the Server Setup program. See "Configuration Prompts (NT)" on page 391.
- Directly editing the Windows NT Registry. See "Directly Specifying Configuration Variables (NT)" on page 395.

You can also set Multiplexor configuration variables by running the `PopProxy` and `ImapProxy` commands with the `install` option as described in "Command-Line Configuration Options" on page 366.

Listing Options (NT)

To display a list of the command-line configuration options stored in the Registry, you can execute Multiplexor for either POP or IMAP from the command line, using the `-h` option with no attributes:

```
PopProxy -h
ImapProxy -h
```

Configuration Prompts (NT)

Table 13.9 lists the Multiplexor configuration prompts and associated configuration parameters for Windows NT. (Table 13.3 on page 360 describes the configuration parameters that the prompts specify.) Configuration variables are stored in the Windows NT registry key: `HKEY_LOCAL_MACHINE/SOFTWARE/Netscape/MMP/`*`instance`*.

Note: The installation windows that you see are determined by the kind of installation you choose to perform: Express, Typical, or Custom, and by the choices you make. If you do not see a particular window or prompt shown below, it is because that window is not displayed for the type of installation you are doing.

Table 13.9 Multiplexor configuration prompts (NT)

Window and prompt	Environment variable
`Enter the instance name you wish to use for this Messaging Multiplexor`	
`Instance Name:`	
`Specify the connection information for the LDAP server you wish to use.`	
`Host Name:`	`LDAPHost`
`Server Port:`	`ListenPort`
`Base Distinguished Name:`	`BaseDN`

Table 13.9 Multiplexor configuration prompts (NT) (Continued)

Window and prompt	Environment variable
Enter the response you want the Messaging IMAP Multiplexor to respond with when queried about its IMAP4 capabilities.	
IMAP Capability Response:	Capability
For security reasons you may wish to specify an account for the Messaging Multiplexor to use when accessing the LDAP service.	
Bind as anonymous	BindDN
Bind as this account:	BindDN
LDAP Bind DN:	BindDN
Password:	BindPass
Please choose a directory path to place log files created by the Messaging Multiplexor.	
Log Directory:	LogDir
Enter the banners you wish the Messaging Multiplexor to display when a client connects.	
POP3 Banner:	Banner
IMAP Banner:	IMAP Banner
The Messaging Multiplexor will issue the following LDAP search to try to find a user who is attempting to log in. You may change the search if you wish.	
LDAP Search:	UidSearch

Table 13.9 Multiplexor configuration prompts (NT) (Continued)

Window and prompt	Environment variable
The Messaging Multiplexor supports Virtual Domain mapping. To do virtual domain mapping, enter the file name here.	(See Table 13.1 on page 355 for information on virtual domain variables)
Do not use Virtual Domain Mapping:	VDomain
Use Virtual Domain Mapping:	VDomain
Virtual Domain File:	VirtualDomainFile
Do you want the IMAP Multiplexor to use Secure Socket Layer (SSL) for its connections?	
Do not allow IMAP over SSL:	SSLenable
Allow IMAP over SSL:	SSLenable
IMAP4 SSL Port:	SSLlistenPort
Enter the incoming and outgoing ports the Messaging POP3 and IMAP4 Multiplexors should use.	
POP3 Ports:	
Incoming:	ListenPort
Outgoing:	BacksidePort
IMAP Ports:	
Incoming:	ListenPort
Outgoing:	BacksidePort
Enter the location of the Certificate Database that the Messaging IMAP4 Multiplexor will use.	
Certificate Database File:	SSLCertFile
Enter the name of a server certificate the Messaging IMAP Multiplexor should use. The Multiplexor will use the certificate with the given name from the server certificate database.	
Certificate Name:	SSLCertNickname

Table 13.9 Multiplexor configuration prompts (NT) (Continued)

Window and prompt	Environment variable
SSL uses a set of widely-known cipher algorithms to encrypt sessions. The client and server agree to one of them. If you do not want to support a specific cipher, unselect it from the list.	
Ciphers Checkboxes: (Check those that you want to use.)	SSLCipherSecs
Enter the location of the Key Database the Messaging IMAP4 Multiplexor will use. If the Key Database is password protected, enter the password in the field below.	
Key Database File: Password:	SSLKeyFile SSLKeyPasswd
If you have a secmod database click "Yes, I have a secmod database" and enter the location of the database file. Otherwise, click "No, I do not have a secmod database" and click Next to continue.	
No, I do not have a secmod database: Yes, I have a secmod database: Secmod Database File:	SSLSecmodFile SSLSecmodFile SSLSecmodFile
For security reasons, you may want to provide the Messaging POP3 Multiplexor with a "Spoof" message. Enter the file name containing the message here.	
Do not use a "spoof" message: Use a "spoof" message: "Spoof" message file:	SpoofMessageFile SpoofMessageFile SpoofMessageFile

Directly Specifying Configuration Variables (NT)

In a Windows NT environment, you can set configuration environment variables by modifying entries in the NT Registry at:

```
\\HKEY_LOCAL_MACHINE\SOFTWARE\Netscape\MMP
```

For example, to specify IMAP capability with Netscape servers you would add this line:

```
Capability="IMAP IMAP4rev1 AUTH=LOGIN AUTH=PLAIN X-NETSCAPE"
```

Running Multiplexor

Once a Multiplexor instance has been installed it must be configured before it can be run. There are two basic ways of setting configuration parameters for Multiplexor:

- **Environment variables.** You can specify and store configuration environment variables prior to running Multiplexor. When you run Multiplexor, it uses these environment variables as its configuration parameters. For general information about setting configuration parameters, see "Multiplexor Configuration" on page 357; for platform specific information, see "Multiplexor Configuration (Unix)" on page 378 and "Multiplexor Configuration (NT)" on page 390; for a description of available configuration parameters, see Table 13.3 on page 360.
- **Command-line options.** You can specify configuration parameters as command-line options at the time you run Multiplexor. When you use command-line options, they are used instead of any corresponding environment variable for that session of Multiplexor only. For more information, see "Command-Line Configuration Options" on page 366; for a description of available options, see Table 13.4 on page 367.

Running Multiplexor (Unix)

To run an installed Multiplexor instance in a Unix environment, you can use either the `PopProxy` command or `PopMMP` script to start up POP service, or either the `ImapProxy` command or the `ImapMMP` script to start up IMAP service.

Using PopProxy and ImapProxy (Unix)

When you start Multiplexor with either the `PopProxy` or `ImapProxy` commands, you can specify configuration options on the command line. Configuration parameters that you do not specify on the command line are taken from previously set environment variables or set to default values.

The `PopProxy` or `ImapProxy` commands are stored in *`MMPRoot`*`/bin`.

To run Multiplexor for POP service in a Unix environment, use this syntax:

```
PopProxy [options]
```

To run Multiplexor for IMAP service, use this syntax:

```
ImapProxy [mode] [options]
```

For both POP and IMAP, *`options`* are optional configuration parameters as described in Table 13.4 on page 367.

Using PopMMP and ImapMMP (Unix)

You can start Multiplexor by running either the `PopMMP` or `ImapMMP` scripts. These scripts can be placed in your `init` directory for automatic startup, or used in the same way as any other Unix command scripts.

When you start Multiplexor with either the `PopMMP` or `ImapMMP` scripts, configuration parameters are taken from the corresponding `PopMMP.config` and `ImapMMP.config` files. You do not directly specify command-line options with the scripts. You specify the configuration variables in the `config` files by running the `mmp-setup` program as described in "Configuration Prompts (Unix)" on page 379 or by directly editing the `config` file as described in "Directly Setting Configuration Variables (Unix)" on page 383.

The `PopMMP` and `ImapMMP` scripts and the `PopMMP.config` and `ImapMMP.config` files are all stored in *`MMPRoot`*`/MMP_`*`instanceName`*.

To run Multiplexor for POP service in a Unix environment, use this syntax:

```
PopMMP
```

To run Multiplexor for IMAP service, use this syntax:

```
ImapMMP
```

Running Multiplexor (NT)

The preferred method of running an installed Multiplexor instance in a Windows NT environment is with the Service Control Manager in the Windows Control Panel. You can use the Service Control Manager to start and stop Multiplexor instances.

The Service Control Manager runs each instance as it has been configured. However, if you want to override an instance's configuration options, you can start it up with either the `PopProxy` or `ImapProxy` command and specify different options on the command line. Configuration parameters that you do not specify on the command line are taken from previously set environment variables as stored in the Windows Registry or set to default values.

The `PopProxy` or `ImapProxy` commands are stored in the Multiplexor instance directory. Only run these commands from the instance directory; do not run these commands from *`server-root`*`\bin\mmp`.

To run Multiplexor from the command line for POP service in an NT environment, use this syntax:

```
PopProxy start [options]
```

To run Multiplexor from the command line for IMAP service, use this syntax:

```
ImapProxy start [options]
```

For both POP and IMAP, *`options`* are optional configuration parameters as described in Table 13.4 on page 367.

You can use the `PopProxy` or `ImapProxy` command to change the Service Control Manager as follows:

```
PopProxy service [options]
```

```
ImapProxy service [options]
```

For both POP and IMAP, *service* is one of the options shown in Table 13.10:

Table 13.10 Command-line service control options (NT)

Option	Description
`start`	Starts running Multiplexor service. May be used with command-line options.
`stop`	Stops Multiplexor. Command-line options have no effect when used with `stop`.
`install`	Writes any command-line options into the Registry so that they become environment variables. Sets the Service Control Manager to automatically run Multiplexor whenever the system starts up.
`refresh`	Instructs a running instance of Multiplexor to reload configuration parameters from the Registry or as specified with command-line options.
`uninstall`	Removes this instance of Multiplexor. Command-line options have no effect when used with `uninstall`.

Uninstalling Multiplexor

This section describes how to remove Multiplexor instances.

Removing Multiplexor (Unix)

To uninstall Multiplexor in Unix environments:

1. Remove `CONFIG_FILE` from `/etc/netscape.mmp.conf`.
2. Remove the *MMPRoot* directory and its subdirectories.

Caution: This removes *all* Multiplexor instances. You cannot remove selected instances, you can only uninstall all instances.

Removing Multiplexor (NT)

To uninstall all Multiplexor instances on a given machine in a Windows NT environment:

1. From the command line, go to the *server-root* directory:

   For example, `cd c:\netscape\server4`

2. Enter the following command:

   ```
   uninstall
   ```

The `uninstall` command will remove all Multiplexor instances under that server-root.

To uninstall a single Multiplexor instance on a given machine:

1. From the command line, go to the directory of the instance you want to uninstall

   For example, `cd netscape\server4\mmp-mplex1`.

2. Enter the following commands:

   ```
   ImapProxy uninstall
   PopProxy uninstall
   ```

This removes that Multiplexor instance. Other instances are not affected.

Appendix

A

Command-line Utilities

Netscape Messaging Server provides a set of command-line utilities in addition to its graphical user interface. This appendix describes utilities for Messaging Server installation, migration, starting, stopping, administration, message management, problem recovery, monitoring, and reporting.

Each command-line utility has a set of options. Please note that these options might change in future releases as product functionality evolves.

This appendix includes the following sections:

- Overview of Command-Line Utilities
- Command-Line Utilities—General Information
- Messaging Server Utilities—Descriptions
- Alarm Attributes

There is a separate set of utilities used to migrate from a Unix `sendmail` messaging environment to Netscape Messaging Server 4.x. The `sendmail` utilities are described in Appendix B. The Messaging Multiplexor and Mailstone components also have their own set of utilities. The Messaging Multiplexor utilities are described in Chapter 13. The Mailstone utilities are described in the Mailstone document.

Overview of Command-Line Utilities

This overview briefly describes the Netscape Messaging Server command-line utilities. This section contains two tables:

- Table A.1 groups the installation, message management, problem recovery, monitoring, and reporting utilities into categories according to their function and purpose.
- Table A.2 describes the installation, message management, problem recovery, monitoring, and reporting utilities in alphabetical order. For complete information about each utility, see "Messaging Server Utilities—Descriptions" on page 405.

Table A.1 Command-line utilities by category

Category	Command-line Utilities
Installation and migration	`MoveUser, qconvert, upgrade`
Management	`configutil, imscripter, mboxutil, NscpMsg, processq`
Recovery	`deliver, reconstruct`
Background and daily tasks	`stored`
Monitoring and reporting	`counterutil, hashdir, mailq, quota, readership`

Table A.2 Command-line utilities and what they do

Command	What it does
`configutil`	Displays and makes changes to configuration information stored in the Directory Server or local configuration file.
`counterutil`	Displays all counters in a counter object. Monitors a counter object.
`deliver`	Delivers mail to a message store accessible by IMAP or POP.
`hashdir`	Identifies the directory that contains the message store for a particular user.

Table A.2 Command-line utilities and what they do (Continued)

Command	What it does
`imscripter`	The IMAP server protocol scripting tool. Executes a command or sequence of commands.
`mailq`	Checks the mail queue and reports the number of messages awaiting delivery.
`mboxutil`	Lists, creates, deletes, renames, or moves mailboxes.
`MoveUser`	Moves messages in a user's mailbox from one Messaging Server to another.
`NscpMsg` (Unix)	Starts and stops Messaging Server and runs recovery utilities.
`processq`	Manually delivers queued messages from the mail queue.
`qconvert`	Converts the Netscape Messaging Server 3.x message queue to the 4.x format.
`quota`	Reports quota usage.
`readership`	Collects readership information on mailboxes.
`reconstruct`	Reconstructs mailboxes that have been damaged or corrupted.
`stored`	Performs background and daily tasks, expunges, and erases messages stored on disk.
`upgrade`	Converts Messaging Server mailboxes stored in 3.x format on a 3.x server to mailboxes in 4.x format on a 4.x server.

Command-Line Utilities—General Information

This section provides general information about using Netscape Messaging Server command-line utilities.

Tip: A note about internationalization: If these utilities do not work correctly in your native environment, check the setting of the LANG environment variable.

Messaging Server File Locations

Messaging Server files are located in the following locations:

Table A.3 Command-line utilities location

Location	Utilities
server-root/bin/msg/admin/bin	configutil, counterutil, deliver (NT), hashdir, imscripter, mailq (NT), mboxutil, MoveUser, NscpMsg, processq (NT), qconvert, quota, readership, reconstruct, stored, upgrade
server-root/bin/msg/store/bin	deliver (Unix)
/bin/mailq	mailq (Unix)
/usr/lib/processq	processq (Unix)

Location of Configuration Data

All user and group configuration information is stored on the LDAP Directory Server.

Most Messaging Server configuration data is also stored on the LDAP server. Some Messaging Server configuration information is stored in a local file named `configdb` on the Messaging Server. The `configdb` file contains the following kinds of configuration information:

- Start up (bootstrap) information needed to locate the configuration data on the LDAP Directory Server.
- Host and server names and types.
- Directory locations.
- Installation information. For example, in Unix environments the UID and GID the server is run as.

The `configdb` file is located as follows:

- **Windows NT environments:** Messaging Server looks for the local `configdb` file in the directory specified in the registry. By default, that directory is *`server-root`*`\msg-`*`instance`*`\config`.
- **Unix environments:** Messaging Server looks for the local `configdb` file in the directory specified by the `CONFIGROOT` environment variable.

Usage Requirements

Most of the command-line utilities must be run locally on the messaging server. See the description of each utility for exceptions and command-specific login requirements.

Unix Login Requirements

In Unix environments, command-line utilities can only be run as `root` or as the login ID originally specified at the time of installation. By default, that is the `mailsrv` login ID. In other words, you have to log in as `mailsrv` to run these utilities. When running commands as `root` in some Unix environments, you must precede the command with `./` to specify the path. For example, `./mboxutil`.

Windows NT Login Requirements

In Windows NT environments, the default is that command-line utilities are run from the same account that is used to run services. By default, that is the Administrator account. Therefore, you must have administrator privileges to run most of the command-line utilities.

Messaging Server Utilities—Descriptions

This section describes what the main Netscape Messaging Server command-line utilities do, defines their syntax, and gives examples of how they are used. The utilities are listed in alphabetical order.

configutil

The `configutil` utility enables you to list and change Messaging Server configuration options.

Most Messaging Server configuration options and values are stored in the LDAP database on Directory Server with the remaining options and values stored locally on Messaging Server in a file named `configdb` (for details, see "Location of Configuration Data" on page 404).

Note: If the administrator has defined any language-specific options (such as messages), you must use the `language` option at the end of the command in order to list or change them. Commands entered without a `language` option are only applied to attributes that do not have a specified language parameter.

configutil Syntax

```
configutil [-f configdbfile] [command-options] [;language]
configutil -i < inputfile
```

Requirements: Must be run locally on the Messaging server.
Location: *server-root*/bin/msg/admin/bin

You can use `configutil` to perform four tasks:

- Display particular configuration options using `-o` *option*.
  - Add `;lang-`*xx* after the option to list options with a specified language parameter. For example, `;lang-jp` to list options specified for the Japanese language.
- List configuration option values using the `-e`, `-l`, or `-p` *prefix* options.
  - Use `-e` to include configuration options with empty values in the list.
  - Use `-l` to just list local configuration options from the server's local configuration file.
  - Use `-p` *prefix* to just list those configuration options whose names begin with the letters specified in *prefix*.
- Set configuration options using the `-o` *option* and `-v` *value* options.
  - Include the `-l` option with `-o` *option* and `-v` *value* to store the new value in the server's local configuration file.

  - To read the actual value from `stdin`, specify a dash (`-`) as the *`value`* on the command line.
  - Add `;lang-`*`xx`* after the option to set options for a specified language parameter. For example, `;lang-jp` to set options specified for the Japanese language.

- Import configuration option values from `stdin` using the `-i` option.
  - Include the `-e` option with the `-i` option to import configuration options even if the value of the configuration option is empty.
  - Include the `-l` option with the `-i` option to import all configuration options to the server's local configuration file.

Table A.4 configutil options

Option	Description
`-e`	Lists configuration options that do not have values specified. May be used with the `-l`, `-p`, and `-i` options.
`-f` *`configdbfile`*	Enables you to specify a local configuration file other than the default. (This option uses information stored in the CONFIGROOT environment variable by default.)
`-i <` *`inputfile`*	Imports configurations from a file. Data in the file to be entered in *`option`*`|`*`value`* format with no spaces on either side of the pipe. If you use `-e` with `-i`, and specify an option without a value, any existing value for that option is deleted. (If you do not use `-e`, when you specify an option without a value, no change is made to any existing value for that option.) Note that a Unix command line like `cat` *`inputfile`* `| configutil -i` is not valid syntax.
`-l` *`option`*	Lists configuration options stored in the local server configuration file. When used in conjunction with the `-v` option, specifies that a configuration option value be stored in the local server configuration file.
`-o` *`option`*	Specifies the name of the configuration option that you wish to view or modify. May be used with the `-l` and `-i` options. Configuration option names starting with the word `local` are stored in the local server configuration file.

Table A.4 configutil options (Continued)

Option	Description
`-p` *`prefix`*	Lists configuration options with the specified prefix.
`-v` *`value`*	Specifies a value for a configuration option. To be used with `-o` *`option`*. If the `-l` option is also specified or the configuration option name specified with the `-o` option begins with `local`, the option value is automatically stored in the local server configuration file rather than the Directory Server.

If you specify no command-line options, all configuration options are listed.

Using the getconf and setconf Scripts

You can use the `getconf` and `setconf` scripts to more easily perform simple `configutil` tasks. These scripts use the following syntax to call `configutil` to perform these tasks:

To display the entire configuration database:

```
getconf
```

To display the value of a specified option:

```
getconf option
```

To set a specified option to a specified value:

```
setconf option value
```

To clear a specified option's value (that is, reset the option so that it has no value):

```
setconf option ""
```

For example, to set the value of the `service.smtp.port` configuration option to 25:

```
setconf service.smtp.port 25
```

To use `setconf` to set an SMTP banner with the text: "Welcome to Airius"

```
setconf service.smtp.banner "Welcome to Airius"
```

To use `setconf` to clear an SMTP banner:

```
setconf service.smtp.banner ""
```

You cannot use prefix pattern matching or language-specific values with these scripts.

Examples of Ways to Use configutil

To list all configuration options and their values in the both the Directory Server LDAP database and local server configuration file:

```
configutil
```

To import configurations from an input file named `config.cfg`:

```
configutil -i < config.cfg
```

To list all configuration options with the prefix `service.imap`:

```
configutil -p service.imap
```

To list all configuration options with the prefix `service.imap`, including those with empty values:

```
configutil -e -p service.imap
```

To display the value of the `service.smtp.port` configuration option:

```
configutil -o service.smtp.port
```

To set the value of the `service.smtp.port` configuration option to 25:

```
configutil -o service.smtp.port -v 25
```

To clear the value for the `service.imap.banner` configuration option:

```
configutil -o service.imap.banner -v ""
```

For information on using `configutil` to set alarm attributes, see "Alarm Attributes" on page 437.

Language Specific Options

To list or set options for a specific language, append `;lang-`*`xx`* immediately after the option with no spaces, where *`xx`* is the two-letter language identifier. For example, to view the text of the Japanese version of the `store.quotaexceededmsg` message:

```
configutil -o "store.quotaexceededmsg;lang-jp"
```

counterutil

The `counterutil` utility displays and changes counters in a counter object. It can also be used to monitor a counter object every 5 seconds.

counterutil Syntax

`counterutil -o` *`counterobject`* `[-r` *`registryname`*`]`

Requirements: Must be run locally on the Messaging server.
Location: *`server-root`*`/bin/msg/admin/bin`

Table A.5 counterutil options

Option	Description
`-o` *`counterobject`*	Continuously display the contents of a particular counter object every 5 seconds.
`-r` *`registryname`*	Indicates the counter registry to use. If no *`registryname`* is specified with the `-r` *`registryname`* option, the default is *`server-root`*`/msg-`*`instance`*`/counter/counter.`

Examples of Ways to Use counterutil

To list all counter objects in a given server's counter registry:

```
counterutil
```

To display the content of counter object `imapstat` every 5 seconds:

```
counterutil -o imapstat -r server-root/msg-instance/counter/counter
```

deliver

The `deliver` utility delivers mail directly to the message store accessible by IMAP or POP mail clients.

If you are administering an integrated messaging environment, you can use this utility to deliver mail from another MTA, a `sendmail` MTA for example, to the Messaging Server message store.

deliver Syntax

```
deliver [-l] [-d] [-r address] [-f address] [-m mailbox] [-a authid]
  [-q] [-F flag]...[userid]...
```

Requirements: Must be run locally on the Messaging server; the `stored` utility must also be running.
Location on Unix: *server-root*`/bin/msg/store/bin`
Location on NT: *server-root*`/bin/msg/admin/bin`

Table A.6 deliver options

Option	Description
`-a` *authid*	Specifies the authorization ID of the sender. Defaults to anonymous.
`-d`	This option is recognized by `deliver` in order to maintain compatibility with `/bin/mail`, but it is ignored by `deliver`.
`-F` *flag*	Sets the system flag or keyword flag on the delivered message.
`-f` *address*	Inserts a forwarding path header containing *address*.
`-l`	Accepts messages using the LMTP protocol (RFC 2033).
`-m` *mailbox*	Delivers mail to *mailbox*. • If any user ids are specified, attempts to deliver mail to *mailbox* for each user id. If the access control on a mailbox does not grant the sender the "`p`" right or if the `-m` option is not specified, then this option delivers mail to the inbox for the user ID, regardless of the access control on the inbox. • If no user ids are specified, this option attempts to deliver mail to *mailbox*. If the access control on a mailbox does not grant the sender the "`p`" right, the delivery fails.
`-q`	Overrides mailbox quotas. Delivers messages even when the receiving mailbox is over quota.
`-r` *address*	Inserts a `Return-Path:` header containing *address*.
userid	Deliver to inbox of the user specified by *userid*.

If you specify no options, mail is delivered to the inbox.

Examples of Ways to Use deliver

To deliver the contents of a file named `message.list` to Fred's Tasks mailbox:

```
deliver -m tasks fred < message.list
```

In this example, if the tasks mailbox does not grant "`p`" rights to the sender, the contents of `message.list` are delivered to the inbox of the user `fred`.

hashdir

The `hashdir` utility identifies the directory that contains the message store for a particular account. This utility reports the relative path to the message store, such as `d1/a7/`. The path is relative to the directory level just before the one based on the user ID. The utility sends the path information to the standard output.

hashdir Syntax

```
hashdir [-a] [-i] account_name
```

Requirements: Must be run locally on the Messaging server.
Location: *server-root*`/bin/msg/admin/bin`

Table A.7 hashdir options

Option	Description
`-a`	Appends the directory name to the output.
`-i`	Allows you to use the command in interactive mode.

imscripter

The `imscripter` utility connects to an IMAP server and executes a command or a sequence of commands.

imscripter Syntax

```
imscripter [-h] [-f script | [-c command] -f datafile]] [-c command]
  [-s serverid | -p port | -u userid | -x passwd |-v verbosity]
```

Requirements: May be run remotely.
Location: *server-root*/bin/msg/admin/bin

Table A.8 imscripter options

Option	Description
`-c` *command*	Executes *command*, which can be one of the following: create *mailbox* delete *mailbox* rename *oldmailbox newmailbox* [*partition*] getacl *mailbox* setacl *mailbox userid rights* deleteacl *mailbox userid* If one or more of the above variables are included, the option executes the given command with that input. For example, `create lincoln` creates a mailbox for the user `lincoln`. If the `-f` *file* option is used, the option executes the command on each variable listed in the file.
`-f` *file*	The *file* may contain one or more commands, or a list of mailboxes on which commands are to be executed.
`-h`	Displays help for this command.
`-p` *port*	Connects to the given port. The default is 143.
`-s` *server*	Connects to the given server. The default is `localhost`. The *server* can either a host name or an IP address.
`-u` *userid*	Connects as *userid*.

Table A.8 imscripter options (Continued)

Option	Description
`-v` *`verbosity`*	String containing options for printing various information. The options are as follows. The default is `EPBibo` (`EBb` if `-f` is specified). E - Show errors I - Show informational messages P - Show prompts C - Show input commands c - Show protocol commands B - Show BAD or NO untagged responses O - Show other untagged responses b - Show BAD or NO completion results o - Show OK completion results A - Show all of the above The letters designating options can be entered in any order.
`-x` *`passwd`*	Uses this password.

imscripter File Formats

Data files used with the `-f` option have to be formatted as a list of mailboxes with fully qualified paths, one mailbox per line. For example:

```
shared folders/paco/INBOX
shared folders/desmond/INBOX
shared folders/percival/INBOX
```

Scripts used with the `-f` option have to be in the following format:

```
command options
```

For example, an `imscripter` script file that adds the mailboxes contained in a file named `add.list`, and deletes mailboxes contained in a file named `delete.list`, looks like this:

```
create -f add.list
delete -f delete.list
```

Suppose the example file shown above is named `dothis.script`. To run it showing only errors, input commands, BAD and NO untagged and completion responses, enter:

```
imscripter -v ECBb -f dothis.script
```

Examples of Ways to Use imscripter

To run `imscripter` in interactive mode:

```
imscripter [options]
```

In interactive mode, each command is entered one line at a time at the `imscripter` prompt. The command is then executed and responses displayed.

To execute the commands specified in a script file:

```
imscripter [options] -f script
```

To execute a specific command on *`mailbox`*:

```
imscripter [options] [-l] -c command mailbox
```

To execute a specific command for each line of data in a data file:

```
imscripter [options] -c command -f datafile
```

To execute commands from the specified script file:

```
imscripter -s username.airius.com -f script
```

mailq

The `mailq` utility checks and reports on the queue of messages awaiting delivery.

mailq Syntax

```
mailq [-v]
```

Requirements: Must be locally run on the server.
Location on Unix: `/bin/mailq`
Location on NT: *`server-root`*`/bin/msg/admin/bin`

Table A.9 mailq options

Option	Description
-v	Provides more verbose information about the queue.

mailq Examples

To check the mail queue, type `mailq` at a command prompt. If there are no queued messages, that fact will be reported:

```
% mailq
Mail queue is empty.
```

If there are queued messages, each host that has queued messages waiting to be delivered will be listed, along with the number of pending deliveries. For example, output might look like this:

```
% mailq
Queued Messages       Destination Host
---------------       ----------------
              2         math.marsu.edu
              3   universal-robots.com
```

In the example above, five messages are waiting for delivery. Delivering all of them should require two connections to other machines because the messaging server attempts to deliver all queued mail for a host before disconnecting.

For information how to manually force immediate delivery of messages in the mail queue, see “processq” on page 424.

mboxutil

The `mboxutil` utility lists, creates, deletes, renames, or moves mailboxes (folders).

mboxutil Syntax

```
mboxutil [-c mailbox] [-d mailbox] [-r oldname newname [partition]]
  [-l] [-p pattern] [-x] [-k] [-u]
```

Requirements: Must be run locally on the Messaging server; the `stored` utility must also be running.
Location: *server-root*`/bin/msg/admin/bin`

Mailbox Naming Conventions. You must specify mailbox names in the following format: `user/`*`userid`*`/`*`mailbox`*, where *`userid`* it the user that owns the mailbox and *`mailbox`* is the name of the mailbox. For example, the following command creates the mailbox named `INBOX` for the user whose user ID is `crowe`. `INBOX` is the default mailbox for mail delivered to the user `crowe`.

```
mboxutil -c user/crowe/INBOX
```

Important: The name `INBOX` is reserved for each user's default mailbox. `INBOX` is the only folder name that is case-insensitive. All other folder names are case-sensitive.

Table A.10 mboxutil options

Option	Description
`-c` *`mailbox`*	Creates the specified mailbox.
`-d` *`mailbox`*	Deletes the specified mailbox.
`-k` *`mailbox cmd`*	Locks the specified mailbox at the folder level; runs the specified command; after command completes, unlocks the mailbox. When a mailbox is locked the owner can view the messages it contains, but no new messages can be added and no existing messages can be deleted or moved. You should use the `-k` option before performing backups, for example.
`-l`	Lists all of the mailboxes on a server.

Table A.10 mboxutil options (Continued)

Option	Description
`-p` *`pattern`*	When used in conjunction with the `-l` option, lists only those mailboxes with names that match *`pattern`*. You can use IMAP wildcards.
`-r` *`oldname newname`* `[`*`partition`*`]`	Renames the mailbox from *`oldname`* to *`newname`*. To move a folder from one partition to another, specify the new partition with the *`partition`* option. Note that you cannot rename a user's `INBOX`. Nor can you use `mboxutil -r` to move mail stored under one user ID to another user ID.
`-u`	Lists user information such as current size of mail store, quota (if one has been set), and percentage of quota currently in use.
`-x`	When used in conjunction with the `-l` option, shows the path and access control for a mailbox.

Examples of Ways to Use mboxutil

To list all mailboxes for all users:

```
mboxutil -l
```

To list all mailboxes and also include path and `acl` information:

```
mboxutil -l -x
```

To create the default mailbox named `INBOX` for the user `daphne`:

```
mboxutil -c user/daphne/INBOX
```

To delete a mail folder named `projx` for the user `delilah`:

```
mboxutil -d user/delilah/projx
```

To delete the default mailbox named `INBOX` and *all mail folders* for the user `druscilla`:

```
mboxutil -d user/druscilla/INBOX
```

To rename Desdemona's mail folder from `memos` to `memos-april`:

```
mboxutil -r user/desdemona/memos user/desdemona/memos-april
```

To lock a mail folder named `legal` for the user `dulcinea`:

```
mboxutil -k user/dulcinea/legal cmd
```

where *`cmd`* is the command you wish to run on the locked mail folder.

To move the mail account for the user `dimitria` to a new partition:

```
mboxutil -r user/dimitria/INBOX user/dimitria/INBOX partition
```

where *`partition`* specifies the name of the new partition.

To move the mail folder named `personal` for the user `dimitria` to a new partition:

```
mboxutil -r user/dimitria/personal user/dimitria/personal partition
```

To list usage statistics:

```
mboxutil -u

   quota     use    %use    user
   10240     297    29%     daphne
   no quota  168            delilah
   no quota    0            druscilla
```

Quotas and usage figures are reported in kilobytes.

MoveUser

The `MoveUser` utility moves a user's account from one messaging server to another. When user accounts are moved from one messaging server to another, it's also necessary to move the user's mailboxes and the messages they contain from one server to the other. In addition to moving mailboxes from one server to another, `MoveUser` updates entries in the Directory Server to reflect the user's new `mailhost` name and message store path.

MoveUser Syntax

```
MoveUser -s srcmailhost[:port] -x proxyuser -p password
  [-u uid | -u uid -U newuid] [-l ldapURL -D bindDN -w password]
  -d destmailhost[:port]   [options]
```

Requirements: May be run remotely.
Location: *`server-root`*`/bin/msg/admin/bin`

Table A.11 MoveUser options

Option	Description
`-a` *`destproxyuser`*	ProxyAuth user for destination messaging server.
`-A`	Do not add an alternate email address to the LDAP entry.
`-d` *`destmailhost`*	Destination messaging server. By default, `MoveUser` assumes IMAP port `143`. To specify a different port, add a semi-colon and the port number after *`destmailhost`*. For example, to specify port `150` for `myhost`, you would enter `-d myhost:150`.
`-D` *`binddn`*	Binding *`dn`* to the given *`ldapURL`*.
`-F`	Delete messages in source messaging server after successful move of mailbox. (If not specified, messages will be left in source messaging server.)
`-h`	Display help for this command.
`-l` *`ldapURL`*	URL to establish a connection with the Directory Server: `ldap://`*`hostname`*`:`*`port/`* *`base_dn?attributes?scope?filter`* For more information about specifying an LDAP URL, see your Directory Server documentation. Cannot be used with the `-u` option.
`-L`	Add a license for Messaging Server if not already set.
`-m` *`destmaildrop`*	Message store path for destination messaging server. (If not specified, the default is used.)
`-n` *`msgcount`*	Number of messages to be moved at once.
`-o` *`srcmaildrop`*	Message store path for source messaging server. (If not specified, the default is used.)
`-p` *`srcproxypasswd`*	ProxyAuth password for source messaging server.

Table A.11 MoveUser options (Continued)

Option	Description
`-s srcmailhost`	Source messaging server. By default, `MoveUser` assumes IMAP port `143`. To specify a different port, add a semi-colon and the port number after *`srcmailhost`*. For example, to specify port `150` for `myhost`, you would enter `-s myhost:150`.
`-S`	Do not set new message store path for each user.
`-u uid`	User ID for the user mailbox that is to be moved. Cannot be used with `-l` option.
`-U newuid`	New (renamed) user ID that the mailbox is to be moved to. Must be used with `-u uid`, where `-u uid`, identifies the old user name that is to be discontinued. Both the old and the new user ID must currently exist on both the source and the destination mailhost. After migration you are free to manually remove the original user ID from LDAP if you wish to do so.
`-v destproxypwd`	ProxyAuth password for destination messaging server.
`-w bindpasswd`	Binding password for the *`binddn`* given in the `-D` option.
`-x srcproxyuser`	ProxyAuth user for source messaging server.

ProxyAuth User

The ProxyAuth user is the user given the privilege to run the `MoveUser` command. In older Messaging Server 3.x environments the user ID of the Proxyauth user was specified in the file: `/etc/netscape.mail.conf`. In Messaging Server 4.x environments, any valid Message Store Administrator can be used to run the `MoveUser` command as the ProxyAuth user. The list of Message Store Administrators can be viewed and edited from the Netscape Console. For more information, see “Specifying Administrator Access to the Store” on page 154.

Examples of Ways to Use MoveUser

To move all users from *host1* to *host2*, based on account information in the Directory Server `airius.com`:

```
MoveUser -l \
  "ldap://airius.com:389/o=Airius.com???(mailhost=host1.domain.com)" \
  -D "cn=Directory Manager" -w password -s host1 -x admin -p password \
  -d host2 -a admin -v password
```

To move one user from *host1* which uses port 150 to *host2*, based on account information in the Directory Server `airius.com`:

```
MoveUser -l \
  "ldap://airius.com:389/o=Airius.com???(uid=userid)" \
  -D "cn=Directory Manager" -w password -s host1:150 -x admin \
  -p password -d host2 -a admin -v password
```

To move a group of users whose uid starts with letter 's' from `host1` to `host2`, based on account information in the Directory Server `server1.airius.com`:

```
MoveUser -l \
  "ldap://server1.airius.com:389/o=Airius.com???(uid=s*)" \
  -D "cn=Directory Manager" -w password -s host1 -x admin \
  -p password -d host2 -a admin -v password
```

To move a user's mailboxes from `host1` to `host2` when the user ID of `admin` is specified in the command line:

```
MoveUser -u uid \
  -s host1 -x admin -p password \
  -d host2 -a admin -v password
```

To move a user named `aldonza` from `host1` to a new user ID named `dulcinea` on `host2`:

```
MoveUser -u aldonza -U dulcinea \
  -s host1 -x admin -p password \
  -d host2 -a admin -v password
```

NscpMsg

The `NscpMsg` utility starts, stops, and refreshes the Messaging Server services. This utility is available in Unix environments only.

NscpMsg Syntax

```
/NscpMsg start   [ imap | pop | http | smtp | store | snmpagt ]
/NscpMsg stop    [ imap | pop | http | smtp | store | snmpagt ]
/NscpMsg refresh [ imap | pop | http | smtp | store | snmpagt ]
/NscpMsg list    [ imap | pop | http | smtp | store | snmpagt ]
```

Requirements: Must be run as root locally on the Messaging server.
Location: `/etc`
`server-root``/bin/msg/admin/bin`

The `NscpMsg` utility sets the `CONFIGROOT` and `LD_LIBRARY_PATH` environment variables for the commands it runs.

`NscpMsg` is similar to the `/etc/NscpMail` utility supplied with Messaging Server 3.x.

Table A.12 NscpMsg options

Option	Description
`refresh`	Refresh messaging server processes.
`start`	Starts all messaging server processes (`stored`, `smtpd`, `popd`, `imapd`, `httpd`), or optionally, one specified service.
`stop`	Stops all messaging server processes (`stored`, `smtpd`, `popd`, `imapd`, `httpd`), or optionally, one specified service.
`list`	Lists all messaging server processes.

processq

The `processq` utility immediately delivers messages in the deferred queue.

processq Syntax

`processq [[-R]`*`hostname`*`]`

Requirements: Must be locally run on the server.
Location on Unix: `/usr/lib`
Location on NT: *`server-root`*`/bin/msg/admin/bin`

Table A.13 processq options

Option	Description
`hostname`	Deliver mail addressed to *`hostname`*. The *`hostname`* can be the full name of the host as reported by `mailq`, or any pattern contained in the name. If the pattern matches more than one *`hostname`*, each match will have its queue processed.
`-R`	Specify a specific domain.

The deferred queue is automatically processed at regular intervals, so you normally never need to deliver the queue manually with the `processq` utility. You can also manage the queue by using the Netscape Console. For more information, see "Message Queue Concepts" on page 114.

processq Examples

To deliver all messages in the deferred queue:

```
/usr/lib/processq
```

To deliver all queued mail addressed to `math.marsu.edu`:

```
/usr/lib/processq -Rmath.marsu.edu
```

To deliver all queued mail addressed to all hosts in the domain `marsu.edu`:

```
/usr/lib/processq '*.marsu.edu'
```

qconvert

The `qconvert` utility converts the Netscape Messaging Server 3.x message queue to the Netscape Messaging Server 4.x format.

Examples of Ways to Use qconvert

`qconvert [-s sourceq -d targetq] [-l] [-r] [-h] [-f configdbfile]`

Requirements: Must be run locally on the Messaging server.
Location: *server-root*`/bin/msg/admin/bin`

Table A.14 qconvert options

Option	Description
`-d targetq`	Specifies the path name of the new message queue.
`-h`	Displays help for this command.
`-l`	Writes the results to the screen. If you do not specify this option, `qconvert` writes results to the default log file in the `log/default` directory.
`-r`	Removes messages after conversion.
`-s sourceq`	Specifies the path name of the old message queue.
`-f configdbfile`	Specifies the path name of the configuration database.

If you do not specify the location of the 3.x message queue or the 4.x message queue, the `qconvert` utility reads the 3.x and 4.x configuration files to locate the message queue directories. Consequently, if you do not specify the message queue locations, the following configuration information must be available to the `qconvert` utility.

For Messaging Server 3.x, configuration information is determined as shown in Table A.15:

Table A.15 Messaging Server 3.x configuration information

Unix	Windows NT
`MASTERCONFIG` environment variable `/etc/netscape.mail.conf`	Registry key in HKEY_LOCAL_MACHINE: `SOFTWARE\\Netscape\MessagingServer\\3.0`

For Messaging Server 4.x, configuration information is determined as shown in Table A.16 :

Table A.16 Messaging Server 4.x configuration information

Unix	Windows NT
`CONFIGROOT` environment variable `/etc/nsserver.cfg`	Registry key in HKEY_LOCAL_MACHINE: `SOFTWARE\\Netscape\MessagingServer\\4.0`

The `qconvert` utility reads the 3.x directories in the following order: *`sourceq`*`/control`, *`sourceq`*`/deferred`, *`sourceq`*`/messages`. These directories are subdirectories of the post office directory that was specified at installation time.

The `qconvert` utility reads and converts the 3.x directories as follows:

1. The utility reads the 3.x control files and rewrites them into the 4.x envelope log file located in *`targetq`*`/control/env-`*`date`*`-1`.
2. The utility reads the 3.x `deferred` directory and creates an envelope file for every subdirectory in the 4.x `deferred` directory.
3. The utility merges the 3.x message header file (`-Header`) and message body file (`-Body`) into one 4.x file located in *`targetq`*`/control/`*`msgname`*.

The utility automatically converts 3.x access rights to 4.x access rights so that users have access to the appropriate files.

quota

The `quota` utility reports mailbox quota usage. This utility generates a report listing quotas, giving their limits and usage.

quota Syntax

```
quota [user/user-id...]
```

Requirements: Must be run locally on the Messaging server; the `stored` utility must also be running.
Location: *`server-root`*`/bin/msg/admin/bin`

Table A.17 quota options

Option	Description
`user/`*`user-id`*	The quota listing is limited to quota roots with names that start with one of the given user IDs.

Netscape Messaging Server supports quotas at the root (inbox) level. Quotas on subdirectories are not supported.

readership

The `readership` utility reports on how many users other than the mailbox owner have read messages in a shared IMAP folder.

An owner of a IMAP folder may grant permission for others to read mail in the folder. A folder that others are allowed to access is called a *shared folder*. Administrators can use the `readership` utility to see how many users other than the owner are accessing a shared folder.

This utility scans all mailboxes.

readership Syntax

```
readership [-d days] [-p months]
```

Requirements: Must be run locally on the Messaging server; the `stored` utility must also be running.
Location: *server-root*`/bin/msg/admin/bin`

Table A.18 readership options

Option	Description
`-d` *days*	Counts as a reader any identity that has selected the shared IMAP folder within the indicated number of days. The default is 30.
`-p` *months*	Does not count users who have not selected the shared IMAP folder within the indicated number of months. The default is infinity and removes (prunes) the seen flag data for those users. This option also removes the "seen" flag data for those users from the store.

This utility produces one line of output per shared folder, reporting the number of readers followed by a space and the name of the mailbox.

Each reader is a distinct authentication identity that has selected the shared folder within the past specified number of days. Users are not counted as reading their own personal mailboxes. Personal mailboxes are not reported unless there is at least one reader other than the folder's owner.

reconstruct

The `reconstruct` utility rebuilds one or more mailboxes, or the master mailbox file, and repairs any inconsistencies. You can use this utility to recover from almost any form of data corruption in the mail store. Note that low-level database repair, such as completing transactions and rolling back incomplete transactions is performed with `stored -d`.

reconstruct Syntax

```
reconstruct [-p partition] [-r [mailbox [mailbox...]] | [-m] [-q]
   [-o [-d filename]]
```

Requirements: Must be run locally on the Messaging server; the `stored` utility must also be running.
Location: *server-root*`/bin/msg/admin/bin`

Note: Netscape recommends that you shut down your server before running the `reconstruct` utility—unless advised otherwise by Netscape Technical Support.

Table A.19 reconstruct options

Option	Description
`-m`	Performs a high-level consistency check and repair of the mailboxes database. This option examines every mailbox it finds in the spool area, adding or removing entries from the mailboxes database as appropriate. The utility prints a message to the standard output file whenever it adds or removes an entry from the database.
`-o`	Checks for orphaned accounts. This option searches for inboxes in the current messaging server host which do not have corresponding entries in LDAP. For example, the `-o` option would find inboxes of owners who have been deleted from LDAP or moved to a different server host. For each orphaned account it finds, `reconstruct` writes the command: `mboxutil -d user/`*`userid`*`/INBOX` to the standard output.
`-o -d` *`filename`*	If `-d` *`filename`* is specified with the `-o` option, `reconstruct` opens the specified file and writes the `mboxutil -d` commands into that file. The file may then be turned into a script file to delete the orphaned accounts.
`-p` *`partition`*	Specifies a partition name. You can use this option on the first usage of `reconstruct`.

Table A.19 reconstruct options (Continued)

Option	Description
`-q`	Fixes any inconsistencies in the quota subsystem, such as mailboxes with the wrong quota root or quota roots with the wrong quota usage reported. The -q option can be run while other server processes are running.
`-r [`*`mailbox`*`]`	Performs a consistency check and repairs the partition area of the specified mailbox or mailboxes. The `-r` option also repairs all sub-mailboxes within the specified mailbox. If you specify `-r` with no mailbox argument, the utility repairs the spool areas of all mailboxes within the database.

The *`mailbox`* argument indicates the mailbox to be repaired. You can specify one or more mailboxes. Mailboxes are specified with names in the format *`user/userid/sub-mailbox`*. Where *`userid`* is the user that owns the mailbox. For example, the inbox of the user `dulcinea` is entered as: `user/dulcinea/INBOX`.

Examples of Ways to Use reconstruct

Rebuilding Mailboxes

To rebuild mailboxes, use the `-r` option. You should use this option when:

- Accessing a mailbox returns one of the following errors: “System I/O error” or “Mailbox has an invalid format”.
- Accessing a mailbox causes the server to crash.
- Files have been added to or removed from the spool directory.

For example, to rebuild the spool area for the mailboxes belonging to the user Daphne, use the following command:

```
reconstruct -r user/daphne
```

To rebuild the spool area for all mailboxes listed in the mailbox database:

```
reconstruct -r
```

You must use this option with caution, however, because rebuilding the spool area for all mailboxes listed in the mailbox database can take a very long time for large message stores. (See "reconstruct Performance" on page 432.) A better method for failure recovery might be to use multiple disks for the store. If one disk goes down, the entire store does not. If a disk becomes corrupt, you need only rebuild a portion of the store by using the `-p` option as follows:

```
reconstruct -r -p subpartition
```

To rebuild mailboxes listed in the command line argument only if they are in the `primary` partition:

```
reconstruct -p primary mbox1 mbox2 mbox3
```

If you do need to rebuild all mailboxes in the `primary` partition:

```
reconstruct -r -p primary
```

Checking and Repairing Mailboxes

To perform a high-level consistency check and repair of the mailboxes database:

```
reconstruct -m
```

You should use the `-m` option when:

- One or more directories were removed from the store spool area, so the mailbox database entries also need to be removed.
- One or more directories were restored to the store spool area, so the mailbox database entries also need to be added.
- The `stored -d` option is unable to make the database consistent.

  If the `stored -d` option is unable to make the database consistent, you should:

  1. Shut down all servers.
  2. Remove all files in *`server-root`*`/msg-`*`instance`*`/store/mboxlist`.
  3. Run `stored`.
  4. Run `reconstruct -m` to build a new mailboxes database from the contents of the spool area.

5. After `reconstruct -m` completes, restart the server processes.

Removing Orphaned Accounts

To search for orphaned accounts (orphaned accounts are mailboxes that do not have corresponding entries in LDAP):

```
reconstruct -o
  reconstruct: Start checking for orphaned mailboxes
  mboxutil -d user/test/annie/INBOX
  mboxutil -d user/test/oliver/INBOX
  reconstruct: Found 2 orphaned mailbox(es)
  reconstruct: Done checking for orphaned mailboxes
```

To create a file listing orphaned mailboxes that can be turned into a script file that deletes the orphaned mailboxes, where the file is to be named `orphans.cmd`:

```
reconstruct -o -d orphans.cmd
  reconstruct: Start checking for orphaned mailboxes
  reconstruct: Found 2 orphaned mailbox(es)
  reconstruct: Done checking for orphaned mailboxes
```

reconstruct Performance

The time it takes `reconstruct` to perform an operation depends on a number of factors including:

- The kind of operation being performed and the options chosen
- Disk performance
- The number of folders when running `reconstruct -m`
- The number of messages when running `reconstruct -r`
- The overall size of the message store
- What other processes the system is running and how busy it system is
- Whether or not there is ongoing POP, IMAP, HTTP, or SMTP activity

In one example with approximately 2400 users, a message store of 85GB, and concurrent POP, IMAP, or SMTP activity on the server:

- `reconstruct -m` took about 1 hour
- `reconstruct -r` took about 18 hours

Note: A `reconstruct` operation may take significantly less time if the server is not performing ongoing POP, IMAP, HTTP, or SMTP activity.

stored

The `stored` utility performs the following functions:

- Background and daily messaging tasks
- Deadlock detection and rollback of deadlocked database transactions
- Cleanup of temporary files on startup
- Implementation of aging policies
- Periodic monitoring of server state, disk space, service response times, and so on
- Issuing of alarms if necessary

The `stored` utility automatically performs cleanup and expiration operations once a day at midnight. You can choose to run additional cleanup and expiration operations.

How Messages are Deleted

Messaging Server messages are erased in three stages:

1. **Delete**. A client marks the message to be deleted. This is referred to as *deleting* a message. At this point, the client can restore the message by removing the "deleted" marking.

2. **Expunge**. A client, or the expiration policies you have specified, expunges messages that have been marked deleted from the mailbox. Once messages are expunged, the client can no longer restore them, but they are still stored on disk. (A second client with an existing connection to the same mailbox may still be able to fetch the messages.)

3. **Cleanup**. The `stored` utility erases messages from the disk that have been expunged for at least one hour.

stored Syntax

To run `stored` from the command line to perform a specific operation:

```
stored [-1] [-c] [-n] [-v [-v]]
```

To run `stored` as daemon:

```
stored [-d] [-v [-v]]
```

Requirements: Must be run locally on the Messaging server.
Location: *`server-root`*`/bin/msg/admin/bin`

Table A.20 stored options

Option	Description
`-c`	Performs one cleanup pass to erase expunged messages. Runs once, then exits. The `-c` option is a one-time operation, so you do not need to specify the `-1` option.
`-d`	Run as daemon. Performs system checks and activates alarms, deadlock detection, and database repair.
`-1`	Run once, then exit.
`-n`	Run in trial mode only. Does not actually age or cleanup messages. Runs once, then exits.
`-v`	Verbose output.
`-v -v`	More verbose output.

Examples of Ways to Use stored

To test expiration policies:

```
stored -n
```

To perform a single aging and cleanup pass:

```
stored -l -v
```

If you want to change the time of the automatic cleanup and expiration operations, use the `configutil` utility as follows:

```
configutil -o store.expirestart -v 21
```

Occasionally, you might need to restart the `stored` utility; for example, if the mailbox list database becomes corrupted. To restart `stored` on Unix, use the following commands at the command line:

```
/etc/NscpMsg stop store
/etc/NscpMsg start store
```

To restart `stored` on Windows NT, use the Service Control Manager.

If any server daemon crashes, you must stop all daemons and restart all daemons including `stored`.

upgrade

The `upgrade` utility transfers mailboxes stored in 3.x format on a 3.x server to Netscape Messaging Server 4.x format mailboxes. (For complete information about upgrading your server, see the *Messaging Server Installation Guide.)*

The `upgrade` utility is similar to the `migrate` utility provided by 3.x. The 3.x utility migrated both users and their mailboxes. The Messaging Server `upgrade` utility only transfers mailboxes since the way users are stored has not changed from 3.x to 4.x.

The 4.x `upgrade` utility assumes that both Messaging Server 3.x and Messaging Server 4.x reside on the same machine. The utility transfers the 3.x mailboxes on a machine to 4.x format mailboxes on the same machine. Once `upgrade` has been run, you cannot start up the 3.x processes.

The 4.x upgrade process occurs in two steps: (1) The folders are upgraded. (2) The messages are upgraded. You must update the folders (by using the `-s` option) before you update the messages (by using the `-m` or `-u` options). The message upgrade is a multi-threaded process. You can specify the number of threads by using the `-t` option.

The 4.x `upgrade` utility first searches the LDAP server to find all the user mailboxes in that machine (users are considered to belong the 3.x server if their `mailhost` attribute is one of the `MessageHostNames` in that 3.x server). It then creates a one-to-one mailbox mapping in the 4.x mailbox database. These mailboxes are marked as "`TRANSITION`".

Based on the options you specify, `upgrade` can transfer all 3.x mailboxes immediately. In NT environments, the `upgrade` utility retrieves the 3.x information through the registry. In Unix environments, the `upgrade` utility

retrieves the 3.x information through a predefined configuration file. The `upgrade` does not change the 3.x directory structure, so 4.x must be put in a different directory.

If you have 3.x mailboxes located in non-default mailbox paths, the `upgrade` utility tries to create a 4.x mailbox directory in that mailbox path, and uses a number (such as `001`) to automatically assign the 4.x partition name. In 4.x, the partition name is a logical name of a physical directory where user mailboxes can be physically created.

You can find the detailed mailbox-mapping information in the `upgrade.conf` file in the default 3.x mailbox directory.

If you have servers on multiple machines, you must run `upgrade` on each different machine.

Your mailboxes will require more disk space after the upgrade to 4.x as follows:

- **Single copy off**. If you were running Messaging Server 3.x with Single Copy turned off, your 4.x message store will approximately require 15% more disk space than it did under 3.x.
- **Single copy on**. Messaging Server 4.x uses a new process for handling messages when running in single copy mode that is different from the old 3.x method. As a result, the `upgrade` utility has to write individual copies of all single copy messages to disk in order to transfer them to the 4.x server. If you have many single copy messages with large numbers of recipients, this could require a sizable amount of disk space.

If you want to save disk space, you can use the `-r` option to remove the messages after the upgrade.

upgrade Syntax

```
upgrade [-s] [-m] [-u userlist] [-i] [-r] [-h] [-f configdbfile] [-F]
   [-t number]
```

Requirements: Must be run locally on the Messaging server; the `stored` utility must also be running.
Location: *server-root*`/bin/msg/admin/bin`

Table A.21 upgrade options

Option	Description
`-f configdbfile`	Identifies the filename of the *`configdbfile`* instance.
`-F`	Creates a backup copy of the existing `upgrade4x.conf` file. Use this option if you are upgrading your server, and you want to retain the existing `upgrade4x.conf` file from a previous upgrade. The `upgrade` process will create a new `upgrade4x.conf` file and save the old file as `upgrade4x.conf.bak`.
`-h`	Displays help for this command.
`-i`	Transfer the inbox first. Can be used only with the `-m` option.
`-m`	Transfers the mailboxes immediately. You must create mailbox mapping with the `-s` option before using `-m` for the first time. For example, `-s -m`. This can be used in a script.
`-r`	Remove messages after transfer.
`-s`	Creates the one-to-one mailbox mapping, but delays transfer. This can be used in a script. You must run the `-s` option before you run the `-m` option or the `-u` option.
`-t` *`number`*	Specifies how many threads to start up for upgrading mailboxes. The default number is 5 threads. For optimal performance, do not use more than 10 threads.
`-u` *`userlist`*	Transfers all mail folders belonging to the user or users in specified in *`userlist`*.

Alarm Attributes

Alarm attributes specify how often system checks and other monitoring functions are performed. When a problem is detected an alarm message is sent to the specified person. (System checks are performed by the `stored` utility.)

You can modify the following alarm attributes by using the `configutil` command.

Table A.22 Email alarm attributes

Email Attributes	Default Value
alarm.msgalarmnoticehost	localhost
alarm.msgalarmnoticeport	25
alarm.msgalarmnoticercpt	Postmaster@localhost
alarm.msgalarmnoticesender	Postmaster@localhost

Table A.23 Disk space alarm attributes

Disk Space Attributes	Default Value
alarm.diskavail.msgalarmstatinterval	3600 seconds
alarm.diskavail.msgalarmthreshold	10%
alarm.diskavail.msgalarmwarninginterval	24 hours

Table A.24 Server response alarm attributes

Server Response Attributes	Default Value
alarm.serverresponse.msgalarmstatinterval	600 seconds
alarm.serverresponse.msgalarmthreshold	10 seconds
alarm.serverresponse.msgalarmwarninginterval	24 hours

Alarm Attribute Examples

To modify the `msgalarmnoticercpt` attribute to send warning email to `joe@airius.com`:

```
configutil -o alarm.msgalarmnoticercpt -v joe@airius.com
```

To modify the `msgalarmstatinterval` attribute to monitor disk space every 600 seconds:

```
configutil -o alarm.diskavail.msgalarmstatinterval -v 600
```

To modify the `msgalarmthreshold` attribute to send a warning when the server response time is greater than 15 seconds:

```
configutil -o alarm.serverresponse.msgalarmthreshold -v 15
```

Appendix

B

sendmail Migration and Compatibility

Netscape Messaging Server includes a `sendmail` emulator program to maintain compatibility with mail programs that employ Unix `sendmail` to deliver their mail. This appendix explains how to move user mail accounts and mail messages from a Unix `sendmail` system to Messaging Server. It also describes the similarities and differences between the Unix `sendmail` application and the Messaging Server `sendmail` emulator.

This appendix includes the following sections:

- Moving Users to Messaging Server
- Moving sendmail Messages to Messaging Server
- Compatibility with Unix sendmail

Moving Users to Messaging Server

To migrate from a Unix `sendmail` system to Netscape Messaging Server, you need to perform the following procedures:

1. Use the `unix2ldif` utility to convert Unix user account information into an LDIF format file, for example, a file named `file.ldif`. For a detailed description of this step, see "Running the unix2ldif Utility" on page 442.

2. If the LDAP Directory already contains data, run the `ldifsplit` utility to split the LDIF file into two different LDIF format files:

   - A file that contains new entries for users that are *not* already in the LDAP Directory.
   - A file that contains the entries for those users who *are* already in the LDAP Directory.

   For a detailed description of this step, see "Running the ldifsplit Utility" on page 451.

3. Use the `chkuniq` utility to check for duplicate DNs, user IDs, and email addresses. For a detailed description of this step, see "Running the chkuniq Utility" on page 453.

4. Update the LDAP Directory with the user information that is now in the LDIF format files. For a detailed description of this step, see "Updating the LDAP Directory" on page 454.

5. Use the `chkuniq` utility to perform a final check for duplicate DNs, user IDs, and email addresses on the LDAP Directory.

6. Use the `MigrateUnixSpool` utility to move messages from users' `sendmail` spool files to the Messaging Server mailbox directories. See "Moving sendmail Messages to Messaging Server" on page 456.

Running the unix2ldif Utility

The `unix2ldif` utility writes your `sendmail` user account information to an LDIF (LDAP Data Interchange Format) file. The `unix2ldif` utility creates one LDAP entry in the LDIF file for each user account. If an account is skipped, the utility gives you a warning. The LDIF file contains the following information:

- User accounts
- Mail groups
- Forwarding aliases

By default, the `unix2ldif` utility gets its input from the following `/etc` directory files:

- `/etc/passwd`
- `/etc/shadow` (`/etc/security/shadow` on IBM AIX systems.)
- `/etc/aliases` /etc/passwd file.

By default, the `unix2ldif` utility operates on the three `/etc` files listed above. If those are the files that `unix2ldif` is to use, you do not need to explicitly specify the input files with the `-a`, `-p`, or `-s` options. If the input files are not in the `/etc` directory, or if they do not exist at all, or if they have different names, you can use the `-a`, `-p`, or `-s` options to specify the files you want `unix2ldif` to work on. For example, if you want to migrate your users in stages, you could create input files containing just a subset of users and then run `unix2ldif` on those files using the `-p`, `-s`, and `-a` options.

Note that `unix2ldif` can also read SunOS and Solaris NIS map files once they've been converted to ASCII format as described in "unix2ldif and NIS Maps" on page 450.

The `unix2ldif` utility is located in *`server-root`*`/bin/msg/admin/bin`

To run the utility, run the following command:

```
unix2ldif -b dn -d domain [options] > file.ldif
```

The `-b` *`dn`* and `-d` *`domain`* parameters are required. You can name the `unix2ldif` output file whatever you want (this document uses the name `file.ldif`).

The command-line options for `unix2ldif` are described in Table B.1.

Table B.1 unix2ldif options

Option	Description
`-a` *`file`*	For specifying an absolute path and name for the `aliases` input file. To specify no file, use `/dev/null`. Use this option if the aliases file: • does not exist • is not `/etc/aliases`
`-b` *`dn`*	**Required.** The base DN (distinguished name) to use in each DN constructed. To specify no DN, use empty double quotes (" "). For more information about the base DN, see your Directory Server documentation.
`-d` *`domain`*	**Required**. The address completion domain. This specifies the domain name that will be to the right of the `@` in each user's email address. For example, to specify that each user's email address is in the form: *`user`*`@airius.com`, you would use the option `-d airius.com`. You can use the `-d` option multiple times in a single command to produce multiple addresses for each user. The first address appears as the mail attribute; the other addresses appear as `mailAlternateAddress` attributes. For more information, see "Address and Host Completion Domains" on page 447.
`-D` *`domain`*	The host completion domain; `host` becomes `host.domain`. The default is the first value given for the `-d` option. Used to complete host names found in the `aliases` file. For more information, see "Address and Host Completion Domains" on page 447.
`-f`	Create forwarding-only entries in LDAP. Use for unresolved, single-target aliases from the `aliases` file.
`-F`	Create forwarding-only entries in LDAP. Use for all unresolved aliases in the `aliases` file.
`-g`	Create group entries in LDAP. Use for unresolved, multi-target aliases in the `aliases` file.
`-G`	Create group entries in LDAP. Use for all unresolved aliases in the `aliases` file.
`-h` *`host`*	The `mailHost` value to use for each user when no value can be derived from the `aliases` file. By default, `unix2ldif` relies on the `aliases` file for this information. For more information, see "Routing Aliases" on page 446.

Table B.1 unix2ldif options

Option	Description
`-H` *`host`*	An alternative to `-h`, `-H` forces the `mailHost` values for all users to be the value specified by *`host`*. This value cannot be overridden by the `aliases` file. For more information, see "Routing Aliases" on page 446.
`-m`	Turn off the automatic inclusion of `maildeliveryoption: mailbox` in the `ldif` entry.
`-n`	Turn off the automatic migration of `.forward` information for all users.

`-o`	Represent each forwarding-only entry in the directory as an organizational person. With this option, the object classes of the forwarding-only entries will be:
	
	`objectclass: top`
	`objectclass: person`
	`objectclass: organizationalPerson`
	`objectclass: inetOrgPerson`
	`objectclass: mailRecipient`
	`objectclass: nsMessagingServerUser`
	
	Without this option, forwarding-only entries will have only these object classes:
	
	`objectclass: top`
	`objectclass: mailRecipient`
	`objectclass: nsMessagingServerUser`
	
	If you represent the forwarding-only entry as an organizational person, the entry will show up in the Administration Server's User/Group UI under Users. Otherwise, the entry does not show up in the Administration Server's User/Group UI at all.

`-p` *`file`*	For specifying an absolute path and name for the `passwd` input file. Use this option if the password file is not `/etc/passwd`.

Table B.1 unix2ldif options

Option	Description
`-s file`	For specifying an absolute path and name for the `shadow` input file. Use this option if the shadow file: • does not exist • is not `/etc/shadow` (or `/etc/security/shadow` in AIX environments) To specify no file, use `/dev/null`.
`-v string`	An attribute of the form `attr: value` to include with each LDAP user entry constructed. To include individual user names (`uid`) in the string, use a percent sign (`%`) in the string where you want the user name to be. You can have up to four `%` signs for each `-v` option; however, you can have as many `-v` options as you need. This option is available for LDAP person entries only; it is not available for forwarding-only or group entries.
`-u`	Use the user ID (`uid`) for creating a DN for a user entry. The default is `cn`.
`-z`	Turn on debugging mode.

Routing Aliases

A routing alias for a user can be in the format `user: user@hostname` or the format `user@hostname.HostCompletionDomain`. For example, `esperanza: esperanza@sirius` or `esperanza@sirius.airius.com`. The user ID and the user portion of the routing alias must match exactly (case-sensitive match).

The `unix2ldif` utility determines how to set the `mailHost` value as follows:

- If the `-H` option is used to specify a host name, that host name is used to set the `mailHost` value. When the `-H` option is used, the `aliases` file is not used to determine how to set `mailHost`. (For more information on `aliases` files, see "unix2ldif and aliases Files" on page 448.)
- If the `-h` option is used to specify a host name, that host name is used to set the `mailHost` value for users who do not have a routing alias in the `aliases` file. Users that do have a routing alias in the `aliases` file have that alias used as their `mailHost` value.

- If neither the `-H` nor the `-h` options are used to specify a host name, `unix2ldif` obtains the `mailHost` value from the `aliases` file.

`HostCompletionDomain` is defined by the `-D` option as explained in "Using the -D Option to Set the Host Completion Domain" on page 448.

Address and Host Completion Domains

This section describes how to use the `-d` and `-D` address and host completion domain options.

Using the -d Option to Set the Address Completion Domain

The `-d` option is used to set the address completion domain(s). Address completion domains are used to generate the email addresses of a given user.

You can use `-d` more than once on a single command line to set several address completion domains. If you use more than one `-d` option, the first one is considered to be the primary domain. For example, to specify two address completion domains named `airius.com` and `other.com`, you enter:

```
unix2ldif -b dn -d airius.com -d other.com > file.ldif
```

For the user ID `lincoln`, this results in the following email addresses; `lincoln@airius.com` is the primary address:

```
mail:lincoln@airius.com
mailAlternateAddress: lincoln@other.com
```

Address completion domains are also used to generate more addresses for the user if the user has one or more nickname aliases in the `aliases` file. (For additional information on `aliases` files, see "unix2ldif and aliases Files" on page 448.)

Nicknames can be in the form *`nickname`*`: `*`user`*, where *`user`* is the user ID. Recursive nickname aliases are also recognized, for example, *`othername`*`: `*`nickname`*. Using the example address completion domains shown above, if `lincoln` has the nicknames `abe: lincoln` and `my_captain: abe` in the `aliases` file, the following email addresses are generated:

```
mail:lincoln@airius.com
mailAlternateAddress: lincoln@other.com
mailAlternateAddress: abe@airius.com
mailAlternateAddress: abe@other.com
mailAlternateAddress: my_captain@airius.com
mailAlternateAddress: my_captain@other.com
```

Using the -D Option to Set the Host Completion Domain

The `-D` option is used to set the host completion domain. If `-D` is not used, `unix2ldif` uses the primary address completion domain (the value specified with the `-d`) as the host completion domain.

The host completion domain is used as follows:

- When determining whether an `aliases` file entry is a routing alias of the form *`user@host.HostCompletionDomain`*, `unix2ldif` uses the host completion domain as defined here in the comparison. (For more information on routing aliases, see "Routing Aliases" on page 446, and "unix2ldif and aliases Files" on page 448.)

- When formulating a user's `mailHost` value, if the information is a host name with no dot because the user's routing alias was *`user@hostname`*, or the `-H` or `-h` options were used with a *`host`* value with no dot, then `unix2ldif` appends a dot and the host completion domain to get the fully-qualified host name for the `mailHost` value. For example, if `-h sirius` is used and the *`HostCompletionDomain`* is `airius.com`, the `mailHost` value becomes `sirius.airius.com`.

- When creating a Forwarding Alias LDAP entry or a Mail Group LDAP entry, the forwarding target or group member is specified as *`user@hostname`* (where *`hostname`* is a host name with no dot). The *`HostCompletionDomain`* is then added to complete the entry in the format: *`user@hostname.HostCompletionDomain`*.

Note that *`HostCompletionDomain`* is not applied to addresses found in a user's `.forward` file when creating a user LDAP entry and reading the `.forward` file to find forwarding addresses. Forwarding addresses read from a user's `.forward` file must be of the form *`something@domain`*, where *`domain`* is a DNS domain string containing at least one dot.

unix2ldif and aliases Files

When `unix2ldif` reads an `aliases` file, each entry is assumed to be in one of two forms:

```
key: value
key value
```

The *key*: *value* form is typical of /etc/aliases files, the *key value* form is typical of the output of ypcat -k aliases which dumps the NIS aliases map as explained in "unix2ldif and NIS Maps" on page 450. Any spaces between the colon and the value in the /etc form are ignored. No leading spaces are allowed before the *key*, but spaces are allowed within the *key*. Spaces are not allowed between the *key* and the colon (:) in the /etc form. Trailing white spaces after *value* are allowed but ignored.

The :include directive is not supported. If this directive is used in an aliases file, the results are undefined.

If there is more than one alias with the same key, this is considered to be invalid, and the results are undefined.

The *value* may contain one target. For example: name: miyoko. Or the *value* may contain multiple targets. For example: name: miyoko, hirokani, atemi. A comma is used to separate multiple targets. White space before or after a comma is allowed, but is ignored.

Each target must be a valid email address. For example: sarah, sarah@antares, sarah@antares.airius.com, sarah@airius.com.

Alias targets that contain certain unexpected characters such as: # & / % | (for example, /dev/null) are ignored. In such cases, the alias is still used if any usable targets are present.

The unix2ldif utility makes multiple passes through an aliases file. If an alias is selected in a given unix2ldif pass, it is not used in subsequent passes. The aliases are used as follows.

Pass 1. When creating user accounts for each user ID found in the passwd file, unix2ldif checks to see if an alias has a key equal to the user ID. If so, it considers it selected in this pass (and not available in subsequent passes). If the -H option was not used, and the alias fits the form of a routing alias (as discussed in "Routing Aliases" on page 446), it is used to determine the user's mailHost value, but otherwise, the alias is not used.

Pass 2. When creating user accounts for each user ID found in the passwd file, unix2ldif checks to see if there are any nickname aliases for the user ID. If so, they are used to generate addresses.

Pass 3. Remaining single-target aliases are used to create `Forwarding Alias` entries if the `-f` or `-F` options were specified. Or `Mail Group` entries if the `-G` option was specified. Remaining multi-target aliases are used to create Forwarding Alias entries if the `-F` option was specified, or Mail Group entries if the `-g` or `-G` options were specified.

Unexpected results may occur if there are conflicts between user IDs and aliases. For example, if `chou` and `wong` are both user IDs in the `passwd` file, and there is an alias `chou: wong`, unexpected results may be produced. (It is considered unexpected that aliases are being used to forward from one `passwd` file user account to another `passwd` file user account because this is ordinarily done with `.forward`).

At the end of its run, `unix2ldif` outputs those aliases that were not used.

unix2ldif and passwd Files

A `passwd` file entry is considered by `unix2ldif` as a valid `passwd` file user account if all of the following are true:

- None of the following fields are empty: user ID, password, gecos (full name), and home directory.
- The encrypted password string is 13 characters. The `unix2ldif` utility first checks the password field in the `passwd` file, and if the field is not 13 characters long, it checks the `shadow` file where the password string must be 13 characters long.
- The user has a mail host. That is, the user has a routing alias in the `aliases` file, or the `-h` or `-H` options were specified.

If the passwd entry is valid, an LDAP user entry is created for the user in the `unix2ldif` output file.

unix2ldif and NIS Maps

The `unix2ldif` utility can also read Network Information Service (NIS) map files.

If your NIS environment **does not include** `/etc/shadow` files (or `/etc/security/shadow` files in AIX environments), follow these steps to run `unix2ldif` on NIS maps:

1. Use `ypcat` to copy the NIS map data to ASCII files as follows:

```
ypcat passwd > /tmp/passwd.txt
ypcat -k aliases > /tmp/aliases.txt
```

2. Run `unix2ldif` on the two `/tmp/*.txt` files you just created.

```
unix2ldif -b dn -d domain -p /tmp/passwd.txt \
-s /dev/null -a /tmp/aliases.txt > file.ldif
```

If your NIS environment **does include** `/etc/shadow` files (or `/etc/security/shadow` files in AIX environments), follow these steps to run `unix2ldif` on NIS maps:

1. Use `ypcat` to copy the NIS map data to ASCII files as follows:

```
ypcat passwd > /tmp/passwd.txt
ypcat shadow > /tmp/shadow.txt
ypcat -k aliases > /tmp/aliases.txt
```

2. Run `unix2ldif` on the three `/tmp/*.txt` files you just created.

```
unix2ldif -b dn -d domain -p /tmp/passwd.txt \
-s /tmp/shadow.txt -a /tmp/aliases.txt > file.ldif
```

If your NIS environment does not use a `shadow` file, and therefore you do not have a `shadow.txt` file, use `-s /dev/null` instead of `-s /tmp/shadow.txt`.

Running the ldifsplit Utility

The `ldifsplit` utility takes the LDIF file created by the `unix2ldif` utility (*`file`*`.ldif`) and splits it into two files:

- A file of LDAP entries that are already in the directory (the DN already exists). You can name this file whatever you want (this document uses the name `mod.ldif` because it will be used to modify the existing LDAP Directory entries).

- A file of LDAP entries that are not already in the directory (no DN exists). You can name this file whatever you want (this document uses the name `add.ldif` because it will be used to add new entries to the LDAP directory).

If your LDAP Directory is empty, you do not need to run `ldifsplit`. But if your LDAP Directory already contains user account information, running `ldifsplit` at this time may prevent later problems.

The `ldifsplit` utility is located in *`server-root`*`/bin/msg/admin/bin`.

To run the `ldifsplit` utility, follow these steps:

1. Export the contents of your LDAP server instance into an LDIF file. You can name the export file whatever you want (this document uses the name `existing.ldif`). See your Directory Server documentation for details on how to export a directory server instance.

2. Run the `ldifsplit` utility using the following syntax (assuming that the `unix2ldif` output file is named `file.ldif` and the LDAP Directory export file is named `existing.ldif`):

   ```
   ldifsplit -f file.ldif -e existing.ldif -a add.ldif -m mod.ldif
   ```

   With this syntax, `ldifsplit` compares the `unix2ldif` output file (`file.ldif`) to the existing contents of the LDAP Directory as contained in the export file (`existing.ldif`). The utility then creates two new files:

   - `mod.ldif` containing entries for users who are already in the LDAP Directory.
   - `add.ldif` containing new entries that are not in the LDAP Directory

3. Check the `add.ldif` and `mod.ldif` files to see if they contain any entries.

   ```
   tail add.ldif
   tail mod.ldif
   ```

   If a file is empty, you do not need to do anything with it.

   If either the `add.ldif` file or the `mod.ldif` file contains data, that data needs to be written into the LDAP Directory as described in "Updating the LDAP Directory" on page 454.

The command-line options for `ldifsplit` are described in Table B.2.

Table B.2 ldifsplit options

Option	Description
`-a` *`add`*`.ldif`	Name of file containing new entries to be added.
`-e` *`existing`*`.ldif`	Name of the file containing the data that currently exists in the LDAP Directory server instance.
`-f` *`file`*`.ldif`	Name of the `unix2ldif` output file.
`-m` *`mod`*`.ldif`	Name of file containing LDAP entries that ned to be modified in the LDAP Directory.

Running the chkuniq Utility

The `chkuniq` utility checks the output files of the `unix2ldif` and `ldifsplit` utilities and the contents of the existing LDAP Directory for duplicate entries. It checks the specified files for

- Duplicate DNs
- Duplicate user IDs
- Duplicate email addresses

The `chkuniq` utility is located in *`server-root`*`/bin/msg/admin/bin.`

The steps that follow describe how to eliminate duplicate entries before writing data to the LDAP Directory.

Note: The steps below assume that you used `ldifsplit` to create the `add.ldif` file as described in "Running the ldifsplit Utility" on page 451. If you did not run `ldifsplit`, then run `chkuniq` on the `unix2ldif` output file (`file.ldif`) rather than `add.ldif`.

1. Use `chkuniq` to make sure that the `add.ldif` file (or `file.ldif`) is internally consistent with no duplicates.

```
chkuniq add.ldif
```

2. Use `chkuniq` to make sure that the `existing.ldif` file is internally consistent with no duplicates. (The `existing.ldif` file is created by exporting the current contents of the LDAP Directory to a file named `existing.ldif`.)

   ```
   chkuniq existing.ldif
   ```

3. Use `chkuniq` to compare the `add.ldif` file and `existing.ldif` files against each other.

   ```
   chkuniq add.ldif existing.ldif
   ```

When `chkuniq` finds a duplicate, it reports that to standard output. Or you can use `> filename` to redirect output to a file.

There is no output if chkuniq does not find any duplicates.

If `chkuniq` reports duplicates or errors, inspect your files and correct and resolve all problems before writing data to the LDAP Directory as described in "Updating the LDAP Directory" on page 454. (The most common cause of duplicates are errors in the `passwd` and `aliases` files that were given to `unix2ldif` as input.)

The command-line syntax options for the `chkuniq` utility are described in Table B.3.

Table B.3 chkuniq options

Syntax	Description
`chkuniq file1`	Check for duplicates within *file1*.
`chkuniq file1 file2`	Check for duplicates among *file1* and *file2*.

Updating the LDAP Directory

The final stage of moving users from a Unix `sendmail` system to Netscape Messaging Server is to update the LDAP Directory with the account information converted from `sendmail`.

- If you ran `ldifsplit` to create `add.ldif` and `mod.ldif` files, update the LDAP Directory from each of those two files.

- If you did not run `ldifsplit`, update the LDAP Directory from the `file.ldif` file created by `unix2ldif`.

There are two ways to update the LDAP Directory from the LDIF files:

- **Database Manager**. The Netscape Directory Server provides a Database Manager that can be used to update the LDAP Directory as described in "Updating the LDAP Directory with Database Manager" on page 455.
- **`ldapmodify`**. You can also use the `ldapmodify` command-line utility to update the LDAP Directory as described in "Updating the LDAP Directory with ldapmodify" on page 456.

Updating the LDAP Directory with Database Manager

You can use the Database Manager feature to write the user account information in the LDIF files to the LDAP Directory instance on the Directory Server.

To update the LDAP Directory instance with the Database Manager, follow these steps:

1. Make sure the Directory Server is running.
2. Log in to the Directory Server. You must have read and execute permission to use Database Manager and read and write permissions for the targeted entries in the LDAP directory. (You can run this remotely using `-h` *`host`*.)
3. Go to the Database Manager screen as described in your Directory Server documentation.
4. Select the Add Entries form.
5. In the Full Path to LDIF File field, enter the full path name to the LDIF file that contains the entries you want to add.

   If you created `add.ldif` and `mod.ldif` files by running `ldifsplit`, enter the full path and name of the `add.ldif` file. For example, if you ran `ldifsplit` in the `/tmp` directory: `/tmp/add.ldif`. Then repeat the process for `mod.ldif`.

   If you did not run `ldifsplit`, enter the full path and name of the `unix2ldif` output file. For example: `/tmp/file.ldif`.

6. Leave the Bind to Server field as is.
7. The Password field should already contain the correct password.
8. Choose Okay. The entries are added to your Directory Server manager.

Updating the LDAP Directory with ldapmodify

You can use the `ldapmodify` command-line utility to write the `sendmail` user account information in the LDIF files to the LDAP Directory instance on the Directory Server.

The `ldapmodify` utility is located in *`server-root`*`/shared/bin/`.

To update the LDAP Directory with `ldapmodify`, follow these steps:

1. Make sure the Directory Server is running.
2. Log in to the Directory Server system as `root`.
3. Go to the directory containing the LDIF files.
4. Run `ldapmodify` to update the LDAP Directory:

   If you created `add.ldif` and `mod.ldif` files by running `ldifsplit`, run `ldapmodify` on both files:

   ```
   ldapmodify -D "cn=Directory Manager" -w password -f mod.ldif
   ldapmodify -D "cn=Directory Manager" -w password -a -f add.ldif
   ```

   If you did not run `ldifsplit`, run `ldapmodify` on the `unix2ldif` output file. For example: `file.ldif`

   ```
   ldapmodify "cn=Directory Manager" -w password -a -f file.ldif
   ```

Moving sendmail Messages to Messaging Server

Before users can access their messages with Netscape Messaging Server 4.0, their messages must be moved from the `sendmail` spool file to their Messaging Server mailbox with the `MigrateUnixSpool` utility. This is done with the `MigrateUnixSpool` command-line utility.

Running the MigrateUnixSpool Utility

The `MigrateUnixSpool` utility reads messages from the user's `sendmail` mailbox and appends them to the user's Messaging Server mailbox.

Administrators can use the `MigrateUnixSpool` utility to move users' messages for them. Netscape recommends that administrators move messages for users rather than having users move their own messages. To do this, the administrator must be an authorized mail store administrator. When administrators run `MigrateUnixSpool` to move messages for others, they must use the `-a` and `-v` options.

Although users can run the `MigrateUnixSpool` utility to move their own messages from their `sendmail` spool file to their Messaging Server mailbox, Netscape recommends *against* having users move their own messages. If users run `MigrateUnixSpool` for themselves, they do not use the `-a` and `-v` options.

`MigrateUnixSpool` is located in *`server-root`*`/bin/msg/admin/bin`.

`MigrateUnixSpool` is run with the following syntax:

```
MigrateUnixSpool -u uid -o mailspool -d destmailhost
  -a admin -v password -m destmaildrop
```

The command-line syntax options for the `MigrateUnixSpool` utility are described in Table B.4.

Table B.4 MigrateUnixSpool options

Option	Description
`-a` *`admin`*	The user ID of the administrator running `MigrateUnixSpool` on *`destmailhost`*
`-d` *`destmailhost`*	The destination messaging server.
`-h`	Invokes help for this command.
`-m` *`destmaildrop`*	The name of the partition where the user's account is to be created.
`-o` *`mailspool`*	The mail spool path on the source messaging server.

Table B.4 MigrateUnixSpool options (Continued)

Option	Description
`-u` *`uid`*	The user ID of the user whose mailbox needs to be moved.
`-v` *`password`*	The password for the administrator running `MigrateUnixSpool` on *`destmailhost`*

MigrateUnixSpool Example

In this example, an administrator logged in as `admin` moves the spool file for someone with the user ID `lincoln` to the Messaging Server host `mailserver` in the `primary` partition.

```
MigrateUnixSpool -u lincoln -o /var/mail/lincoln -d mailserver
   -a admin -v password -m server-root/msg-instance/store/partition/
   primary
```

Compatibility with Unix sendmail

In order to maintain compatibility with Unix applications and processes that make use of the Unix `sendmail` command, Netscape Messaging Server 4.0 includes its own `sendmail` program that replaces the Unix `/usr/lib/sendmail` software.

Command-line Compatibility

The Messaging Server includes a program named `sendmail` that emulates the Unix `sendmail` program in the `/usr/lib` directory. Most of Unix `sendmail`'s functionality is performed by one or more modules in the Messaging Server, so the `sendmail` emulator is primarily for compatibility with many mail programs that employ Unix `sendmail` (rather than SMTP) to deliver their mail. The Messaging Server `sendmail` emulator program can also be used to start up the Messaging Server mail system and to check and deliver the mail queue.

Sending Mail with the sendmail Emulator

The Messaging Server `sendmail` emulator maintains compatibility with existing software that delivers mail using the Unix `sendmail` command.

Some examples of `sendmail` emulator commands that work for sending mail are:

```
/usr/lib/sendmail -t < /tmp/message
cat file1 | /usr/lib/sendmail -oem recip1,recip2
```

For a complete list of command-line options and options related to sending mail, see "sendmail Emulator Options and Aliases" on page 462.

Starting the Messaging Server with sendmail

Because Unix `sendmail` comes installed on most Unix-based machines, many scripts, such as system boot scripts, exist to start up `sendmail`. This is done with a command such as:

```
/usr/lib/sendmail -bd -q30m
```

The Messaging Server `sendmail` emulator recognizes this command and starts up the Messaging Server if it isn't already running. The `-q30m` option is ignored.

Checking the Mail Queue with mailq

Some system administrators are used to typing `mailq` to check for queued messages. The `sendmail` emulator provided with the Messaging Server will respond to this command with the contents of the mail queue. However, many server administrators now prefer to use the Messaging Server mail queue form, which makes processing the queue easier.

To check the mail queue with the `sendmail` emulator program, type `mailq` at a command prompt. If there are no queued messages, that fact will be reported.

If there are queued messages, each host that has queued messages waiting to be delivered will be listed, along with the number of pending deliveries.

Other Modes

The Unix `sendmail` program has several other operating modes that aren't necessary or are not supported by the Messaging Server. For a complete list of supported operating modes and command-line options and options, see "Functional Compatibility" on page 460.

Functional Compatibility

The following sections specify the differences between Unix `sendmail` and the Messaging Server `sendmail` emulator with regard to SMTP, aliases and mail forwarding, program delivery, file delivery, and mailing lists.

For information about how Netscape Messaging Server routes messages, see the chapter titled Message Routing.

Network Interface

The Messaging Server is preconfigured for interacting with other machines on the Internet.

In contrast, the Unix `sendmail` program needs to exchange mail with remote destinations using SMTP. Mail routing is achieved with rule sets containing address production rules written in a specialized programming language used to take addresses apart and put them back together in useful ways. Although this is a very powerful facility, it is error prone and requires extensive knowledge of Internet standards to set it up correctly.

Aliases and Mail Forwarding

Unix `sendmail` supports several types of aliases in the `/etc/aliases` file and in users' personal `.forward` files. The various types of aliases allow:

- Local users to receive mail at several addresses
- Messages to be distributed to multiple recipients
- Messages to be forwarded to users at other machines

With Messaging Server, each type of alias is created differently because of the structure of user accounts in the Directory Server database.

Delivery to Programs

Using Unix `sendmail`, you can create an alias or forward that delivers incoming mail to a program. The program then reads the mail and performs some operation depending on the mail contents. These types of programs usually filter messages into different mailboxes or send out automatic replies such as vacation notices. This functionality makes it easy to extend the mail system, but is problematic with respect to security.

You can use the Server Account Management window to set up Messaging Server program delivery for a particular user. However, because there are security issues specific to program delivery, program delivery is disabled by default. See the chapter titled "Program Delivery" for information on how to enable Messaging Server program delivery.

Delivery to Files

Unix `sendmail` makes it possible to set up an alias or `.forward` file to append mail to a file. This can be used to keep a record of incoming mail or to delete incoming mail by sending it to `/dev/null`. However, for delivery-to-file needs the author of `sendmail` recommends using an alternate delivery agent invoked through the delivery-to-program facility.

The Messaging Server's `sendmail` emulator will append undeliverable messages to users' `dead.letter` files, for users with Unix Delivery enabled. No general delivery-to-file facility is planned for the Messaging Server; appending mail to a file should be with the delivery-to-program facility as described in the appendix titled "Program Delivery."

Mailing Lists

Mailing lists in Unix `sendmail` are implemented using aliases and program deliveries. List recipients are stored either in the aliases database or in an external file using an `:include:` alias. Several mailing-list administration programs are available that can automate the task of maintaining recipient distribution lists, while `sendmail` handles the delivery of the messages.

With Messaging Server, you create groups with the Netscape Console as described in Chapter 4, "Managing Mail Users and Mailing Lists."

sendmail Emulator Options and Aliases

Table B.5 in this section lists the alias names you can use to run the Messaging Server `sendmail` emulator program. Table B.6 lists and describes the available command-line arguments that the `sendmail` emulator program recognizes. Certain options are recognized (through the `-o` command-line option), and their effects are described in Table B.7.

Alternate Names for sendmail

You can run the Unix `sendmail` program under several names as a shorthand way to specify the action to perform. The Messaging Server `sendmail` emulator program recognizes several of these alternate names. The behavior that results from invoking the `sendmail` emulator with an alternate name is summarized in Table B.5.

Table B.5 Invoking sendmail with alternate names

Name	What running under this name does
`bsmtp`	Prints an error message because batch SMTP is not supported.
`mailq`	Reports the contents of the mail queue.
`newaliases`	Prints an error message because the `aliases` file is not used.
`sendmail`	Sends a single mail message.

Note that the result described in Table B.6 will result if no other result is specified using a command-line option such as `-b` or `-I`.

Command-line options are processed using `getopt(3)` as in V8 `sendmail`. All the options supported by V8 `sendmail`, IDA sendmail, and other versions of `sendmail` are recognized; the extent of support for these options is given in Table B.6.

Table B.6 sendmail emulator program command-line options

Option	Description
`-B7`	If set to 7 bit, the high bit is stripped from every byte of the input message.
`-b`*x*	Changes the mode of operation. Where *x* is one of the following: The following modes are supported: `-be` Starts the Messaging Server mail system. `-bm` Sends a single mail message. `-bp` Shows the status of the mail queue. These modes are recognized but not supported: `-ba` Uses Arpanet protocols. `-bb` Does batch SMPT on standard input. `-bi` Initializes the aliases database. `-bs` Does SMTP on standard input. `-bt` Goes into address-testing mode. `-bz` Freezes the configuration.
`-C`	None. There is no configuration file, so this option is ignored.
`-c`	None. This option is obsolete.
`-d`	None. This option is ignored because there is no debug mode.
`-e`	Sets the error-reporting mode (see option `e` in Table B.7).
`-F`	Sets the full name of the sender. If the user running `sendmail` isn't `root`, `daemon`, `UUCP`, `SMTP`, `mail`, or `sendmail`, a header is added to the message indicating the actual sender.
`-f`	Sets the email address of the sender. The same precaution is taken as in the `-F` option.
`-h`	None. The hop count is determined by counting the number of received headers in the message.
`-I`	Runs as if invoked as `newaliases`, which just prints an error message.
`-i`	None. This is the default behavior. If `sendmail` is run interactively, a single "." (`.`) will end the message. If it is run non-interactively (for example, through a pipe to standard input), the end-of-file condition determines the end of the message.
`-M`	The entire queue is processed regardless of the specified Message ID.

Table B.6 sendmail emulator program command-line options (Continued)

Option	Description
`-m`	None. This is the default behavior. The sender is never removed from the list of recipients if it is listed as a recipient.
`-n`	None. This option is not supported.
`-o`	Sets an option. See Table B.7 for a list of supported options.
`-p`	None. This option is not supported.
`-q`	The deferred message queue is processed. If a time interval is given (for example, `sendmail -bd -q30m`), this option is ignored. When this option is specified as `-qR`, `-qS`, or `-qI` (as in V8 `sendmail`), then the behavior is the same as `-R`, `-S`, or `-M`, respectively.
`-R`	Attempts to process the queue for hosts matching the pattern provided (for example, `sendmail -Rabc` will start delivery of queued messages for all hosts containing the string `abc`).
`-r`	Same as `-f` option.
`-S`	The entire queue is processed regardless of the specified sender.
`-s`	None. This option is obsolete.
`-T`	None. This option is obsolete.
`-t`	Recipients are gathered from both the command line and the message header, and the message is delivered.
`-v`	Output is more verbose when sending mail.
`-x`	None. This is an illegal option that is recognized only to prevent printing an error message.
`-Z`	None. There is no frozen configuration file (or even a regular configuration file).

Options for sendmail

The Messaging Server `sendmail` emulator doesn't need a configuration file (`sendmail.cf`), yet most of Unix `sendmail`'s options can be set from the command line. Many of the options are meant for the `sendmail` daemon, but some of them are relevant to the normal operation of sending mail.

All the options supported by V8 `sendmail` are recognized, and the extent of the support for these options is shown in Table B.7. The options listed in Table B.7 refer only to the `sendmail` emulator, not to Messaging Server as a whole. Many of the options not supported by the `sendmail` emulator are supported by the Messaging Server in one way or another. Refer to the relevant sections of this guide to determine how to set parameters within the Messaging Server.

Table B.7 Options supported by V8 sendmail

Option	Description
`7`	If set, the high bit is stripped from every byte of the input message. Also see the `-B` command-line option.
`B`	This is always set to "." (period) and cannot be changed.
`d`	None. Because messages are always posted to the local SMTP server, the turn-around time is fairly quick, so the "i" or interactive mode is always used. However, support for other delivery modes may be added in the future.
`e` *`mode`*	Changes the error-reporting mode. Valid modes are `e`, `m`, `p`, `q`, and `w`. The behavior for each mode is the same as with Unix `sendmail`. However, if the local SMTP server is unavailable for some reason and mode `m` is chosen, the error message will not be deliverable either. In this case, the message is saved in the sender's `~/dead.letter` file.
`f`	None. When a "From:" line is received, it is changed to "X-Unix-From:" so that it will be RFC822 compliant.
`i`	None. See the `-i` command-line option for details.
`o`	None. This is the default behavior and cannot be disabled.
`v`	Turns on verbose output. Also see the `-v` command-line option.
Others	No other options have any effect. All other options, even invalid ones, are ignored.

Appendix

C

SNMP MIB

The Netscape Management Information Base (MIB) for Messaging Server stores the server information that administrators manage through the Simple Network Management Protocol (SNMP).

This appendix describes the Messaging Server SNMP MIB and provides its complete text. For information about SNMP in Netscape Messaging Server, see "Using SNMP on Unix Platforms" on page 299.

This appendix contains the following sections:

- About the Messaging Server MIB
- How the MIB Is Activated
- Format of MIB Entries
- Description of the MIB File
- The Messaging Server MIB

About the Messaging Server MIB

The Messaging Server Management Information Base (MIB) is a private SNMP MIB module designed for Netscape Messaging Server.

Using SNMP with network management software, such as HP OpenView, server administrators can manage Netscape Messaging Server as a network element, performing remote monitoring and data exchange between servers. The SNMP MIB allows the server to monitor server statistics, query static variables, and notify the management station of Messaging Server events.

Each installation of SNMP has its own MIB, a database or information store with a tree-like hierarchy that contains definitions of managed objects, or variables, that store network information for the server. The most general information about the network is at the top level of the hierarchy. Each level below contains information that describes specific parts of the network. For detailed information about MIB structure, see RFC 1155.

Messaging Server's private Messaging Server MIB file, named `netscape-mail.mib`, contains the definitions for variables that store network information for Messaging Server. The MIB is installed during Messaging Server installation in the *`ServerRoot`*`/plugins/snmp` directory. It is a subtree under the object identifier (OID) `enterprises 1450`, which is the Netscape OID.

You can view the information in the Messaging Server MIB through two SNMP tables, using a program such as HP OpenView. The tables display both SNMP static variables and MTA variables. For information about these variables, see "MIB Variables" on page 472.

For detailed information about the MIB in SNMP, version 1, see RFC 1212 and RFC 1215.

Note: The Messaging Server MIB supports a subset of the objects defined by the MADMAN MIB, as described in RFC 1566. For further information about this subset, contact the Netscape Server Group.

How the MIB Is Activated

To activate the MIB, you must first install the SNMP master agent for the server and then enable the SNMP subagent, which calls the MIB.

The master agent exchanges information between the network management station (NMS) and its subagents. For more information about the master agent, see the documentation for the Netscape Administration Server.

Once you have installed the master agent, you enable the SNMP subagent. The master agent asks the subagent for information, and the subagent queries the variables defined in the MIB. For more information about the subagent, see "The Messaging Server Subagent" on page 301.

Format of MIB Entries

The MIB file contains the definitions for managed objects, or variables, that store network information for the server. Each variable definition includes the variable name, its data type and read/write access level, a brief description, and a permanent object identifier. All MIB objects are defined using Abstract Syntax Notation One (ASN.1).

This sample entry shows the definition for the `nsmailEntityDescr` variable:

```
nsmailEntityDescr      OBJECT-TYPE     / object type
           SYNTAX      DisplayString (SIZE (0..255))      / syntax
           ACCESS      read-only      / read/write access level
           STATUS      mandatory      / status
           DESCRIPTION                / description
           "A general textual description of the Netscape Mail Server."
           ::= { nsmailEntityInfo 1 }  / object identifier
```

This definition contains the following information:

- The variable name, in this case `nsmailEntityDescr`, which is of `OBJECT-TYPE`.
- **Syntax** gives the abstract data type of the variable object type in ASN.1 notation. For example, the Syntax of the `nsmailEntityDescr` variable is `DisplayString (SIZE (0..255))`. See "Syntax Types" on page 470.
- **Access** gives the read/write access level to the variable. Possible access levels are read-only, read-write, write-only, or not-accessible. The access level to all Messaging Server MIB elements is either `read-only`, like that of the `nsmailEntityDescr` variable, or `not-accessible`.
- **Status** tells whether the element is mandatory, optional, or obsolete. The Status for all Messaging Server MIB elements is mandatory, or required for the server.

- **Description** is a text description of the element, enclosed in quotes. For example, the description of the `nsmailEntityDescr` variable is "A general textual description of the Netscape Mail Server."
- **Object Identifier** is an assigned name that serves as a permanent identifier for each managed object in the MIB name tree in its namespace. Objects in SNMP are hierarchical; the object identifier is a sequence of labels that represents the object in the hierarchy. For example, `nsmailEntityDescr` is identified as `nsMailEntityInfo 1`. This means that it has the label 1 in the subtree `nsMailEntityInfo`. `nsMailEntityInfo`, in turn, has the label 1 in the `nsmail` subtree.

Syntax Types

The Syntax line of the MIB entry gives the abstract data type of the variable, which must resolve to an instance of an ASN.1 syntax type. These types are defined in various Network Working Group Requests for Comments (RFCs); their definitions are imported into the MIB file, as described in "MIB Imports List" on page 471.

Table C.1 lists the data types used in the Messaging Server MIB, with descriptions, import sources, and examples from the MIB.

Table C.1 Data Types of Messaging Server MIB Variables

Data type	Description and import	Example
`Counter32`	32-bit unsigned long; increases to a maximum value, then restarts from zero. Imported from SNMPv2-SMI.	`mtaReceivedMessages`
`DisplayString`	Human-readable ASCII string with length of 0 to 255 characters. Imported from SNMPv2-TC.	`nsmailEntityDescr`
`Gauge32`	32-bit unsigned long; can increase or decrease between set values. Imported from SNMPv2-SMI.	`mtaStoredMessages`
`INTEGER`	Integer.	`mtaId`

Table C.1 Data Types of Messaging Server MIB Variables (Continued)

Data type	Description and import	Example
`MtaEntry`	Table format for MTA data retrieval; defines an entry in the `MTAtable`. Defined in the Messaging Server MIB.	`mtaEntry`
`SEQUENCE OF MtaEntry`	Constructor type that generates a table using the specified list constructor; here, it generates the MTA table using the `MtaEntry` type. Defined in ASN.1.	`mtaTable`

Description of the MIB File

The MIB file is organized into several main parts:

- MIB Imports List. The MIB imports list tells the sources of the object and data type definitions used in the MIB.
- Module Definition. The module defines the current subtree. It contains organization information and a description of the MIB.
- MIB Variables. MIB variables store the basic information required for network management.
- MIB Traps. A MIB trap is a message sent by the server to alert the NMS about a server event.

MIB Imports List

The Imports list at the beginning of the MIB file gives the sources of the object and data type definitions used in the MIB. For example, the `Counter32` and `Gauge32` data type definitions are imported from the SNMPv2-SMI module. For the text of the import section, see the Imports list in "The Messaging Server MIB" on page 476.

Module Definition

The module defines the current subtree that is being described by the MIB file. It contains organization information and a description of the MIB. The Messaging Server MIB names the Netscape Communications Corporation as its organization, gives the company name and address as its contact information, and provides a text description of the MIB. The Messaging Server OID is defined as `netscape 5`. For the text of the module definition, see "The Messaging Server MIB" on page 476.

MIB Variables

The MIB defines a number of managed objects, or variables, which have names and values. Variables store information required for network management. You can view the data stored in MIB variables using a program such as HP OpenView.

Messaging Server MIB variables are organized into data for two tables.

- The static variables table provides basic information about the Messaging Server installation. See "Static Variables" on page 472.
- The MTA Table and MTA variables store information specific to the MTA (Message Transfer Agent). See "MTA Table for Server Statistics" on page 473.

Static Variables

The static variables table provides basic information about the Messaging Server installation, such as the version number, the physical location of the server, and contact information. Static variable values are set during server installation and cannot be updated while the server is running.

Table C.2 lists static variables alphabetically by name, with descriptions and examples.

Table C.2 Quick Reference to MIB Attributes: Static Variables

Attribute	Description	Examples
nsmailEntityContact	Contact person for Messaging Server at this installation, usually a server administrator.	John Smith (jsmith@acme.com)
nsmailEntityDescr	Text description of Messaging Server.	Netscape Messaging Server
nsmailEntityLocation	Physical location of Messaging Server, usually a street address.	999 Worldend Rd.
nsmailEntityName	Name assigned to Messaging Server at this installation; should match the link on the Administration Server selection page for Messaging Server.	MTA-40
nsmailEntityOrg	Organization using Messaging Server, usually a department or company name.	Acme Corp.
nsmailEntityVers	Version of Messaging Server that is installed.	4.1

MTA Table for Server Statistics

The MTA Table and MTA variables store information specific to the MTA (Message Transfer Agent), such as the number and volume of messages sent, received, or stored.

The MTA is a program for message routing and delivery. Several MTAs can cooperate in getting messages for the intended recipient. Messages are either delivered to the local message store by the MTA or routed to another MTA for remote delivery.

Table C.3 lists MTA MIB variables alphabetically by name, with descriptions.

Table C.3 Quick Reference to MIB Attributes: MTA Table

Attribute	Description
`mtaId`	Identifier of the MTA for this installation. Since only one Netscape Messaging Server can be installed, this attribute can have only one value.
`mtaReceivedMessages`	Total number of messages received since MTA initialization.
`mtaReceivedRecipients`	Number of recipients for all messages received since MTA initialization.
`mtaReceivedVolume`	Total volume of messages received since MTA initialization, in kilo-octets.
`mtaStoredMessages`	Total number of messages stored in the MTA.
`mtaStoredRecipients`	Number of recipients for all messages stored in the MTA.
`mtaStoredVolume`	Total volume of messages stored in the MTA, in kilo-octets.
`mtaTransmittedMessages`	Total number of messages transmitted since MTA initialization.
`mtaTransmittedRecipients`	Number of recipients specified in all messages transmitted since MTA initialization.
`mtaTransmittedVolume`	Total volume of messages transmitted since MTA initialization, in kilo-octets.

MIB Traps

A MIB trap is a message from the server that notifies the NMS of server events. Messaging Server MIB traps notify the NMS when the server may be down, does not respond to polling, or has restarted.

The SNMP trap is an example of managed device-initiated communication. For more information, see “Communication Between the NMS and the Managed Device” on page 300.

Each trap sends additional MIB information about the server when it reports an event. The following example shows the information sent with the `nsMailServerDown` trap: Netscape Messaging Server's general description, version number, location, and contact information.

```
nsmailEntityDescr, nsmailEntityVers,
nsmailEntityLocation, nsmailEntityContact
```

Note: To make sure that traps are sent to the NMS, you must set the correct community and trap destination information through the Administration Server. See the documentation for the Administration Server.

Table C.4 lists traps alphabetically by name, with descriptions.

Table C.4 Quick Reference to MIB Attributes: Traps

Trap	Description
`nsMailServerDown`	Messaging Server may be down. Sends general description, version number, location, and contact information for the server at the time of the event.
`nsMailServerNoResponse`	Messaging Server does not respond to its polls, but may still be up and working. Sends general description, version number, location, and contact information for the server at the time of the event.
`nsMailServerStart`	Messaging Server starts or restarts. Sends general description, version number, and location of the server at the time of the event.

The Messaging Server MIB

For your convenience, this section includes the text of the Messaging Server MIB.

```
SNMP MIB


Netscape Messaging Server 4.1


This file contains Netscape Messaging Server's Simple Network
Management Protocol (SNMP) Management Information Base (MIB).


NSMAIL-MIB DEFINITIONS ::= BEGIN


IMPORTS
           MODULE-IDENTITY, OBJECT-TYPE, OBJECT-IDENTITY
                     FROM SNMPv2-SMI
           enterprises
                     FROM ObjectIds
           Counter32, Gauge32
                     FROM SNMPv2-SMI
           Counter, IpAddress, TimeTicks
                     FROM RFC1155-SMI
           DisplayString, TimeInterval
                     FROM SNMPv2-TC
           TRAP-TYPE
                     FROM RFC-1215;


           netscape OBJECT IDENTIFIER ::= { enterprises 1450 }


           nsmail MODULE-IDENTITY
           LAST-UPDATED         "9706021700Z"
```

```
            ORGANIZATION         "Netscape Communications Corp."
            CONTACT-INFO         "Netscape Communications Corp.
                                  501 E. Middlefield Rd.
                                  Mountain View, CA 94043"

            DESCRIPTION          "A private MIB module for Netscape
                                  Messaging Server"

            ::= { netscape 5 }

-----
-- Static variables
-----
            nsmailEntityInfo OBJECT IDENTIFIER ::= { nsmail 1 }

            nsmailEntityDescr     OBJECT-TYPE
                SYNTAX     DisplayString (SIZE (0..255))
                ACCESS     read-only
                STATUS     mandatory
                DESCRIPTION   "A general textual description
                               of the Netscape Mail Server."
                ::= { nsmailEntityInfo 1 }

            nsmailEntityVers     OBJECT-TYPE
                SYNTAX     DisplayString (SIZE (0..255))
                ACCESS     read-only
                STATUS     mandatory
                DESCRIPTION  “The Version of the Netscape Mail Server."
                ::= { nsmailEntityInfo 2 }

            nsmailEntityOrg     OBJECT-TYPE
                SYNTAX     DisplayString (SIZE (0..255))
```

```
        ACCESS     read-only
        STATUS     mandatory
        DESCRIPTION  "Organization responsible for Netscape
                      Mail Server at this installation."
        ::= { nsmailEntityInfo 3 }

    nsmailEntityLocation     OBJECT-TYPE
       SYNTAX     DisplayString (SIZE (0..255))
       ACCESS     read-only
       STATUS     mandatory
       DESCRIPTION  "Physical location of this entity
                     (Netscape Mail Server). For example:
                     hostname, building number,
                     lab number, etc."
        ::= { nsmailEntityInfo 4 }

    nsmailEntityContact     OBJECT-TYPE
        SYNTAX      DisplayString (SIZE (0..255))
        ACCESS      read-only
        STATUS      mandatory
        DESCRIPTION "Contact person(s) responsible for the
                     Netscape Mail Server at this
                     installation, together with
                     information on how to contact."
        ::= { nsmailEntityInfo 5 }

     nsmailEntityName     OBJECT-TYPE
        SYNTAX     DisplayString (SIZE (0..255))
        ACCESS     read-only
        STATUS     mandatory
        DESCRIPTION  "Name assigned to this entity at the
                      installation site."
```

```
                ::= { nsmailEntityInfo 6 }

-----
-- mta table for statistic information
-----
           mtaTable       OBJECT-TYPE
               SYNTAX     SEQUENCE OF MtaEntry
               ACCESS     not-accessible
               STATUS     mandatory
               ::= { nsmail 2 }

           mtaEntry       OBJECT-TYPE
               SYNTAX     MtaEntry
               ACCESS     not-accessible
               STATUS     mandatory
               INDEX     { mtaId }
               ::= { mtaTable 1 }

           MtaEntry     ::= SEQUENCE {
               mtaReceivedMessages         Counter32,
               mtaStoredMessages           Gauge32,
               mtaTransmittedMessages      Counter32,
               mtaReceivedVolume           Counter32,
               mtaStoredVolume             Gauge32,
               mtaTransmittedVolume        Counter32,
               mtaReceivedRecipients       Counter32,
               mtaStoredRecipients         Gauge32,
               mtaTransmittedRecipients    Counter32,
               mtaId                       INTEGER
               }

           mtaReceivedMessages     OBJECT-TYPE
```

```
    SYNTAX     Counter32
    ACCESS     read-only
    STATUS     mandatory
    DESCRIPTION  "The total number of messages received
                  since MTA Initialization."
    ::= { mtaEntry 1 }

mtaStoredMessages     OBJECT-TYPE
    SYNTAX     gauge32
    ACCESS     read-only
    STATUS     mandatory
    DESCRIPTION  "The total number of messages currently
                  stored in the MTA."
    ::= { mtaEntry 2 }

mtaTransmittedMessages     OBJECT-TYPE
    SYNTAX     Counter32
    ACCESS     read-only
    STATUS     mandatory
    DESCRIPTION  "The total number of messages
                  transmitted since MTA Initialization."
    ::= { mtaEntry 3 }

mtaReceivedVolume     OBJECT-TYPE
    SYNTAX     Counter32
    ACCESS     read-only
    STATUS     mandatory
    DESCRIPTION  "The number of msgs (in kilo-octets
                  received since MTA Initialization."
    ::= { mtaEntry 4 }

mtaStoredVolume     OBJECT-TYPE
```

```
        SYNTAX     Gauge32
        ACCESS     read-only
        STATUS     mandatory
        DESCRIPTION  "The total number of msgs
                     (in kilo-octets) currently stored
                     in the MTA."
        ::= { mtaEntry 5 }

    mtaTransmittedVolume     OBJECT-TYPE
        SYNTAX     Counter32
        ACCESS     read-only
        STATUS     mandatory
        DESCRIPTION  "Number of msgs, in kilo-octets,
                     transmitted since MTA initialization."
        ::= { mtaEntry 6 }

    mtaReceivedRecipients     OBJECT-TYPE
        SYNTAX     Counter32
        ACCESS     read-only
        STATUS     mandatory
        DESCRIPTION  "The number of recipients specified
                     in all messages received since MTA
                     Initialization. Recipients this MTA
                     had no responsibility for are not counted."
        ::= { mtaEntry 7 }

    mtaStoredRecipients     OBJECT-TYPE
        SYNTAX     Gauge32
        ACCESS     read-only
        STATUS     mandatory
        DESCRIPTION  "The total number of recipients
                     specified in all messages currently
```

```
                        stored in the MTA. Recipients this MTA
                        had no responsibility for are not counted."
    ::= { mtaEntry 8 }

mtaTransmittedRecipients     OBJECT-TYPE
    SYNTAX     Counter32
    ACCESS     read-only
    STATUS     mandatory
    DESCRIPTION  "The number of recipients specified in
                  all messages transmitted since MTA
                  Initialization. Recipients this MTA
                  had no responsibility for are not counted.”
    ::= { mtaEntry 9 }

mtaId          OBJECT-TYPE
    SYNTAX     INTEGER
    ACCESS     read-only
    STATUS     mandatory
    DESCRIPTION  The id of the MTA as configured."
    ::= { mtaEntry 10 }

-----
-- Traps
-----

nsMailServerDown     TRAP-TYPE
    ENTERPRISE       netscape
    VARIABLES       { nsmailEntityDescr, nsmailEntityVers,
                      nsmailEntityLocation,
                      nsmailEntityContact }
    DESCRIPTION       This trap is generated whenever the
                      agent detects the Netscape Mail
                      Server to be (potentially) Down."
```

```
        ::= 5001

    nsMailServerStart    TRAP-TYPE
        ENTERPRISE       netscape
        VARIABLES      { nsmailEntityDescr, nsmailEntityVers,
                         nsmailEntityLocation }
        DESCRIPTION     "This trap is generated whenever
                         the agent detects the Netscape
                         Mail Server to have (re)started."
        ::= 5002

    nsMailServerNoResponse  TRAP-TYPE
        ENTERPRISE       netscape
        VARIABLES      { nsmailEntityDescr, nsmailEntityVers,
                         nsmailEntityLocation,
                         nsmailEntityContact }
        DESCRIPTION     "This trap is generated whenever the
                         agent detects the Netscape Mail Server
                         not responding to its polls. This
                         TRAP is different from the
                         'nsMailServerDown' TRAP, as the
                         Netscape Mail Server is still
                         potentially up, serving its main
                         purpose. But, the SNMP data collection
                         entity has most likely gone down."
        ::= 5003

END
```

Glossary

A record — A type of record stored in a DNS server and containing a host name and its associated IP address. A records are used by messaging servers on the Internet to route email. See also **Domain Name System (DNS)** and **MX record**.

access control — A method for controlling access to a server or to folders and files on a server.

access domain — Limits access to certain Messaging Server operations from within a specified domain. For example, an access domain can be used to limit where mail for an account can be collected.

account — Information that defines a specific user or user group. This information includes the user or group name, valid email address or addresses, and how and where email is delivered.

action — A component of a UBE filter; it specifies the action that is to be performed if a match occurs.

address — Information in an email message that determines where and how the message must be sent. Addresses are found both on message headers and on message envelopes.

address handling — The actions performed by the MTA to detect errors in addressing, to rewrite addresses if necessary, and to match addresses to recipients.

addressing protocol — The addressing rules that make email possible. SMTP is the most widely used protocol on the Internet and the protocol supported by the Netscape Messaging Server. Other protocols include X.400 and UUCP (Unix to Unix Copy Protocol).

administrator — A user with administrative privileges for a server or multiple servers. See also **Messaging Server administrator**.

Allow filter — A Messaging Server access-control rule that identifies clients that are to be allowed access to a service such as IMAP, POP, or SMTP. Compare **Deny filter.**

alternate address — A secondary address for an account, generally a variation on the primary address. In some cases it is convenient to have more than one address for a single account.

anonymous access An optional type of access to a server, in which the user named `anonymous` is granted access without need for a password.

attribute An item of information contained in a tag, such as an HTML tag or a Messaging Multiplexor command tag.

AUTH An SMTP command enabling an SMTP client to specify an authentication method to the server, perform an authentication protocol exchange, and, if necessary, negotiate a security layer for subsequent protocol interactions.

authentication 1) The process of proving the identity of a client user to the Netscape Messaging Server. 2)The process of proving the identity of the Netscape Message Server to a client or another server.

authentication certificate A digital file sent from server to client or client to server to verify and authenticate the other party. The certificate ensures the authenticity of its holder (the client or server). Certificates are not transferable.

AutoReply utility A utility that automatically responds to messages sent to accounts with the AutoReply feature activated. Every account in the Netscape Messaging Server can be configured to automatically reply to incoming messages.

backside port The port that the Messaging Multiplexor uses to communicate with the servers that contain the Multiplexor's clients' mailboxes. Compare **listen port.**

banner A text string displayed by a service such as IMAP when a client first connects to it.

base DN A distinguished name entry in the directory from which searches will occur. Also known as a search base. For example, ou=people, o=airius.com.

bind DN A distinguished name used to authenticate to the Directory Server when performing an operation.

body The main part of an email message. Although headers and envelopes must follow a standard format, the body of the message has a content determined by the sender—the body can contain text, graphics, or even multimedia.

capability A string, provided to clients, that defines the functionality available in a given IMAP service.

case tag A tag in the Message Field component of a UBE filter that specifies that matching should be case-sensitive.

certificate-based authentication Identification of a user from a digital certificate submitted by the client. Compare **password authentication.**

certificate database A file that contains a server's digital certificate(s). Also called a *cert file*.

certificate name The name that identifies a certificate and its owner.

cipher An algorithm used in encryption.

client A software entity that requests services or information from a server. Compare **user.**

CNAME record A type of record stored in a DNS server. A CNAME record maps a domain name alias to a domain name.

comment character A character that, when placed at the beginning of a line, turns the line into a nonexecutable comment. For Netscape configuration files, it is the pound sign (#).

config_util A command line utility for making changes to configuration information stored in the directory server or in the local configuration file.

counter_util A command line utility for displaying all counters in a counter object.

daemon A Unix program that runs in the background, independent of a terminal, and performs a function whenever necessary. Common examples of daemon programs are mail handlers, license servers, and print daemons. On Windows NT machines, this type of program is called a service. See also **service**.

default log A Messaging Server log file that is produced by a service or utility other than the principal services Administration, SMTP, IMAP, and POP.

Deny filter A Messaging Server access-control rule that identifies clients that are to be denied access to a service such as IMAP, POP, or SMTP. Compare **Allow filter.**

deliver A command line utility for delivering mail to POP or IMAP folders.

directory lookup The process of searching the directory for information on a given user or resource, based on that user or resource's name or other characteristic.

directory service An application designed to manage information about people and resources within an organization. See also **Lightweight Directory Access Protocol**.

DNS See **Domain Name System**.

DNS alias A host name that the DNS server recognizes as pointing to a different host—specifically a DNS CNAME record. Machines always have one real name, but they can have one or more aliases. For example, `www.airius.domain` might be an alias that points to a real machine called `realthing.airius.domain` where the server currently exists.

DNS spoofing A form of network attack in which a client attempting to access or send a message to a server misrepresents its host name.

domain name A unique name that defines an administrative organization. Domains can contain other domains. Domain names are interpreted from right to left. For example, `airius.com` is both the domain name of the Airius Company and a subdomain of the top-level `com` domain. The `airius.com` domain can be further divided into subdomains such as `corp.airius.com`, and so on. See also **host name** and **fully-qualified domain name**.

Domain Name System (DNS) The system used by machines on a network to associate standard IP addresses (such as 198.93.93.10) with host names (such as `www.airius.com`). Machines normally get this information from a DNS server. See also **A record** and **MX record**.

domain part The part of an email address that identifies the administrative authority responsible for the recipient.

encryption The process of disguising information so that it cannot be deciphered (decrypted) by anyone but the intended recipient.

enterprise network A network that consists of collections of networks connected to each other over a geographically dispersed area. The enterprise network serves the needs of a widely distributed company and is used by the company's mission-critical applications.

entry (1) In a directory, the collected information about a single person to resource. (2) In a log file, a line that represents one logged event.

envelope A container for information about the sender and the recipient of an email message. This information is not part of the message header. Envelopes are used by various email programs as messages are moved from place to place. Users see only the header and body of a message.

envelope field A named item of information, such as `RCPT TO`, in a message envelope.

envonly tag A tag in the Message Field component of a UBE filter that restricts the filter processing to the message envelope alone.

error handler A program that handles errors. In the Messaging Server, issues error messages and processes error action forms after the postmaster fills them out.

Error-Handler Action form A form sent to the postmaster account that accompanies a received message that the Messaging Server cannot handle. The postmaster fills out the form to instruct the server how to process the message.

error message A message reporting an error or other situation. Netscape Messaging Server generates messages in a number of situations, notably when it gets an email message that it can't handle. Others messages, called notification errors, are for informational purposes only.

ETRN An SMTP command enabling a client to request that the server start the processing of its mail queues for messages that are waiting at the server for the client machine. Defined in RFC 1985.

EXPN An SMTP command for expanding a mailing list. Defined in RFC 821.

extension library A shared library used to extend or override the capabilities of a plugin such as the UBE plugin.

extranet The part of a company intranet that customers and suppliers can access. See also **intranet**.

facility In a Messaging Server log-file entry, a designation of the software subsystem (such as Network or Account) that generated the log entry.

filter See **UBE filter, Allow filter, Deny filter**.

filter.cfg The file that holds all the UBE filter rules; used by the UBE plugin.

filter.opt A file that controls certain aspects of the behavior of the UBE plugin.

firewall A network configuration, usually both hardware and software, that forms a barrier between networked computers within an organization and those outside the organization. A firewall is commonly used to protect information such as a network's email, discussion groups, and data files within a physical building or organization site.

folder A named collection of messages. Folders can contain other folders. See also **personal folder** and **shared folder**.

forwarding The act that occurs when an MTA sends a message delivered to a particular account to one or more new destinations as specified by the account's attributes. Forwarding may be configurable by the user. See also **routing**.

FQDN See **fully-qualified domain name**.

fully-qualified domain name (FQDN) The unique name that identifies a specific Internet location. See also **domain name**.

greeting form A message usually sent to users when an account is created for them. This form acts as confirmation of the new account and verification of its contents. The greeting form also instructs users on how to change information related to their mail account. During installation, system administrators have the option of deciding whether to send greeting forms to users.

hashdir A command line utility for determining which directory contains the message store for a particular user.

header The portion of an email message that precedes the body of the message. Headers contain information useful to email programs and to users trying to make sense of the message: they tell whom the message is for, who sent it, when it was sent, and what it is about. Headers must be written according to the SMTP protocol so that email programs can read them.

header field A named item of information, such as `From:` or `To:`, in a message header.

hop A transmission between two computers.

host The machine on which one or more servers reside.

host name The name of a particular machine within a domain. The fully qualified host name consists of two parts: the host name and the domain name. For example, `mail.airius.com` is the machine `mail` in the domain `airius.com`. Host names must be unique within their domains. Your organization can have multiple machines named `mail`, as long as the machines reside in different subdomains; for example, `mail.corp.airius.com` and `mail.field.airius.com`. Host names always map to a specific IP address. See also **domain name**, **fully-qualified domain name**, and **IP address**.

host name hiding The practice of having domain-based email addresses that don't contain the name of a particular host.

HTTP See **HyperText Transfer Protocol**.

hub A host that acts as the single point of contact for the system. When two networks are separated by a firewall, for example, the firewall computer often acts as a mail hub.

HyperText Transfer Protocol A standard protocol that allows the transfer of hypertext documents over the Web. Netscape Messaging Server provides an HTTP service to support web-based email. See **Messenger Express**.

IMAP4 See **Internet Message Access Protocol Version 4**.

ImapMMP.config A configuration file that defines an IMAP4 instance of the Messaging Multiplexor.

ImapMMP.sh A Unix shell script that sets configuration parameters and executes the IMAP Messaging Multiplexor.

ImapProxy The executable file for the IMAP Messaging Multiplexor.

imscripter A command line utility that talks to an IMAP server. You can use this utility to execute a command or batch of commands on IMAP folders.

INBOX The name reserved for a user's default mailbox for mail delivery. INBOX is the only folder name that is case-insensitive. For example: INBOX, Inbox, and inbox are all valid names for a users default mailbox.

installation directory The directory into which the binary (executable) files of a server are installed. For the Messaging Server, it is a subdirectory of the server root: *`serverRoot`*`/bin/msg/`. Compare **instance directory**, **server root**.

instance A separately executable configuration of a server or other software entity on a given host. With a single installed set of binary files, it is possible to create multiple instances of Netscape servers that can be run and accessed independently of each other.

instance directory The directory that contains the files that define a specific instance of a server. For the Messaging Server, it is a subdirectory of the server root: *`serverRoot`*`/msg-`*`instanceName`*`/`, where *`instanceName`* is the name of the server as specified at installation. Compare **installation directory**, **server root**.

Internet The name given to the worldwide network of networks that uses TCP/IP protocols.

Internet Message Access Protocol Version 4 (IMAP4) A standard protocol that allows users to be disconnected from the main messaging system and still be able to process their mail. The IMAP specification allows for administrative control for these disconnected users and for the resynchronization of the users' message store once they reconnect to the messaging system.

Internet Protocol (IP) The basic network-layer protocol on which the Internet and intranets are based.

intranet A network of TCP/IP networks within a company or organization. Intranets enable companies to employ the same types of servers and client software used for the World Wide Web for internal applications distributed over the corporate LAN. Sensitive information on an intranet that communicates with the Internet is usually protected by a firewall. See also **firewall** and **extranet**.

IP See **Internet Protocol**.

IP address A set of numbers, separated by dots, such as `198.93.93.10`, that specifies the actual location of a machine on an intranet or the Internet.

key database A file that contains the key pair(s) for a server's certificate(s). Also called a key file.

label A component of a UBE filter; it provides a named destination for the actions of other filters.

LDAP See **Lightweight Directory Access Protocol**.

LDAP Data Interchange Format (LDIF) The format used to represent Directory Server entries in text form.

LDAP search string A string with replaceable parameters that defines the attributes used for directory searches. For example, an LDAP search string of `"uid=%s"` means that searches are based on the user ID attribute.

LDIF See **LDAP Data Interchange Format**.

level A designation of logging verbosity, meaning the relative number of types of events that are recorded in log files. At a level of Emergency, for example, very few events are logged; at a level of Informational, on the other hand, very many events are logged.

Lightweight Directory Access Protocol (LDAP) Directory service protocol designed to run over TCP/IP and across multiple platforms. A simplification of the X.500 Directory Access Protocol (DAP) that allows a single point of user and group account management across Netscape servers. The Netscape Directory Server uses the LDAP protocol.

listen port The port that a server uses to communicate with clients and other servers. The Messaging Multiplexor can use both a listen port and a separate backside port.

local part The part of an email address that identifies the recipient. See also **domain part**.

log directory The directory in which all of a service's log files are kept.

log expiration Deletion of a log file from the log directory after it has reached its maximum permitted age.

log rotation Creation of a new log file to be the current log file. All subsequent logged events are to be written to the new current file. The log file that was the previous current file is no longer written to, but remains in the log directory.

mailbox See **folder**.

mail client The programs that help users send and receive email. This is the part of the various networks and mail programs that users have the most contact with. Mail clients create and submit messages for delivery, check for new incoming mail, and accept and organize incoming mail.

mail exchange record See **MX record**.

Mailstone A utility for performing stress tests. The Mailstone utility enables you to perform capacity planning by testing the ability of your messaging server to function properly under maximum loads.

Management Information Base (MIB) A database containing data about managed network objects.

master agent The SNMP agent that exchanges information between the Network Management station (NMS) and its subagents. See also **subagent**.

match criterion A component of a UBE filter; it is a string or expression that represents an envelope or header phrase (such as `"Easy Money"`) to be matched against incoming messages.

mboxutil A command line utility for managing mail folders.

MD5 A message digest algorithm by RSA Data Security. MD5 can be used to produce a short digest of data that is unique with high probability. It is mathematically extremely hard to produce a piece of data that produces the same message digest email.

message The fundamental unit of email, a message consists of a header and a body and is often contained in an envelope while it is in transit from the sender to the recipient.

message delivery When the MTA delivers a message to a local recipient (a mail folder or a program).

message field A component of a UBE filter; it specifies the specific envelope or header item (such as `Subject:`) whose contents are to be matched against incoming messages.

Message Handling System (MHS) A group of connected MTAs, their user agents, and message stores.

message queue The directory where messages accepted from clients and other mail servers are queued for delivery (immediate or deferred).

message quota A limit defining how much disk space a particular folder can consume.

message store The database of all locally delivered messages for a Messaging server instance. Messages can be stored on a single physical disk or stored across multiple physical disks.

message store partition A message store or subset of a message store residing on a single physical file system partition.

Message Transfer Agent (MTA) A specialized program for routing and delivering messages. MTAs work together to transfer messages and deliver them to the intended recipient. The MTA determines whether a message is delivered to the local message store or routed to another MTA for remote delivery.

Messaging Multiplexor A specialized Netscape Messaging Server that acts as a single point of connection to multiple mail servers, facilitating the distribution of a large user base across multiple mailbox hosts.

Messaging Server administrator The administrator whose privileges include installation and administration of a Netscape Messaging Server instance.

Messenger Express A mail client that enables users to access their mailboxes through a browser-based interface. Messages, folders, and other mailbox information are displayed in HTML in a browser window. See also **webmail**.

MHS See **Message Handling System**.

MIB See **Management Information Base**.

MIME See **Multipurpose Internet Mail Extension**.

mmp-setup The Unix filename for the Messaging Multiplexor installer.

MoveUser A command line utility for moving messages in a user's mail folder from one Messaging Server to another.

MTA See **Message Transfer Agent**.

MTA hop The act of routing a message from one MTA to another.

Multiplexor See **Messaging Multiplexor**.

Multipurpose Internet Mail Extension (MIME) A protocol you can use to include multimedia in email messages by appending the multimedia file in the message. Because not all mail clients support MIME, you should make sure that the message recipient has a MIME-enabled mail client.

MX record A type of record stored in a DNS server that maps a domain name to a host name.

net mask A 32-bit value used in conjunction with an IP address to separate the network and subnet IDs from the host ID.

Netscape administrator The administrator whose privileges include installation and administration of all Netscape servers, including the Netscape Directory Server.

Netscape Console The administrator interface from which you administer all Netscape servers.

Netscape Setup The installation program for all Netscape servers and for Netscape Console.

next-hop list A list of adjacent systems a mail route uses to determine where to transfer a message. The order of the systems in the next-hop list determines the order in which the mail route transfers messages to those systems.

notification message A type of message, sent to the postmaster account by the Messaging Server, that is for information al purposes and requires no action from the postmaster. Compare **error message.**

NscpMsg (Unix only) A command line utility for starting and stopping the Netscape Messaging Server and for running recovery utilities.

nsfilter The shared library file that contains the UBE plugin.

partition See **message store partition**.

password authentication Identification of a user through user name and password. Compare **certificate-based authentication.**

pattern A string expression used for matching purposes, such as in Allow and Deny filters.

personal folder A folder that can be read only by the owner. See also **shared folder**.

plain text authentication	See **password authentication.**
plugin	A server extension program, implemented on the Messaging Server as a shared library that uses the Messaging Server Plugin API.
POP3	See **Post Office Protocol Version 3**.
PopMMP.config	A configuration file that defines a POP3 instance of the Messaging Multiplexor.
PopMMP.sh	A Unix shell script that sets configuration parameters and executes the POP3 Messaging Multiplexor.
PopProxy	The executable file for the POP Messaging Multiplexor.
port number	A number that specifies an individual TCP/IP application on a host machine, providing a destination for transmitted data.
postmaster	By convention, an account used to communicate with the person (or people) responsible for maintaining a messaging server.
Post Office Protocol Version 3 (POP3)	A protocol that provides a standard delivery method and that does not require the message transfer agent to have access to the user's mail folders. Not requiring access is an advantage in a networked environment, where often the mail client and the message transfer agent are on different computers.
process	A self-contained, fully functional execution environment set up by an operating system. Each instance of an application typically runs in a separate process. Compare **thread.**
qconvert	A command line utility for converting the Netscape Messaging Server 3.x message queue to the 4.x MTA format.
quota	A command line utility for viewing reports about and fixing message quota usage. See also **message quota**.
RC2	A variable key-size block cipher by RSA Data Security.
RC4	A stream cipher by RSA Data Security. Faster than RC2.
readership	A command line utility for collecting readership information on mail folders.
reconstruct	A command line utility for reconstructing mail folders.
regular expression	A text string that uses special characters to represent ranges or classes of characters for the purpose of pattern matching.

routing The act of transferring a message from one MTA to another when the first MTA determines that the recipient is not a local account, but might exist elsewhere. Routing is normally configurable only by a network administrator. See also **forwarding**.

routing tables The internal databases that hold the information about message originators and recipients. See also **SMTP mail routing table**.

RUN action A special action of the UBE plugin; it calls external programs that extend the plugin.

schema Definitions of the types of information that can be stored as entries in the Netscape Directory Server. When information that does not match the schema is stored in the directory, clients attempting to access the directory might be unable to display the proper results.

search base See **base DN**.

Secure Sockets Layer (SSL) A software library establishing a secure connection between two parties (client and server).

security-module database A file that contains information describing hardware accelerators for SSL ciphers. Also called *secmod*.

sendmail A common MTA used on Unix machines. In most applications, the Netscape Messaging Server can be used as a dropin replacement for sendmail. Netscape provides a set of sendmail migrations tools.

server instance The directories, programs, and utilities representing a specific server installation.

server root The directory into which all Netscape servers associated with a given Administration Server on a given host are installed. Typically designated *`serverRoot`*. Compare **installation directory**, **instance directory**.

service (1) A background process on Windows NT that does not have a user interface. Netscape servers on Windows NT platforms run as services. Equivalent to **daemon.** (2) A function provided by a server. For example, the Netscape Messaging Server provides IMAP, POP, and SMTP services.

session An instance of a client-server connection.

shared folder A folder that can be read by more than one person. Only IMAP folders can be shared. Compare **personal folder**.

Simple Mail Transfer Protocol (SMTP) The email protocol most commonly used by the Internet and the protocol supported by the Netscape Messaging Server.

Simple Network Management Protocol (SNMP) A network protocol that allows administrators to monitor server processes remotely on SNMP-compatible servers through the use of SNMP station software.

SIZE An SMTP command enabling a client to declare the size of a particular message to a server. The server may indicate to the client that it is or is not willing to accept the message based on the declared message size; the server can declare the maximum message size it is willing to accept to a client. Defined in RFC 1870.

SMTP See **Simple Mail Transfer Protocol**.

smtpAccept The stage of SMTP message processing that involves accepting messages sent from clients or other servers.

smtpDeliver The stage of SMTP message processing that involves transferring messages to other servers.

SMTP mail routing table Provides a way to redirect mail based on the domain to which it is being sent. Each entry in the SMTP Mail Routing table consists of a pattern and a domain. Before sending a message, the destination domain is compared to the patterns in the table. If a match is found, the destination host is replaced by the domain corresponding to the pattern that matched.

SNMP See **Simple Network Management Protocol**.

spoof message A message that the Messaging Multiplexor can send to a user when the Multiplexor cannot connect to the user's mailbox server.

spoofing Misrepresentation of its host name, domain name, or IP address by a client attempting to gain access to or send a message to a server.

SSL See **Secure Sockets Layer**.

subagent An SNMP agent that gathers information regarding network activity of a particular device, such as the Netscape Messaging Server.

subdomain A portion of a domain. For example, in the domain name corp.airius.com, corp is a subdomain of the domain airius.com. See also **host name** and **fully-qualified domain name.**

subnet The portion of an IP address that identifies a block of host IDs.

TCP See **Transmission Control Protocol**.

TCP/IP See **Transmission Control Protocol/Internet Protocol**.

thread A lightweight execution instance within a process.

Transmission Control Protocol (TCP) The basic transport protocol in the Internet protocol suite that provides reliable, connection-oriented stream service between two hosts.

Transmission Control Protocol/ Internet Protocol (TCP/IP) The name given to the collection of network protocols used by the Internet protocol suite. The name refers to the two primary network protocols of the suite: TCP (Transmission Control Protocol), the transport layer protocol, and IP (Internet Protocol), the network layer protocol.

UA See **user agent**.

UBE See **Unsolicited Bulk Email**.

UBE filter A rule that defines how messages that fit a certain match criterion are to be handled. See also **Unsolicited Bulk Email (UBE).**

UBE plugin An SMTP plugin that applies UBE filters to incoming messages.

Unsolicited Bulk Email (UBE) Unrequested and unwanted email, sent from bulk distributors, usually for commercial purposes.

user (1) A person that makes use of computer software. (2) An account for accessing a server, maintained as an entry on a directory server.

user agent (UA) The client component, such as Netscape Communicator, that allows users to create, send, and receive mail messages.

virtual domain A domain name added by the Messaging Multiplexor to a client's user ID for LDAP searching and for logging into a mailbox server.

VRFY SMTP command for verifying a user name. Defined in RFC 821.

webmail A generic term for browser-based email services. A browser-based client—known as a "thin" client because more processing is done on the server—accesses mail that is always stored on a server. See also **Messenger Express**.

wildcard A special character in a search string that can represent one or more other characters or ranges of characters.

Index

Symbols

.forward files 461

/dev/null directory 461

/etc/aliases files 449
  sendmail compatibility 460
  unix2ldif utility 443

/etc/passwd files 443

/etc/security/shadow files 451

/etc/shadow files 443, 451

/etc/shells file 336, 341

/server-root/bin/msg/admin/bin directory 457

/server-root/userdb/ldap/tools directory 456

/usr/lib directory 458

A

access control
  access to TCP services, overview 189
  client access 78
  creating access filters 198
  filter syntax 191
  HTTP service 78, 189
  IMAP service 78, 189
  message store 154
  POP service 78, 189
  SMTP service 112, 114, 189
  TCP wrappers 189

action field (UBE filters) 220, 221, 230

active queue 115

address completion domain
  sendmail migration 447
  specifying 91
  unix2ldif utility 447

addressing information
  Address tab 107
  alternate addresses 129, 139, 259
  domain completion 91
  envelope rewrite methods 108
  for mailing lists 139
  for mail users 128
  forwarding addresses 132
  From address rewrite style 109
  host name 259
  local domains 92
  primary address 128, 139, 259
  recipient search methods 110
  user attributes 259
  verifying recipients 97

address rewrite style
  for routing to remote MTA 268
  From address 109

administrative domains 45, 63, 186

administrator access control
  configuring 185
  to server as a whole 187
  to server tasks 188

aging policies
  message store 162
  number of days 162
  number of messages 162
  size of mailbox 162
  specifying 162
  to control disk space 283

alarm attributes
  disk space 167
  overview 437
  response time 285

aliases
  sendmail compatibility 460
  sendmail emulator 462

aliases files
  include directives 449

key value format 448
sendmail migration 448
unix2ldif utility 448
alternate email addresses 129, 139
alternate search methods
and performance 290
custom domain 110, 267
routing 266
specifying 110
truncated domain 110, 267
user ID 110, 267
anonymous login 72
anti-relay (UBE filter) 247
anti-relay plug-in
configuring 250
enabling 249
relaying defenses 244
source code 249
A records 28, 275
argument field (UBE filters) 220
authentication
certificate-based 352
HTTP 72
IMAP 72
Multiplexor 352
password-based 146
POP 72
SMTP 112, 146, 173
automatic reply
echo mode 105
reply mode 105, 134
settings 133
specifying default information 104
vacation mode 105

B

banners
IMAP 72
POP 72
SMTP 408
bsmtp command 462

C

CA certificates
installing 180
managing 181
case tag (UBE filters) 219
certificate-based login 74, 184, 352
certificates
installing, server 179
installing, trusted CA 180
managing 181
obtaining 177
requesting, server 178
chkuniq utility
description 453
options 454
ciphers
about 181
selecting 183
command-line utilities
configutil 406
counterutil 410
deliver 410
file locations for 404
getconf script 408
hashdir 412
imscripter 413
mailq 415
mboxutil 417
MoveUser 419
NscpMsg 423
overview 402
processq 424
qconvert 425
quota 427
readership 427
reconstruct 428
setconf script 408
stored 433
upgrade 435
usage requirements 405
configuration directory 28, 34, 64, 65
configutil
command-line utility 305, 406

scripts 408
control directory 116
counterutil command-line utility 410
custom domain, search by 110, 267

D

Database Manager, and sendmail migration 455
dead.letter files 461
deferred
delivery 96
queue processing 103
deferred directory 116
deferred queue 115
delegated administration 186
deliver command-line utility 410
delivery options
deferred delivery 96
mail users 130
POP/IMAP delivery 131
program delivery 94, 131
specifying 93
Unix delivery 132
Unix mail folders 93
deployment considerations
DNS 27
enterprise vs. ISP 32
firewalls 31
LDAP directory 28
mail-account migration 32
overview 25
redundancy 30
separation of services 29
sizing and topology 26
dial-up connections
limiting 289
performance 289
directories
for installed server files 40
for log files 313
message queue 115
message store 150
disk quota
configuring 156
grace period 159
monitoring 167
to control disk space 283
warning message 159
disk space
aging policies 283
controlling usage of 282
disk quotas 283
message size limits 283
monitoring 167, 282
quotas for 156
RAID technology 295
reserving 100
reserving for the message queue 284
DNS
See Domain Name System (DNS)
DNS records
A records 275
MX records 275
PTR records 276
Domain Name System (DNS)
A records 275
deployment considerations 27
message routing
MX records 275
overview 273
PTR records 276
domains
access domain 131
address completion 91
defined 90
Domain Name System (DNS)
language configuration 59
local domain 92
remote domain 90
search methods 110
See also administrative domains
virtual 354

E

echo mode 105, 134

email-only members (of a group) 136
Enable Statistics Collection box 304
encryption
  accelerators for 177
  defined 488
  Multiplexor 351
end-user help, configuring access to 55
entry format, MIB 469
envelope fields (UBE filters) 225
envelope rewrite methods 108
envonly tag (UBE filters) 219
error handling 106
ETRN command 103
events, reporting 300
EXPN command 102
external modules (PKCS #11) 177

F

facility categories (for logging) 310
failover and redundancy 30
file and directory organization 40
filenames
  for installed server files 40
  for log files 311
filtering mail
  program delivery 326
  UBE 215
filters
  for program delivery 326
  for UBE. *See* UBE filters
firewalls 31
forwarding addresses 132, 448
FQDN
  See Fully Qualified Domain Name
free disk space, reserving 100
From address, rewriting 109
Fully Qualified Domain Name (FQDN) 274

G

groups
  See also mailing lists
  email-only members of 136
  mailing lists 122
  Members tab 136

H

hardware configuration, and performance 295
hashdir command-line utility 412
header fields (UBE filters) 225
hiding a host name 129
HKEY_LOCAL_MACHINE 391
host, defined 490
host completion domain
  sendmail migration 447
  unix2ldif utility 448
host name hiding 28, 129, 140
host name resolution
  See IP addresses, reverse lookups
HP OpenView 299, 468
HTTP service
  access control filters 198
  anonymous login 72, 83
  certificate-based login 74
  client access control 78
  configuring 82
  connection settings 83
  connections per process 75
  customizing 86
  disabling 83
  dropping idle connections 77
  enabling 83
  logging out clients 77
  login requirements 72
  message settings 83
  MTA settings 83
  number of processes 74
  password-based login 73, 83
  performance parameters 74
  port numbers 70

process settings 84
proxy authentication 200
security 171
session ID 171
specialized web server 82
SSL port 72
starting and stopping 61, 281
threads per process 76

I

idle connections, dropping 77
ImapMMP 396
ImapMMP.config 374, 383
ImapMMP.sh 376
ImapProxy 396
ImapProxy command 367
IMAP service
access control filters 198
anonymous login 72, 80
banner 72, 81
certificate-based login 74, 184
client access control 78
configuring 80
connection settings 80
connections per process 75
disabling 80
dropping idle connections 77
enabling 80
login requirements 72
number of processes 74
password-based login 73, 172
password-based long 80
performance parameters 74
port numbers 70, 71
process settings 80
readership utility 427
shared folders 23, 427
SSL 71, 175
SSL port 71
starting and stopping 61, 281
threads per process 76
imports, MIB 471
imscripter command-line utility 413
INBOX, default mailbox 417
include alias 461
include directives 449
INSECURE-PROGRAM-DELIVERIES file 331, 339
installation
configurations for Messaging Server 33
directory 40
Messaging Server 38
instance directory 41
instances. *See* server instances
internal modules (PKCS #11) 177
IP addresses, reverse lookups
performance impact 289
specifying 98

K

key value format 448

L

label field (UBE filters) 218
LDAP Data Interchange Format files 442
LDAP directory
configuring lookups in user directory 63
Database Manager 455
sendmail migration 454
viewing settings in configuration directory 65
ldapmodify command 455, 456
LDAP search URLs 142
LDIF files 442
ldifsplit utility
add.ldif 452
mod.ldif 452
options 453
running 451
local domain 92
logging
analyzing logs 320

architecture of 314
directories for log files 313
facility categories 310
filenames 311
format of log entries 312
formats for specific event types 321
levels of 309
Mailbox-Deliver format 323
options 315
services that are logged 308
severity levels 309
SMTP-Accept format 321
SMTP-Deliver format 322
syslog 309, 312
viewing logs 318
logical queue 115
login
anonymous 72
certificate-based 74, 184
password-based 73, 172

M

mail accounts. *See* mail users
mailAlternateAddress attribute 259
mailboxes
aging policies for 162
default mailbox for delivery 417
INBOX 417
managing 166
mboxutil utility 166
naming conventions for 417
quota usage 427
reconstructing 428
reconstruct utility 166
repairing 166
size and performance 290
mail exchange record 275
mailHost attribute 259, 268
mailing lists
accessing an existing group 138
adding list (email-only) members 144
address (primary) 139
creating a new group 135
duplicate checks for unique members 272
dynamic membership criteria 142
email-only members 136
expansion and delivery 270
host name hiding 140
LDAP search URLs 142
list members 141
list owners 141
Mail tab 136, 139
Members tab (of group) 136
message-rejection actions 146
nested lists 273
Netscape Console access to 135
recipients per envelope 273
restrictions on message posting 145
routing to dynamic members 272
routing to email-only members 272
routing to unique members 272
verifying 102
mailq command-line utility 415
mailRoutingAddress attribute 108, 259, 268
Mail tab 124, 128, 136, 139
mail users
accessing an existing user 127
address (primary) 128
addresses, specifying 128
alternate addresses 129
auto-reply mode 134
auto-reply settings 133
creating a new user 124
delivery-options configuration 130
echo mode 134
forwarding addresses for 132
host name hiding 129
Mail tab 124, 128
migration of accounts 32
Netscape Console access to 124
POP/IMAP delivery option 131
program delivery option 131
Unix delivery option 132
vacation mode 134
managed device 299
managed device-initiated communication 300
See also traps

managed objects
See also variables
communication with 300
defined in MIB 472
MIB 468
Management Information Base
See MIB
master agents
and Admin Server 299
in SNMP 299
installation 299
in the MIB 468
match criteria (UBE filters) 223
mboxutil command-line utility 417
Members tab 136
message field (UBE filters) 218, 227
message queue
active queue 115
alternate paths for queue storage 115, 118
control directory 116
deferred delivery 96
deferred directory 116
deferred queue 115
distribution for improved performance 291
logical 115
mailq utility 415
messages directory 116
physical 115
processq utility 424
message routing
A records 275
Directory Server 258
DNS 273
mailing list expansion 270
MX records 275
PTR records 276
routing attributes 259
SMTP routing table 264
messages directory 116
message size
and performance 289
limiting 103
SIZE command 103
message store
access control 154
administrator access 154
aging policies 162
cleaning up messages 154
configuring disk quotas 156
configuring partitions 160
default partition 161
deleting messages 154
directory layout 150
distribution for improved performance 291
expunging messages 154
maintenance and recovery procedures 165
mboxutil utility 417
MoveUser utility 419
overview 149
partitions 159
primary partition 160
ProxyAuth user 154
RAID technology 161
reconstruct utility 428
stored utility 165
Message Transfer Agent (MTA)
HTTP settings 83
routing to remote MTA 268
See also SMTP service
specifying number of hops 99
thread settings and performance 292
MIB
activating 468
and subagent 301
data types of MIB variables 470
format of entries 469
imports 471
installation 468
MADMAN 468
managed objects 468, 472
Messaging Server MIB file 300, 476
MTA attributes 473
MTA table 473
MTA variables 473
object identifier 468
static variables 472
traps 301, 474
variables 300, 301, 468, 472

MigrateUnixSpool utility
administrators 457
example 458
location of 457
syntax of 457
users 457
mmp-setup 374
MoveUser command-line utility 419
Multiplexor
$CONFIG_FILE 374
authentication 352
BacksidePort 360
Banner 360
BaseDN 360
BindDN 355, 360
BindPass 355, 360
CanonicalVirtualDomainDelim 355, 361
Capability 361
certificate-based authentication 352, 353
CertMapFile 361
certmap plugins 352
command line options 367
configuration (NT) 390
configuration (Unix) 378
configuration parameters 359
configuration prompts (NT) 391
DNComps 352
features 348
files (NT) 387
files (Unix) 373, 374
FilterComps 352
how it works 350
ImapMMP 374, 396
ImapMMP.config 374, 383
ImapMMP.sh 376
ImapProxy 367, 396
ImapProxy file 374
installation (NT) 386
installation (Unix) 372
instances (multiple) 356
LDAPHost 361
LdapURL 355
ListenPort 362
LogDir 362
LogLevel 362
MailHostAttrs 362
message topology 384
mmp-instancename 387
MMPRoot (Unix) 373
mmp-setup 374, 375
netscape.mp.conf 377
NumThreads 363
PopMMP 374, 376, 396
PopMMP.config 374
PopProxy 396
PopProxy command 367
PopProxy file 374
port number (Unix) 379
PreAuth 355, 363
pre-authentication 353
removing (NT) 399
removing (Unix) 398
running (NT) 397
running (Unix) 395
ServerDownAlert 363
SpoofMessageFile 363
SSLBacksidePort 363
SSLCertFile 364, 370
SSLCertName 364
SSLCertNickname 371
SSLCipherSecs 364
SSLCipherSpecs 371
SSL command line options 370
SSLEnable 365
SSLKeyFile 365, 370
SSLKeyPasswd 365, 371
SSLListenPort 365
SSLSecmodFile 365, 370
StoreAdmin 366
store administrator 352
UidSearch 355, 366
uninstalling 398
vdmap 354
VDomain 355
VirtualDomainDelim 366
VirtualDomainFile 366
virtual domains 354
MX records 28, 275

N

negation modifier (UBE filters) 230

netscape.mp.conf 377

Netscape Console
  Messaging Server access from 46
  performing all tasks with 49
  performing typical tasks with 47
  using 44

netscape-mail.mib 468

network information
  exchanging 300
  monitoring 299
  PDUs 300
  requests for 300
  verifying 305

network management station (NMS)
  and master agent 301
  and SNMP 299
  and subagent 301
  defined 468
  network management station-initiated communication 300

nicknames
  message store partitions 161
  sendmail migration 447

NIS and sendmail migration 450

NMS
  See network management station

NscpMsg command-line utility 423

nsmailagent, SNMP subagent on Unix 301

O

object identifier, MIB 470

P

partitions
  adding 161
  configuring for message store 160
  default 161
  message store 159
  nicknames 161
  pathnames 161
  primary 160
  RAID technology 161

passwd files
  and sendmail migration 450
  unix2ldif utility 450

password authentication
  See also login
  for mailing-list posting 146
  HTTP service 73
  IMAP service 73
  POP service 73
  SMTP service 173
  to LDAP user directory 65

password file (for SSL) 178

password login 73, 172

pathnames, to installed files and directories 40

performance
  address lookups per message 294
  address rewrite style 290
  administration server activity 293
  alternate search methods 290
  and dial-up connections 289
  configuration of logging services 290
  directory server activity 294
  disk speed 291
  hardware configuration 295
  mailbox size 290
  message queue distribution 291
  message size 289
  message store distribution 291
  MTA thread settings 292
  overview 286
  plug-in API 290
  POP, IMAP, and HTTP services 287
  RAID technology 295
  ratio of outbound sends 295
  reverse IP address lookups 289
  server locations 294
  SMTP services 288
  users per disk 287
  verifying recipient addresses 289

performance parameters

connections per process 75
number of processes 74
threads per process 76
physical queue 115
PKCS #11
internal and external modules 177
plugins.cfg file 236
plug-ins. *See* SMTP plug-ins, UBE filters
PopMMP 376, 396
PopMMP.config 374
PopProxy 396
POP service
access control filters 198
banner 72
certificate-based login 184
client access control 78
configuring 78
connections per process 75
dropping idle connections 77
login requirements 72
number of processes 74
password-based login 73, 172
performance parameters 74
port numbers 70
SSL 175
starting and stopping 61, 281
threads per process 76
postmaster account
checking 279
defined 279
pre-authentication (Multiplexor) 353
primary email address 128, 139
processes
number of 74
starting and stopping 281
processq command-line utility 424
program delivery
/etc/shells file 336, 341
batch files 330
changing programs 333, 335
default mode 339
designating programs 331, 332
disabling, Unix 343
disabling, Windows NT 346
enabling 330
failures of 327, 333
INSECURE-PROGRAM-DELIVERIES file 331, 337, 339
installing programs 341, 345
links, Unix 331
mailboxes 327
modes of operation 329
multiple programs 333, 335
non-secure mode 329, 331, 333, 335, 338, 339
overview of 325
paths 332, 335
POP/IMAP delivery 332
properties dialog box 332
restrictions in programs 338, 344
root, running programs as 340, 342
scripts 330
secure mode 329, 332, 334, 338, 339
security 327, 328, 330
sendmail compatibility 461
setup, Unix 340
setup, Windows NT 344
shells, valid list of (Unix) 341
sorting mail 326
specifying 131
suspending, Unix 342
suspending, Windows NT 345
SUSPEND-PROGRAM-DELIVERIES file 342
trusted directory 328, 329, 331
trusted programs 328
Unix delivery 332
Unix shells 336
protocol data units (PDUs) 300, 474
protocol-level SMTP plug-ins 203, 212
ProxyAuth user
HTTP service 200
message store 154
MoveUser utility 421

Q

qconvert command-line utility 425

quota command-line utility 427
quotas, disk space 156

R

RAID technology
  for message store 161
  performance 295
readership command-line utility 427
recipient addresses
  envelope, rewriting 108
  verifying 97
  verifying, and performance 289
reconstruct command-line utility 428
recovery tasks
  mailboxes 166
  overview 285
  reconstruct utility 428
redundancy and failover 30
regular expressions (in UBE filters) 223
relaying 216, 245
reply mode 105
requests for network information 300
reserving free disk space 100
response time
  alarm attributes 285
  improving 285
  overview 284
reverse IP address lookups 98
root
  program delivery 337, 342
  running programs as (program delivery) 340
routing aliases
  sendmail 446
  unix2ldif utility 446
routing messages
  Address tab 107
  mail exchange 275
  overview 258
  routing attributes 263
  specifying routing information 107
  to remote MTA 264
routing table, editing entries 111
RUN action (UBE filters) 223

S

Safe user ID box 342
search methods
  custom domain 110, 267
  specifying alternate 110
  truncated domain 110, 267
  user ID 110
security
  about 170
  certificate-based login 74, 184
  client access controls 78
  client access to TCP services 189
  HTTP service 78, 171
  IMAP service 78
  password-based login 73
  POP service 78
  program delivery 327, 328
  SMTP service 112
  SSL 175
sendmail.cf files 464
sendmail command
  starting Messaging Server 459
  V8 465
sendmail compatibility
  aliases 460
  alternate names 462
  bsmtp command 462
  command line 458
  functional compatibility 460
  include alias 461
  lists (mailing) 461
  mail, delivering to files 461
  mail, delivering to programs 461
  mail forwarding 460
  mailing lists 461
  mailq command 462
  Messaging Server, starting 459
  names of (alternate) 462
  network interface 460

newaliases command 462
overview 458
sendmail command 462
sendmail emulator
aliases 462
alternate names 462
mailq command 459
names, alternate 462
operating modes 460
options 462, 463, 464
sendmail.cf files 464
V8 options 465
sendmail migration
address completion domain 447
addresses, duplicate 453
aliases files 448
and NIS maps 450
basic steps 441
chkuniq utility 453
DNs, duplicate 453
host completion domain 447
LDAP directory, updating 454
LDAP entries, creating/converting to 442
ldapmodify command 455, 456
LDIF files 442
ldifsplit utility 451
location of 443
mail, moving 456
MigrateUnixSpool utility 456
nicknames 447
passwd files 450
routing aliases 446
spool files, mail moving 456
steps 441
syntax 443
unix2ldif utility 442
user IDs, duplicate 453
server certificates
installing 179
managing 181
requesting 178
Server Configuration tab 49
server group 34
server information, viewing 54
server instances 36
server root 33, 37, 40
Server Tasks tab 47
service banners 72
services
enabling and disabling 70
HTTP 69
IMAP 69
logging of 308
POP 69
SMTP 90
starting and stopping 61, 281
setconf script 408
severity levels (of logging) 309
shared folders, IMAP 23, 427
Simple Mail Transfer Protocol
See SMTP service
site language configuration 58
SIZE command 103
sizing considerations 26
SMTP-Accept
and plug-ins 204, 205
log format for 321
SMTP authentication 146
SMTP-Deliver
and plug-ins 204
log format for 322
SMTP plug-ins 203
activating/deactivating 208
API for 206
configuring 209, 210
installing 206, 210
manual configuration of 210
Netscape Console configuration of 206
pre-SMTPaccept 203, 212
protocol-level 203, 212
SMTP-Accept 204, 205
SMTP-Deliver 204
uninstalling 207, 212
SMTP routing table 264
SMTP service

access control filters 198
authenticated SMTP 112, 173
banner 408
certificate-based login 175
deferred delivery 96
enabling optional features 100
ETRN command 103
EXPN command 102
login requirements 173
message queue 114
MTA hops 99
overview 90
password-based login 173
port number 175, 257
routing overview 255
routing table 264
routing table entries 111
See also SMTP plug-ins
SIZE command 103
starting and stopping 61, 281
VRFY command 101
SNMP agents
See master agents *and* subagents
SNMP settings, checking 305
SSL
certificates 177
ciphers 181
enabling 181
hardware encryption accelerators 177
installing CA certificates 180
installing server certificates 179
internal and external modules 177
managing certificates 181
overview 175
password file for 178
requesting server certificates 178
turning on 183
sslpassword.conf file 178
stored command-line utility 433
subagents 301
in SNMP 300
in the MIB 469
nsmailagent 301
retrieving information 301
starting 304
stopping 304
SUSPEND-PROGRAM-DELIVERIES file 342, 345
syslog 309, 312

T

TCP client access control
address-spoofing detection 197
examples 196
EXCEPT operator 194
filter syntax 191
host specification 194
how access filters work 189
identd service 195, 197
Netscape Console interface for 198
overview 189
username lookup 195, 197
virtual domains 198
wildcard names 192
wildcard patterns 193
TCP wrappers 189
threads per process 76
topologies for deployment 26, 29, 30, 32
traps
defined 474
listed 475
managed device-initiated communication 300, 474
MIB 301, 474
nsMailServerDown 475
nsMailServerNoResponse 475
nsMailServerStart 475
server events 474
truncated domain, search by 110, 267
trusted directory (program delivery) 328

U

UA. *See* User Agent (UA)
UBE (Defined) 215
UBEfilter.cfg file 236

UBEfilter.opt file 236
UBE filters 215
See also UBE plug-in
action field 220, 221, 230
activating/deactivating 234
anti-relay 247
argument field 220
case tag 219
changing order of 234
comments in 238
configuration files 236
creating 231, 237
editing 233, 237
envelope vs. header fields 225
envonly tag 219
examples of 239
extending, with RUN action 242
extending, with shared library 243
format of 218
label field 218
manual configuration of 235
match criteria 219
match criterion 223
message field 218, 227
negation modifier 230
Netscape Console configuration of 230
omitting parts of 237
options 235
overview 217
regular expressions in 223
RUN action 223
special names (for message field) 227
UBE defined 215
UBE plug-in 216, 236
UBE plug-in
activating/deactivating 231
extending 242
overview 216
UidSearch 355
unix2ldif utility
address completion domain 447
aliases files 448
aliases files, multiple passes 449
forwarding alias LDAP entries 448
HostCompletionDomain 448
host completion domain 448
input sources 443
Mail Group LDAP entries 448
mailHost value 446
NIS maps 450
passwd files 450
routing aliases 446
shadow files 451
Unix delivery 132, 332
unsolicited bulk email. *See* UBE filters
upgrade command-line utility 435
User Agent (UA) 90
user directory 28, 34, 63
user ID, search by 110, 267
user login. *See* login
user names, verifying 101
users
See also mail users
mail accounts 122

V

vacation mode 105, 134
variables
MIB 301, 468, 472
MTA table variables 473
static variables 472
vdmap (Multiplexor) 354
VDomain 355
verbosity (of logging) 309
verifying
mailing list 102
recipient addresses 97
SNMP settings 305
user names 101
virtual domains
controlling access to 198
Multiplexor 354
properties of 355
VRFY command 101