Sun Internet Mail Server™ 4.0 Provisioning Guide

THE NETWORK IS THE COMPUTER™

A Sun Microsystems, Inc. Business
901 San Antonio Road
Palo Alto, CA 94303 USA
650-960-1300 fax 650-969-9131

Part No.: 805-7674-10
Revision A, July 1999

Contents

Figures

Code Samples

Preface

This guide describes how to provision users in the Sun Internet Mail Server 4.0 directory.

Audience

This book is intended for people who want to develop their own customized SIMS provision tools. These include those who want to interface SIMS and its naming service (LDAP directory) to an existing source of user / group / domain information (for example, a company database or order entry system).

Readers are expected to be familiar with LDAP programming and email system concepts. Some basic information on these topics can be found in the *SIMS Concepts Guide*.

How This Book Is Organized

Chapter 1, "Preliminaries," is an overview of provisioning a user. For a more complete and detailed description, refer to the Sun Internet Mail Server 4.0 Concepts Guide.

Chapter 2, "Creating Domains," explains how to set up and provision a hosted domain.

Chapter 3, "Creating Mail Users," explains how to set up a user and provision that user for a hosted domain. Administrative tasks will also be discussed.

Chapter 4, "Creating Groups explains how to set up a group, or distribution list, and provision that group for a hosted domain. Administrative tasks for groups will also be discussed.

Chapter 5, "Creating SIMS Administrators," explains how to set up a service for a hosted domain.

Glossary is a list of words and phrases found in this book and their definitions.

Related Information

The following books are related to Sun Internet Mail Server 4.0. Included in this documentation set are:

- *Sun Internet Mail Server 4.0 Concepts Guide* – Provides a conceptual understanding of the SIMS product. By understanding how SIMS works on a conceptual level, readers will more easily understand the administrative tasks described in the *SIMS System Administration Guide* and *SIMS Reference Manual*.
- *Sun Internet Mail Server 4.0 Provisioning Guide* – Describes how to provision the SIMS LDAP directory with users, distribution lists, administrators, and domains by creating and importing LDIF records.
- *Sun Internet Mail Server 4.0 Advanced Installation Guide* – Describes the planning and installation procedures for the Sun Internet Mail Server (SIMS) 3.5 software on Solaris SPARC and Intel-based x86 systems. In particular, it describes the installation of the software using the Graphical User Interface (GUI).
- *Sun Internet Mail Server 4.0 Administrator's Guide* – Describes how to fine-tune the default configuration, and maintain, monitor, and troubleshoot your mail server using the Administration Console, a GUI.
- *Sun Internet Mail Server 4.0 Reference Manua*l – Provides in-depth information about Sun Internet Mail Server. Many administrative functions can also be accomplished through command line utilities. Other advanced functions can be accomplished only through these utilities and by editing configuration files.
- *Sun Internet Mail Server 4.0 Delegated Management Guide* – Describes the SIMS Delegated Management Console and the tasks associated with the console. In particular, it describes how a delegated administrator for a hosted domain performs tasks on users and distribution lists.
- Reference manual pages (man pages) – Describes command-line utilities and detailed information about the arguments and attributes relevant to each command.

- *Sun Directory Services 3.1 Administration Guide* (http://docs.sun.com:80/ab2/coll.297.1/@Ab2CollToc?subject=sysadmin) - Describes the Sun Directory Services.
- *Netscape Directory Services documentation* (http://home.netscape.com/eng/server/directory/) - Describes the Netscape Directory Services.
- *Web Access Administrator's Guide* – Describes the core system administration tasks for the Sun Web Access software.
- Sun Internet Mail Server 4.0 Release Notes – Covers open issues and late-breaking installation, administration, and reference information that is not published in the product books.
- Sun Internet Mail Server 4.0 Web site (located at `http://www.sun.com/sims`) offers up-to-date information on a variety of topics, including:
  - On-line product documentation and late-breaking updates
  - Data sheets and evaluation guide
  - Technical white papers
  - Product demos
  - Press coverage and customer success stories
  - Client solutions

What Typographic Changes Mean

The following table describes the typographic changes used in this book.

TABLE P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output.	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% You have mail.`

TABLE P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	What you type, contrasted with on-screen computer output.	`machine_name% su` `Password:`
AaBbCc123	Command-line place holder: replace with a real name or value.	To delete a file, type `rm` *filename*.
AaBbCc123	Book titles, new words or terms, or words to be emphasized.	Read Chapter 6 in *User's Guide*. These are called *class* options. You *must* be root to do this.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P-2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	#

Note – Although the majority of commands can be run without special superuser permissions, some commands can be performed only as `root`. These commands include: `imta dirsync`, `imta start`, `imta stop`, and `imta restart`. Other commands that require `root` privileges are noted within the document.

Notice

To better illustrate the process being discussed, SIMS manuals contain examples of data that might be used in daily business operations. The examples might include names of individuals, companies, brands, and products. SIMS manuals use only fictitious names, and any similarity to the names of individuals, companies, brands, and products used by any business enterprise is purely coincidental.

CHAPTER 1

Preliminaries

Provisioning simply means adding users to a SIMS mail system. Having a plan for provisioning is particularly important before you install SIMS. SIMS includes a set of provisioning tools. These are:

- SIMS Administration Console - A browser-based console for performing a variety of configuration and provisioning tasks.
- Delegated Management Console - A browser-based console for adding, modifying and deleting users and groups. This tool is designed for local control of hosted domains.
- SIMS Command Line Utilities - Enables provisioning from a command line interface. You can do bulk loading with these utilities.
- NIS / NIS+ Bulk Loading Scripts - Bulk load and synchronize users / groups with a NIS / NIS+ database using a set of custom-designed scripts and processes. Refer to the sections on populating the directory in the *SIMS System Administration Guide*.
- SIMS Provisioning Guide - This book, which explains how to create LDIF records for performing common provisioning tasks.

You will use tools appropraite for your particular situation. This guide is targeted for those sites where the primary repository of subscriber information is something other than SIMS LDAP directory. The guide provides the detailed information about how users / domains / groups are provisioned in SIMS LDAP directory. We expect this information to be used by sites in developing custom provisioning tools to keep the data in their order entry system and SIMS LDAP directory synchronized.

For example, a site may have an order entry system based on an existing database. You will need to plan for adding users / domains / groups currently in the database to SIMS and also take changes to the order entry database and reflect those changes in SIMS LDAP directory.

Setting Up an Internet-Style DC Tree

SIMS supports the Sun Directory Service and the Netscape Directory Service. The directory service supports the storage and retrieval of data in the SIMS messaging system, including user profiles, distribution lists, access control meta-data, and configuration attributes of services.

The SIMS directory is an LDAP-based system. LDAP directory entries and their relationships can be shown in a “tree” model as shown in the diagram below.

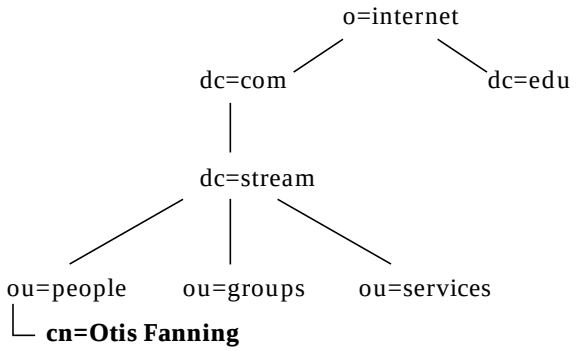


FIGURE 1-1 Directory Information Tree Example.

Data in a DC Tree Model

Data is stored in the directory as directory entries, which consist of object classes and the required/optional attributes of those object classes. Thus, an e-mail user is represented in the directory by an entry which stores information about that user. One of an entry’s object classes are *structural*, that is, an object class that determines the entry type and which cannot be changed. The other object classes are called *auxiliary* object classes, and may be added or deleted to define additional services to an entry.

Directory entries are identified by a unique name, called a *distinguished name (DN)*. The DN is the unique entry that shows the entries location in the DC tree. This tree model is similar to that of most file systems. The root node of the tree is represented by o=internet.

The second-level nodes below the root correspond to the top-level domains in the DNS namespace. In the example above, the DN for the top-level domains have the following suffixes:

- dc=com, o=internet
- dc=edu, o=internet

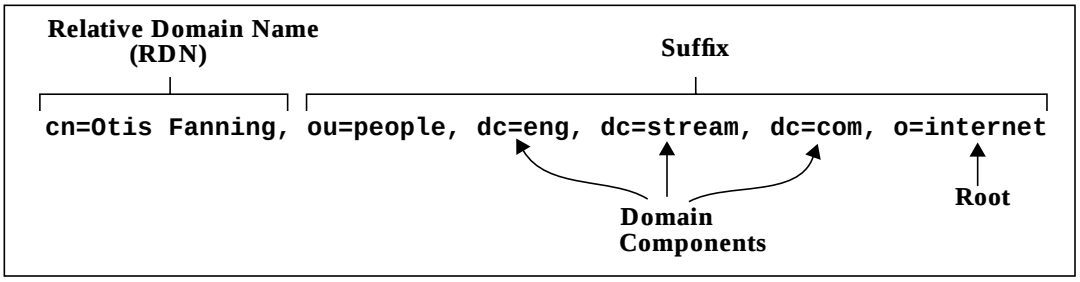
The structure of the DN includes a *relative distinguished name* (RDN). The RDN is the left-most attribute data pair in the DN. The RDN must be a unique value, so you avoid collisions of multiple entries with the same DN. Remember, each DN must be unique, and each RDN must be unique also.

Each successive attribute value pair following the RDN is the RDN of the next parent branch node in the tree hierarchy. The final, right-most attribute value pair represents the conceptual root point of the DIT. The entire string is referred to as the suffix. The RDN is shown in a suffix string with the format cn= xx, where cn is the *common name.*

The hosted domain or organization nodes below the top-level domain nodes are required to have the following organizational containers:

- ou=people - contains user entries for the hosted domain
- ou=groups - contains the distribution list entries for the hosted domain
- ou=services - contains the service state entries for the hosted domain

So, in the following example, the distinguished name is shown for Otis Fanning as shown in FIGURE 1-1.



LDIF Notes

LDIF (LDAP Data Interchange Format) is the standard text-based format for describing directory data. It is used when exporting data from and importing data to the LDAP directory server. We recommend purchasing a good book on LDAP programming to use the manual most effectively. Here are some other LDIF notes:

- Capitalization is not significant in LDIF records, hence `inetMailGroupStatus` is the same as `inetmailgroupstatus`. In this book we generally use capitalization to make attributes more readable.
- The order of LDIF statements in a record after the DN is not significant. That is, the distinguished name (DN) of the entry must be the first attribute-value pair in an LDIF record. After that, all other attribute-value pairs can appear in any order. For readability, the `objectClass` statements are often placed at the beginning, followed by the other attribute-value pairs.
- If an attribute line is more than one line, it must start with a blank space.
- If multiple records are described in a file, a blank line is required between each record.
- Location of LDAP commands for the Sun Directory Server: `/opt/SUNWconn/bin/`
- Format of LDIF record for adding an entry (`ldapadd()`):

```
dn: <dn of entry to be added>
changetype: add
<attribute type>: value
...
```

- Format of LDIF record for deleting an entry (`ldapdelete()`):

```
dn: <dn of entry to be deleted>
changetype: delete
```

- Format of LDIF record for modifying an entry (`ldapmodify()`):

```
dn: <dn of entry to be modified>
changetype: modify
<modify type> <attribute type>
<attribute type>: value
-
...
```

where `<modify type>` can be `add`, `delete`, or `replace`.

Example: The following LDIF file modifies Debbie Gagliano's entry by adding a new mail alias, replacing her surname attribute, and removing her phone number.

```
dn: cn=Debbie Gagliano,ou=People,dc=stream,dc=com,o=internet
changetype:modify
add rfc822MailAlias
rfc822MailAlias: dmizawa@stream.com
-
replace sn
sn: Mizawa
-
delete telephoneNumber
telephoneNumber: 650-767-7777
```

- `ldapmodify()` example format:

  # **`ldapmodify -D "<DN of admin>" -w <passwd> -f <ldif file>`**

- Searching for and viewing an entry using `ldapsearch()` example:

  Person:
  % **`ldapsearch -b "o=internet" "uid=fanning"`**

  Domain:
  % **`ldapsearch -b "o=internet" "dc=*"`**

CHAPTER 2

Creating Domains

This section discusses how to create the domains and organizational units needed to provision those domains. Note that domain components in the DIT directly mirror the domain components and hierarchy of the DNS.

Throughout the section, an example of a DC tree will be used, with the created domain example specified in bold. For each task the relevant tree information will be shown and the required object classes and attributes will be described.

Note – The attribute descriptions in this guide are brief overviews. For the full attribute descriptions refer to the schema section in the *SIMS Reference Manual*.

Creating a Root Entry

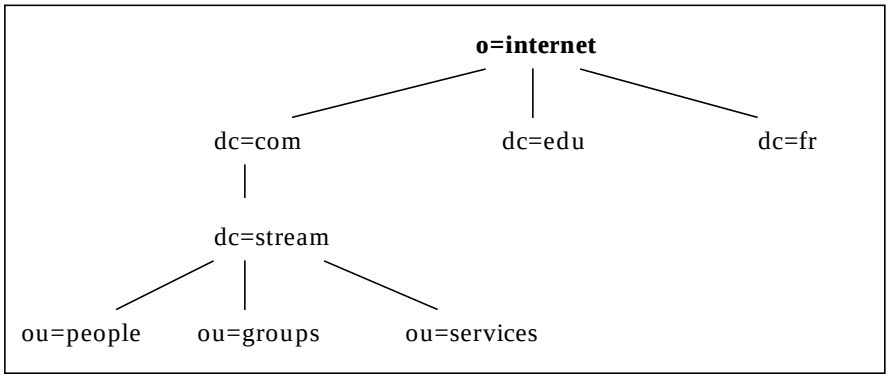


FIGURE 2-1 Root Node 0=internet.

The root entry is the top level node of the DIT. It must always be `o=internet`. The LDIF record for creating the root node is shown in CODE EXAMPLE 2-1. Note that `dc=com,o=internet` is created during SIMS installation, but we show these steps for instructional purposes.

Note – It is a common practice to specify an alias for certain common attributes. These are done in the attributes definition files (`*.at.conf`). Common aliases include `cn` (`commonname`), `ou` (`organizationalUnit`), `o` (`organization`), `sn` (`surname`), `dn` (`distinguishedName`)

CODE EXAMPLE 2-1 LDIF Record for Creating a Root Node

```
dn: o=internet
objectClass: organization
o: internet
```

Root Entry Attributes

This section provides brief descriptions of the root entry attributes. For more complete descriptions of the attributes refer to the SIMS schema section in the *SIMS Reference Manual*.

- `dn: o=internet`

  The distinguished name (dn) uniquely identifies the directory entry in the tree. When creating an LDIF record, the dn must be the first field.

- `objectClass: organization`

  The root node of the DC tree is defined by the object class `organization`. The object class allows you to add other attributes to the entry (see the *SIMS Reference Manual* for details), but only "`o`" is required. `o` must have the same value as set in the `dn` of this entry.

- `o: internet`

  This is the root entry for all DC trees. o stands for organization name, which in this case is `internet`.

Creating a Top-level Domain Entry

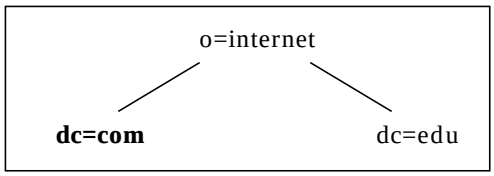


FIGURE 2-2 Top-level Domains.

Top-level domain component entries are created just below the root `o=internet`, and correspond to the top-level DNS domain nodes. In this example they are `dc=com` and `dc=edu`, but they may also include top-level domains in the DNS hierarchy such as `dc=fr`, `dc=jp`, `dc=org`, and so on.

Note that each entry must be created in a separate LDIF record (multiple records may be created in the same file by separating each record with a blank line). For example, you cannot create a top-level domain without first creating a root domain, and you cannot create a second-level domain without first creating a top-level domain.

In this example our top-level domain is `dc=com`. Note that `dc=com,o=internet` is created during SIMS installation, but we show these steps for instructional purposes. The LDIF record is shown below.

CODE EXAMPLE 2-2 LDIF Record for Creating a Second-level Domain

```
dn: dc=com,o=internet
objectClass: domain
dc: com
```

Top-level Domain Attributes

This section provides brief descriptions of the top-level domain attributes. For more complete descriptions of the attributes refer to the SIMS schema in the *SIMS Reference Manual*.

- `dn: dc=com, o=internet`

  The distinguished name (dn) uniquely identifies the directory entry in the tree.

- `objectClass: domain`

  The object class `domain` is used to create this LDAP entry.

- `dc: com`

  The DNS domain component, `com`, is the name of the matching top-level node. For example, to create the hierarchy for stream.com. you need to create a domain component entry corresponding to `.com`.

Creating a Hosted Domain Entry

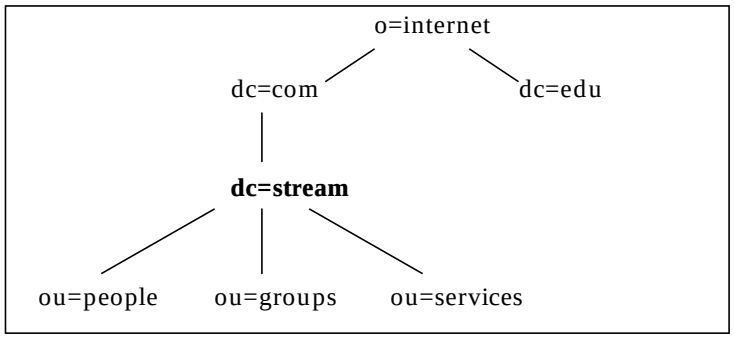


FIGURE 2-3 Third-level Domain stream.com.

Hosted domains are typically created at the third level of the DC tree, which in this example is dc=stream. Note that the node itself is not useful until you create the required containers below it (See “Create the Required Containers for Hosted Domains” on page 19). The LDIF code for creating a hosted domain is shown in CODE EXAMPLE 2-3.

CODE EXAMPLE 2-3 LDIF Record for Creating a Hosted Domain.

```
dn: dc=stream,dc=com,o=internet
objectClass: domain
objectClass: inetDomain
objectClass: simsDomain
description: DC node for stream.com hosted domain
dc: stream
inetTreeStyle: DC
inetDomainStatus: active
inetAuthorizedServices: imap
inetAuthorizedServices: pop3
inetAuthorizedServices: imaps
inetAuthorizedServices: pop3s
inetAuthorizedServices: smtp
inetAuthorizedServices: sunw_webaccess
owner: cn=Mike Marola (Delegated Admin),ou=People,dc=stream,dc=com,o=internet
simsRecursive: 0
simsDomainVersion: 1.0
dnsDomainName: stream.com
rfc822Postmaster: deladmin@stream.com
```

CODE EXAMPLE 2-3 LDIF Record for Creating a Hosted Domain. *(Continued)*

```
mailHosts: route66.stream.com
preferredMailhost: route66.stream.com
domainDiskQuota: 10G
maxMailboxes: 10
maxDistributionLists: 2
maxEntries: 12
```

Hosted Domain Attributes

This section provides brief descriptions of the hosted domain node attributes. For more complete descriptions of the attributes refer to the SIMS schema in the *SIMS Reference Manual*.

- `dn: dc=stream, dc=com, o=internet`

  The distinguished name (dn) uniquely identifies the directory entry in the tree. It consists of a comma separated list of the hierarchical components that specify the entry's location in the DIT. Note that the hosted domain component, `dc=stream` must match the DNS node for the hosted domain.

- `objectClass: domain`
  `objectClass: inetDomain`
  `objectClass: simsDomain`

  These three lines specify the object classes required to create the `dc=stream` entry in the DIT. `domain` is the structural object class and provides attributes useful for describing the domain component nodes of the DC tree.

  `inetDomain` is an auxiliary object class that provides attributes for describing the additional properties of a hosted domain. This object class is associated with directory containers which correspond to a DNS domain. In an internet style DIT, this object class is associated with every domain component node (except the top-level domain, for example, `com`) that represents a DNS domain.

  `simsDomain` is an auxiliary object class that provides attributes useful for describing the additional properties for an e-mail domain. Like `inetDomain`, this object class is associated with entries which correspond to a DNS domain. In an internet style DIT, this object class is associated with every domain component node that represents a DNS domain.

domain Attributes

- `description: DC node for stream.com hosted domain`

Free form text. Description about the organization node in the directory. Usually the full name of the organization that is associated with the value of the attribute `organizationName` for this entry.

- `dc: stream`

  The dc (domain component) is the associated DNS domain for this node.

inetDomain Attributes

- `inetTreeStyle: DC`

  Defines the type of tree associated with this DIT. There are possible two values for this field: OSI and DC. The single Domain Component (DC) tree style is the default since SIMS 4.0 namespace maps to the DC tree style.

- `inetDomainStatus: active`

  Tells the system whether the domain is active, inactive, or deleted. The default is active. To temporarily disable the domain, indicate `inactive` in the LDIF record. To delete the domain, indicate `deleted` in the LDIF record. If this attribute is missing it is implied as `active`.

- ```
  inetAuthorizedServices: imap
  inetAuthorizedServices: pop3
  inetAuthorizedServices: imaps
  inetAuthorizedServices: pop3s
  inetAuthorizedServices: smtp
  inetAuthorizedServices: sunw_webaccess
  ```

  These lines indicate the list of internet services which are authorized within this domain. The services that you can set permissions for include the following:

  - imap - IMAP-based protocol services
  - imaps - secure IMAP-based protocol services
  - pop3 - POP-based message access
  - pop3s - secure POP-based message access
  - smtp - access to SMTP server for authorized message submission.
  - smtps - access to secure SMTP server for message submission.

  If this attribute is missing, it is the same as specifying all services.
  ```

- `owner: cn=Mike Marola (Delegated Admin) (deladmin),ou=People,dc=stream,dc=com,o=internet`

  This is a multi-value attribute specifying the distinguished name of the Delegated Administrator(s). The Delegated Administrator has the privileges to add, modify, delete, and search for group or user entries in the hosted domain. If this attribute is included in the LDIF record, then a corresponding user entry must be included in the container `ou=people`. If the site is not going to support Delegated Management, this value may be excluded.

- `dnsDomainName: stream.com`

  Indicates DNS domain name associated with this node in the DIT.

simsDomain Attributes

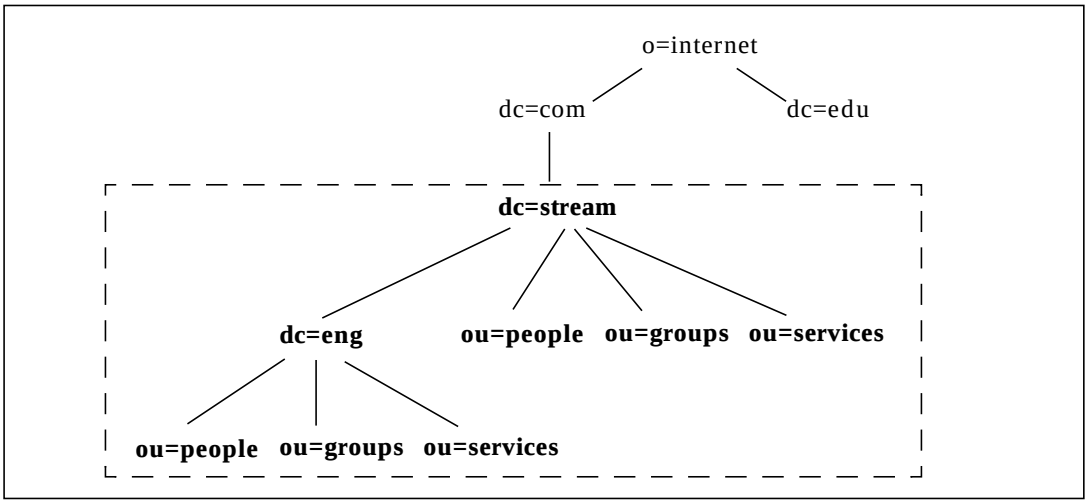
- `simsDomainVersion: 1.0`

  This required attribute indicates which version of the object class is being used in the domain.

- `simsRecursive: 0`

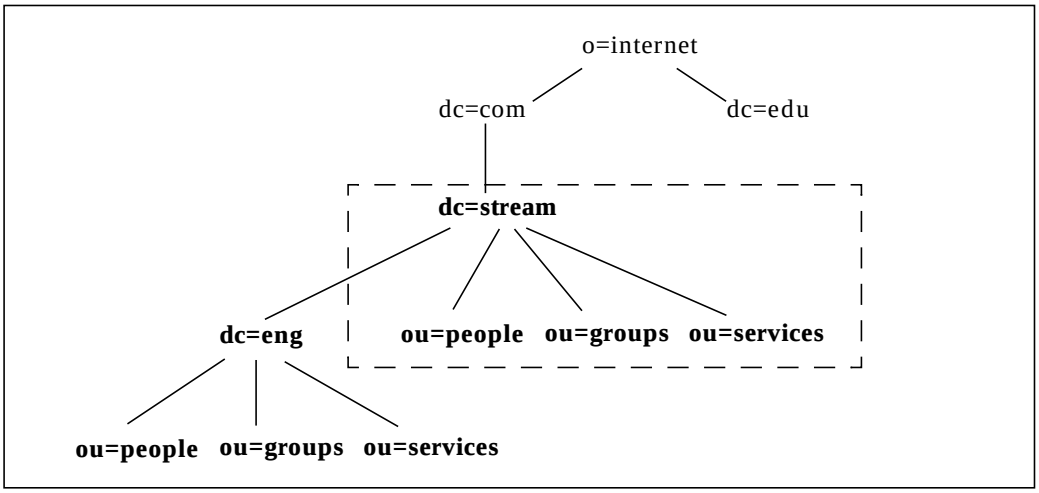
  `simsRecursive` indicates the bounds of the namespace. If this attribute is missing, the default value is 0, however, we recommend adding this attribute for clarity and comprehension. The values are 1 and 0. 1 means that the domain and any sub-domains under it are treated as one flat namespace. For instance, in the following tree, entering 1 for `simsRecursive` for the hosted domain `stream`

means that `stream`, the sub-domain `eng`, and all three containers for `people`, `groups`, and `services` are within one flat namespace in the directory. The dashed boxes represent the namespace within the tree.



If you choose 0 for `simsRecursive`, every level of domain will be its own namespace and a search on the namespace won't look recursively down the tree. For example in the diagram below, choosing 0 for `simsRecursive` means that

only users in that domain are in the scope of the domain's namespace and all sub-domains are separate namespaces whose bounds are determined by the `simsRecursive` flag of that domain.



`simsRecursive=0` is the recommended value. If the `value=0`, the IMTA generates a routing entry for this domain. If you set a value to 1 for a particular site, you must add rewrite rules to the IMTA configuration files. In other words, if the value is set to 0, the system automatically generates the appropriate rewrite rule to route messages to this domain. If the value is 1, the system cannot automatically generate a rewrite rule since an incoming message can go to the domain or any of its subdomains. Thus, you will need to create specific rewrite rules for the domain and each of its subdomains.

- `rfc822Postmaster: deladmin@stream.com`

  Address of the postmaster. Mail addressed to `postmaster@stream.com` will be sent to the address specified in this attribute.

- `mailHosts: route66.stream.com`

  `mailHosts` is a list of fully qualified hostnames of mail servers that have routing responsibility for this domain. You cannot specify a mailserver in a domain entry if that mailserver is specified in a parent domain.

- `preferredMailhost: route66.stream.com`

  Fully qualified hostname of the preferred mail server for this hosted domain. When the delegated administrator adds a new user/group, the new user/group is assigned this value for their mailhost. Service providers can use this attribute to control where new users and groups are created. SIMS provisioning tools (the Delegated Admin Console and the SIMS Administration command line interface) use this value when creating users and groups.

- `domainDiskQuota: 10G`

  Disk quota in bytes for this domain. Disk usage for all users in this hosted domain should not exceed this value. The default unit can be overridden by using one of the following tags:

  `<size>K` - size is specified in kilobytes

  `<size>M` - size is specified in megabytes

  `<size>G` - size is specified in gigabytes

  `<size>T` - size is specified in terabytes

- `maxMailboxes: 10`

  The maximum number of mailboxes allowed in the domain.

- `maxDistributionLists: 2`

  The maximum number of distribution lists allowed in the domain.

- `maxEntries: 12`

  This indicates the number of directory entries allowed for the domain.

Creating a Domain Alias Entries

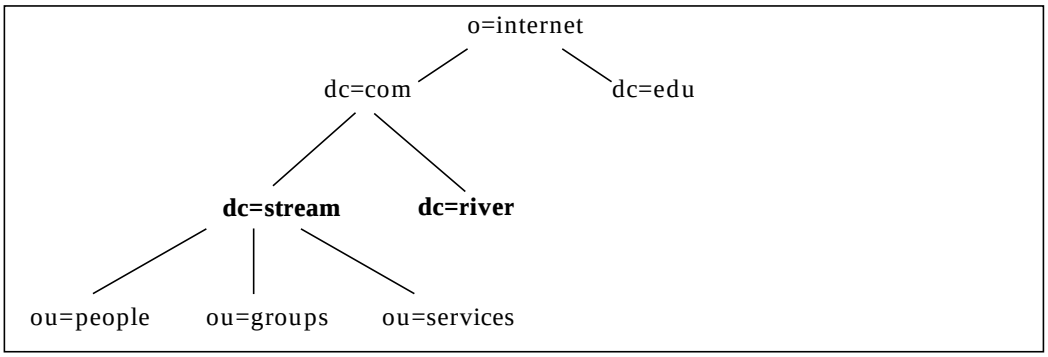


FIGURE 2-4 Third-level Domain stream.com.

A domain alias is an entry that points to another domain. Domain aliasing allows hosted domains to have several domain names. One of these domain name is the official domain name used amongst other things to create internal addresses. This special name is referred to as official domain name.

The LDIF code for creating a hosted domain is shown in CODE EXAMPLE 2-3.

CODE EXAMPLE 2-4 LDIF Record for Creating a Hosted Domain.

```
dn: dc=river,dc=com,o=internet
objectClass: alias
objectClass: aliasobject
aliasedObjectName: dc=stream,dc=com,o=Internet
dc: river
```

Create the Required Containers for Hosted Domains

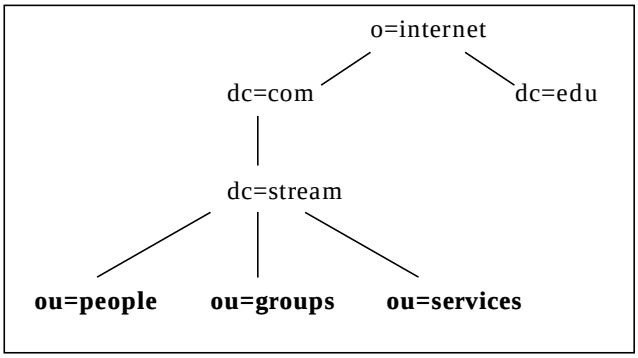


FIGURE 2-5 Third-level Domain stream.com.

Every domain that contains users or groups must have three `organizationalUnit` containers:

- `ou=people` - container for user entries.
- `ou=groups` - container for group entries.
- `ou=services` - container for service entries.

The following example shows the LDIF records for creating these required container entries in the hosted domain stream.

CODE EXAMPLE 2-5 LDIF Records for Hosted Domain Containers.

```
dn: ou=People,dc=stream,dc=com,o=internet
objectClass: organizationalUnit
ou: People

dn: ou=Groups,dc=stream,dc=com,o=internet
objectClass: organizationalUnit
ou: Groups
```

CODE EXAMPLE 2-5 LDIF Records for Hosted Domain Containers. *(Continued)*

```
dn: ou=Services,dc=stream,dc=com,o=internet
objectClass: organizationalUnit
ou: Services
```

Hosted Domain Container Attributes

This section provides brief descriptions of the hosted domain node attributes. For more complete descriptions of the attributes refer to the SIMS schema in the *SIMS Reference Manual*.

- `dn: ou=People,dc=stream,dc=com,o=internet`
  `dn: ou=Groups,dc=stream,dc=com,o=internet`
  `dn: ou=Services,dc=stream,dc=com,o=internet`

  These are the distinguished names for containers required by all hosted domains. `ou=People` contains all the user entries for the hosted domain. `ou=Groups` contains all the group entries for the hosted domain. `ou=Services` contains entries for service objects.

- `objectClass: organizationalUnit`

  The `organizationalUnit` object class is used to create the container entries of the primary DIT in our example.

- `ou: People`
  `ou: Groups`
  `ou: Services`

  These are the three required organizationalUnit entries.

Domain Tasks

This section describes how to implement common domain tasks. The entire LDIF record is given for each task, however, most tasks require only adding one or more attributes to an existing domain. If you are going to modify a record, use only the lines in italics. For example, to do the task described in the following section using `ldapmodify`, you would do:

```
# ./ldapmodify -D "<SIMS Admin DN>" -w <passwd> -f change.ldif
```

where the contents of `change.ldif` is:

```
dn: dc=stream,dc=com,o=internet
changetype: modify
add: owner
owner: cn=Mike Marola (Delegated Admin),ou=People,dc=stream,dc=com,o=internet
owner: cn= Bill Komash (Delegated Bill),ou=People,dc=stream,dc=com,o=internet
```

▼ Creating or Removing Delegated Administrators

To create or remove delegated administrator privileges, use the `owner` attribute. These must be valid existing user entries in the domain to be delegated. The LDIF example below, specifies two delegated administrators. Note that a line may be continued by inserting a single space at the beginning of the next line.

CODE EXAMPLE 2-6 LDIF Record for Creating Delegated Administrators.

```
dn: dc=stream,dc=com,o=internet
objectClass: domain
objectClass: inetDomain
objectClass: simsDomain
description: DC node for stream.com hosted domain
```

CODE EXAMPLE 2-6 LDIF Record for Creating Delegated Administrators. *(Continued)*

```
dc: stream
inetTreeStyle: DC
inetDomainStatus: active
inetAuthorizedServices: imap
inetAuthorizedServices: pop3
inetAuthorizedServices: imaps
inetAuthorizedServices: pop3s
inetAuthorizedServices: smtp
inetAuthorizedServices: sunw_webaccess
owner: cn=Mike Marola (Delegated Admin),ou=People,dc=stream,dc=com,o=internet
owner: cn= Bill Komash (Delegated Bill),ou=People,dc=stream,dc=com,o=internet
simsRecursive: 0
simsDomainVersion: 1.0
dnsDomainName: stream.com
rfc822Postmaster: deladmin@stream.com
mailHosts: route66.stream.com
preferredMailhost: route66.stream.com
domainDiskQuota: 10G
maxMailboxes: 10
maxDistributionLists: 2
maxEntries: 12
```

▼ Assigning or Removing RFC822 Postmasters

To assign or remove RFC822 postmasters, add or remove user entries to the `rfc822Postmaster` attribute. These must be valid email addresses. In the example below, specifies domain postmasters.

CODE EXAMPLE 2-7 LDIF Record for Assigning RFC822 Postmasters.

```
dn: dc=stream,dc=com,o=internet
objectClass: domain
objectClass: inetDomain
objectClass: simsDomain
description: DC node for stream.com hosted domain
dc: stream
inetTreeStyle: DC
inetDomainStatus: active
inetAuthorizedServices: imap
inetAuthorizedServices: pop3
inetAuthorizedServices: imaps
inetAuthorizedServices: pop3s
inetAuthorizedServices: smtp
```

CODE EXAMPLE 2-7 LDIF Record for Assigning RFC822 Postmasters. *(Continued)*

```
inetAuthorizedServices: sunw_webaccess
owner: cn=Mike Marola (Delegated Admin),ou=People,dc=stream,dc=com,o=internet
simsRecursive: 0
simsDomainVersion: 1.0
dnsDomainName: stream.com
rfc822Postmaster: deladmin@stream.com
rfc822Postmaster: billkoma@stream.com
mailHosts: route66.stream.com
preferredMailhost: route66.stream.com
domainDiskQuota: 10G
maxMailboxes: 10
maxDistributionLists: 2
maxEntries: 12
```

▼ Changing the Preferred Mailhost

To change the preferred mailhost, assign the fully qualified domain name of the mailhost to `preferredMailhost`. In the example below, the `preferredMailhost` is highlighted.

CODE EXAMPLE 2-8 LDIF Record for Changing the Preferred Mailhost.

```
dn: dc=stream,dc=com,o=internet
objectClass: domain
objectClass: inetDomain
objectClass: simsDomain
description: DC node for stream.com hosted domain
dc: stream
inetTreeStyle: DC
inetDomainStatus: active
inetAuthorizedServices: imap
inetAuthorizedServices: pop3
inetAuthorizedServices: imaps
inetAuthorizedServices: pop3s
inetAuthorizedServices: smtp
inetAuthorizedServices: sunw_webaccess
owner: cn=Mike Marola (Delegated Admin),ou=People,dc=stream,dc=com,o=internet
simsRecursive: 0
simsDomainVersion: 1.0
dnsDomainName: stream.com
rfc822Postmaster: deladmin@stream.com
mailHosts: route66.stream.com
preferredMailhost: route66.stream.com
```

CODE EXAMPLE 2-8 LDIF Record for Changing the Preferred Mailhost. *(Continued)*

```
domainDiskQuota: 10G
maxMailboxes: 10
maxDistributionLists: 2
maxEntries: 12
```

▼ Adding a New Server to the SIMS System

To add a mail server, add its fully qualified domain name to `mailHosts` as shown below.

CODE EXAMPLE 2-9 LDIF Record for Adding a New Server to the System.

```
dn: dc=stream,dc=com,o=internet
objectClass: domain
objectClass: inetDomain
objectClass: simsDomain
description: DC node for stream.com hosted domain
dc: stream
inetTreeStyle: DC
inetDomainStatus: active
inetAuthorizedServices: imap
inetAuthorizedServices: pop3
inetAuthorizedServices: imaps
inetAuthorizedServices: pop3s
inetAuthorizedServices: smtp
inetAuthorizedServices: sunw_webaccess
owner: cn=Mike Marola (Delegated Admin),ou=People,dc=stream,dc=com,o=internet
simsRecursive: 0
simsDomainVersion: 1.0
dnsDomainName: stream.com
rfc822Postmaster: deladmin@stream.com
mailHosts: route66.stream.com
mailHosts: bowser.isp.net
preferredMailhost: route66.stream.com
domainDiskQuota: 10G
maxMailboxes: 10
maxDistributionLists: 2
maxEntries: 12
```

▼ Adding or Removing Authorized Services

To add or remove authorized services for a domain, add or remove the desired internet services to the `inetAuthorizedServices` attribute. For example, if you wanted to remove IMAP support, then you would remove the two lines highlighted below. This results in IMAP and IMAPS services being disallowed for all users in the domain, even if users have these services listed in their user entries.

CODE EXAMPLE 2-10 LDIF Record for Adding Authorized Services.

```
dn: dc=stream,dc=com,o=internet
objectClass: domain
objectClass: inetDomain
objectClass: simsDomain
description: DC node for stream.com hosted domain
dc: stream
inetTreeStyle: DC
inetDomainStatus: active
inetAuthorizedServices: imap
inetAuthorizedServices: pop3
inetAuthorizedServices: imaps
inetAuthorizedServices: pop3s
inetAuthorizedServices: smtp
inetAuthorizedServices: sunw_webaccess
owner: cn=Mike Marola (Delegated Admin),ou=People,dc=stream,dc=com,o=internet
simsRecursive: 0
simsDomainVersion: 1.0
dnsDomainName: stream.com
rfc822Postmaster: deladmin@stream.com
mailHosts: route66.stream.com
preferredMailhost: route66.stream.com
domainDiskQuota: 10G
maxMailboxes: 10
maxDistributionLists: 2
maxEntries: 12
```

Note – The set of services a user is permitted is derived from the intersection of services user entry and the services specified in the domain entry.

▼ Setting the Domain Quota

The domain quota is the maximum amount of storage space that all the mailboxes of all the users in a particular domain can use. SIMS does not do strict domain quota enforcement. That is, the quota reporting tool uses this value in its report, but the system does not reject messages when `domainQuota` is exceeded.

Set `domainDiskQuota` to the amount of data storage allocated to a domain.

CODE EXAMPLE 2-11 LDIF Record for Changing the Domain Storage Quota.

```
dn: dc=stream,dc=com,o=internet
objectClass: domain
objectClass: inetDomain
objectClass: simsDomain
description: DC node for stream.com hosted domain
dc: stream
inetTreeStyle: DC
inetDomainStatus: active
inetAuthorizedServices: imap
inetAuthorizedServices: pop3
inetAuthorizedServices: imaps
inetAuthorizedServices: pop3s
inetAuthorizedServices: smtp
inetAuthorizedServices: sunw_webaccess
owner: cn=Mike Marola (Delegated Admin),ou=People,dc=stream,dc=com,o=internet
simsRecursive: 0
simsDomainVersion: 1.0
dnsDomainName: stream.com
rfc822Postmaster: deladmin@stream.com
mailHosts: route66.stream.com
preferredMailhost: route66.stream.com
domainDiskQuota: 90000M
maxMailboxes: 10
maxDistributionLists: 2
maxEntries: 12
```

CHAPTER 3

Creating Mail Users

This section discusses creating mail user entries. Also discussed are various tasks for modifying mail user entries. For each task, the attribute will be shown, and an example LDIF record will be shown to illustrate the syntax.

Creating User Entries

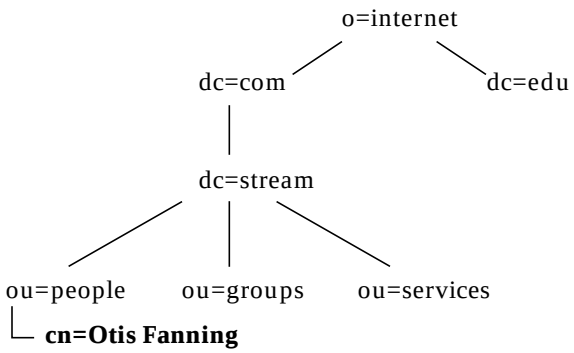


FIGURE 3-1 Creating a User.

E-mail users are represented in the ou=people container of the DIT. Information about users are defined in a set of attributes in an LDIF record within that container. And example is shown below.

CODE EXAMPLE 3-1 LDIF Record for Creating a User.

```
dn: cn=Otis Fanning,ou=People,dc=stream,dc=com,o=internet
objectClass: inetOrgPerson
objectClass: inetSubscriber
objectClass: inetMailRouting
objectClass: inetMailUser
cn: Otis Fanning
sn: Fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@stream.com
uid: fanning
userPassword: secret
inetAuthorizedServices: imap
inetAuthorizedServices: pop3
inetAuthorizedServices: imaps
inetAuthorizedServices: pop3s
inetAuthorizedServices: sunw_webaccess
inetAuthorizedServices: sunw_calendar
```

CODE EXAMPLE 3-1 LDIF Record for Creating a User. *(Continued)*

```
inetMailUserVersion: 1.0
inetSubscriberStatus: active
rfc822MailAlias: ofanning@stream.com
mailDeliveryOption: mailbox
mailHost: route66.stream.com
mailFolderMap: Sun-MS
dataSource: @(#)mkdirdata.sh 1.10 02/19/99
mailQuota: -1
```

User Attributes

- `dn: cn=Otis Fanning,ou=People,dc=stream,dc=com,o=internet`

  The distinguished name of the user Otis Fanning.

- `objectClass: inetOrgPerson`
  `objectClass: inetSubscriber`
  `objectClass: inetMailRouting`
  `objectClass: inetMailUser`

  These are the object classes required in the LDAP directory entries for all users who will receive, send, or read internet email. `inetOrgPerson` is the structural object class.

  `inetOrgPerson` inherits from `organizationalPerson` and `person`. It provides attributes for describing a person belonging to an organization and who interacts with the internet.

  `inetSubscriber` object class provides information needed to manage a subscriber of one or more internet services (for example, sending of email, retrieving received email, calendar access, etc.).

  `inetMailRouting` object class contains the attributes necessary for an MTA to make routing decisions for message recipients. (A distinction is being drawn between an MTA that relays the message and the MTA that is responsible for delivering the message in a users mailbox.) This class is required for entries describing either email users (inetMailUser) or email groups (inetMailGroup).

  `inetMailUser` object class specifies attributes used to store and retrieve information related to storage of incoming email and sending of outbound email.

inetOrgPerson Attributes

- `cn: Otis Fanning`

cn (commonname) is the user's full name and is inherited from the person object class. The cn attribute is required to be unique within a domain, however, the same cn can exist in a different domain.

- `sn: Fanning`

  sn (surname) is the user's last or family name and is inherited from the person object class.

- `initials: OTF`

  initials contains the initials of some or all of an individuals names. In this example OTF stands for Otis Tiberus Fanning.

- `givenName: Otis`

  givenName is the first name of the user.

- `mail: otis.fanning@stream.com`

  mail The user's advertised e-mail address (RFC 822 format).

- `uid: fanning`

  uid is the subscriber's login id used to log in to a computer system. This attribute must be unique within the naming context associated with containing DNS domain.

- `userPassword: {sunds}...`

  userPassword is the encrypted string representing the user's password and is inherited from the person object class.

inetSubscriber Attributes

Note – uid is an attribute for both inetOrgPerson and inetSubscriber. The attribute is exactly the same.

- `inetAuthorizedServices: imap`
  `inetAuthorizedServices: pop3`
  `inetAuthorizedServices: imaps`
  `inetAuthorizedServices: pop3s`
  `inetAuthorizedServices: sunw_webaccess`
  `inetAuthorizedServices: sunw_calendar`

  `inetAuthorizedServices` indicates the list of internet services which the user is authorized to access within the domain. If this attribute is not set, then the user has permission to use all supported services, but still subject to service permitted for the domain.

Note – We recommend adding a directory access control rule to the system to restrict the user's ability to modify this attribute. Refer to the directory service documentation for details on how to do this.

- `inetSubscriberStatus: active`

  This attribute specifies the global access status of the user's account. The intent of this attribute is to allow the ISP to suspend and reactivate the user's account. The values are:

  active - the user's account is active and the user may use all accesses granted by `inetAuthorizedServices`.

  inactive - the user's account is inactive and the user may not use any services granted by inetAuthorizedServices. Service requests for a user marked as inactive must return transient failures. This will allow sending MTAs to retry message delivery for suspended users.

  deleted - the user's account is marked for deletion. The account may remain as marked for deletion within the directory for a period of time, unless there is a purge operation for deleted users. Service requests for a deleted user are returned as permanent failures.

inetMailRouting Attributes

- `mail: otis.fanning@stream.com`

  `mail` is the user's advertised e-mail address (RFC 822 format).

Note – `mail` is an attribute for both `inetMailRouting` and `inetOrgPerson`. The attribute is exactly the same.

- `mailHost: route66.stream.com`

  `mailHost` is a fully-qualified hostname of the mail server where the user's Inbox is located.

- `rfc822MailAlias: ofanning@stream.com`

  `rfc822MailAlias` stores alternate e-mail addresses for the user. Mail to any of the `rfc822MailAlias` will be delivered to the user associated with that entry. The value in this attribute must be unique for all `mail` and `rfc822MailAlias` attributes in a domain.

inetMailUser Attributes

- `inetMailUserVersion: 1.0`

  `inetMailUserVersion` indicates the version tag of this object class. This allows LDAP clients supporting internet e-mail services to retrieve LDAP objects supporting a particular version of schema. The starting value for a version tag is 1.0. Changes to this object class must also increment the value in the `inetMailUserVersion` attribute.

- `dataSource: @(#)mkdirdata.sh 1.10 02/19/99`

  `dataSource` is a free form text string that describes the source or identifier of the provisioning tool.

- `mailDeliveryOption: mailbox`

  This attribute specifies one or more delivery options for inbound messages to a designated recipient. The inbound messages can be delivered into multiple message stores, however, the message access server can read messages from only one message store. This message store is specified by the `mailFolderMap` attribute. The IMTA reads `mailDeliveryOption` to determine message delivery for all messages inbound to a particular user. The values for this attribute can be:

  `mailbox` - Deliver mail to a vendor specific / high performance Message Store mailbox. The `mailFolderMap` attribute specifies the mail store from which a Message Access agent would expect to retrieve delivered mail. For example, in SIMS, provisioning a user to read messages from the Sun Message Store requires

setting the mailDeliveryOption to mailbox, and the associated mailFolderMap attribute to Sun-MS. (Please refer to the description of the mailFolderMap attribute below.)

native - this option applies only to the inetMailUser object class. It specifies delivery of mail to a local sendmail-style file system mailbox (also known as the /var/mail box). If mailDeliveryOption is set to native, then the mailFolderMap attribute must be set to UNIX V7 in order for the user to read messages from the /var/mail using the Sun internet email product's message access services. Please refer to the mailFolderMap below and mailMessageStore ("Setting Up Additional Delivery Files" on page 42)attribute definitions.

autoreply - Deliver mail to an auto-reply facility. When this value is set, the behavior of the autoreply feature of the MTA will be controlled by the following inetMailUser attributes:

mailAutoReplyStartDate, mailAutoReplyExpirationDate, mailAutoReplyTimeout, mailAutoReplySubject, mailAutoReplyText, and mailAutoReplyTextInternal. See "Setting the Message Autoreply" on page 37.

program - delivers mail to a program. For security reasons, the value of this attribute is restricted to authorized programs. This list is configured and maintained by the SIMS Administrator. Refer to the section on Alternative Delivery Programs in the *SIMS System Administration Guide*

forward - forward mail to another RFC 822 compliant address as specified by the attribute mailForwardingAddress

file - appends incoming mail to a file. The mailDeliveryFile attribute must point to a valid file for this option to work. Refer to the schema chapter in the *SIMS Reference Manual*.

- mailFolderMap: Sun-MS

mailFolderMap is the message store for a user's mail folders. Message access servers (IMAP server, POP server, etc.) use this attribute to determine a user's primary mailbox. An IMTA may deliver a message into multiple locations, and message access servers have to be told the default mailbox of the user. Supported values in the unbundled Sun internet email product are:

UNIX V7 - sendmail-style message store

Sun-MS - Sun Message Store, which is accessed through IMAP or POP protocols

- `mailQuota: -1`

  This indicates the maximum size (in bytes) of the user's message store, including the Inbox and any other mailboxes. The values can be -1 or -2 or a specific storage size. A value of -1 denotes no limit on the size of messages in the user's Inbox and folders. A value of -2 sets the mail quota to the system default as specified in the message stores configuration file (see the `ims.cnf` man page).

  The default unit can be overridden by using one of the following tags:

  `<size>K` - size is specified in kilobytes

  `<size>M` - size is specified in megabytes

  `<size>G` - size is specified in gigabytes

  `<size>T` - size is specified in terabytes

Mail User Tasks

This section describes how to implement common tasks on user entries. The entire LDIF record is given for each task, however, most tasks require only adding or modifying one or more attributes to an existing user. Instead of using the entire record, use only the lines in italics. For example, to do the task described in the following section using `ldapmodify`, you would do:

```
# ./ldapmodify -D "<SIMS Admin DN>" -w <passwd> -f change.ldif
```

where the contents of `change.ldif` is:

```
dn: cn=Eileen Fanning,ou=People,dc=stream,dc=com,o=internet
changetype: modify
add: inetSubscriberStatus
inetSubscriberStatus: active
```

Activating and Deactivating Users

To activate or deactivate a user in the hosted domain, you must define the `inetSubscriberStatus` attribute in the user's entry. This attribute specifies the status of the user's account. The values are:

- `active` - the subscriber account is active and the subscriber may use all accesses granted by `inetAuthorizedServices`.
- `inactive` - the subscriber account is inactive. The subscriber may not use any services granted by `inetAuthorizedServices`. An inactive account may be activated by setting the attribute to `active`. Service requests for an inactive user are returned as transient failures.
- `deleted` - the subscriber account is marked for deletion. The account may remain as marked for deletion within the directory for a period of time, unless there is a purge operation for deleted users. Service requests for a deleted user are returned as permanent failures.

An example of an LDIF entry that changes the user's subscriber status follows:

CODE EXAMPLE 3-2 LDIF Record for a User's Subscriber Status.

```
dn: cn=Otis Fanning,ou=People,dc=stream,dc=com,o=internet
objectClass: inetOrgPerson
objectClass: inetSubscriber
objectClass: inetMailRouting
objectClass: inetMailUser
cn: Otis Fanning
sn: Fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@stream.com
uid: fanning
userPassword: secret
inetAuthorizedServices: imap
inetAuthorizedServices: pop3
inetAuthorizedServices: imaps
inetAuthorizedServices: pop3s
```

CODE EXAMPLE 3-2 LDIF Record for a User's Subscriber Status. *(Continued)*

```
inetAuthorizedServices: sunw_webaccess
inetAuthorizedServices: sunw_calendar
inetMailUserVersion: 1.0
inetSubscriberStatus: inactive
rfc822MailAlias: ofanning@stream.com
mailDeliveryOption: mailbox
mailHost: route66.stream.com
mailFolderMap: Sun-MS
dataSource: @(#)mkdirdata.sh 1.10 02/19/99
mailQuota: -1
```

Changing a Password

You can change the user's password in the LDIF record by changing the `userPassword` attribute.

Note – You cannot change the password of the SIMS "super" Administrator (that is, the password of the SIMS Administrator as defined by the `adminBindDN` attribute in the `/etc/opt/SUNWmail/sims.cnf` file) using the LDAP commands.

CODE EXAMPLE 3-3 LDIF Record for Changing a User's Password.

```
dn: cn=Otis Fanning,ou=People,dc=stream,dc=com,o=internet
objectClass: inetOrgPerson
objectClass: inetSubscriber
objectClass: inetMailRouting
objectClass: inetMailUser
cn: Otis Fanning
sn: Fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@stream.com
uid: fanning
userPassword: 2bornot2b
inetAuthorizedServices: imap
inetAuthorizedServices: pop3
inetAuthorizedServices: imaps
inetAuthorizedServices: pop3s
inetAuthorizedServices: sunw_webaccess
inetAuthorizedServices: sunw_calendar
inetMailUserVersion: 1.0
```

CODE EXAMPLE 3-3 LDIF Record for Changing a User's Password. *(Continued)*

```
inetSubscriberStatus: inactive
rfc822MailAlias: ofanning@stream.com
mailDeliveryOption: mailbox
mailHost: route66.stream.com
mailFolderMap: Sun-MS
dataSource: @(#)mkdirdata.sh 1.10 02/19/99
mailQuota: -1
```

Setting the Message Autoreply

You can configure the SIMS server to produce an automatic reply to any incoming email to a user's account. This is handy for when the user goes on vacation or is away from his system for a period of time. The required attributes are part of the `inetMailUser` object class and consist of the following: `mailAutoReplyStartDate`, `mailAutoReplyExpirationDate`, `mailDeliveryOption`, `mailAutoReplySubject`, and `mailAutoReplyTextInternal`.

The `mailDeliveryOption` attribute activates the autoreply feature when it is set to `autoreply`.

`mailAutoReplyStartDate` specifies when the IMTA should enable the automatic replies to incoming mail for the user. The format for setting this attribute is the 4-digit year, 2-digit month, and 2-digit day, 2-digit hour, 2-digit minute, 2-digit second and terminating with a Z for Zulu time. For example to set it to start at 4PM on June 21, 1999, the value of start date is 19990622000000Z

`mailAutoReplyExpirationDate` specifies when the IMTA should disable the automatic replies to incoming mail for the user. The format for setting this attribute is the same as `mailAutoReplyStartDate`.

`mailAutoReplyTimeout` defines the duration (in hours) between successive auto-replies to the incoming message from a specific sender. If you do not specify a timeout in the LDIF record, the timeout will be automatically set for seven days. This is used so that a sender doesn't get an autoreply for every email they send. They will only get the autoreply message as often as specified in this attribute.

`mailAutoReplySubject` specifies the subject line of the autoreply message. If the attribute specification contains the token `$SUBJECT`, then the token is replaced by the subject line of the inbound message.

`mailAutoReplyText` defines the body text of the autoreply message. If the attribute contains `$SUBJECT` or `$BODY` in the LDIF entry, these tokens are replaced by the subject or body of the inbound message. Use `$` as a line separator.

`mailAutoReplyTextInternal` defines the body of autoreply messages for internal auto-replies. Senders within the same domain will receive this message. If the attribute contains `$SUBJECT` or `$BODY` in the LDIF entry, these tokens are replaced by the subject or body of the inbound message. Use `$` as a line separator.

An example of the LDIF entry to enable the message autoreply for a users is below. Note that the `mailAutoReplyText` attribute-value pair spans several lines and that each line after the first must start with a blank space. (If you are reading this on-line, this space may not be displayed in your browser.)

CODE EXAMPLE 3-4 LDIF Record for Activating a User's Autoreply Feature.

```
dn: cn=Otis Fanning,ou=People,dc=stream,dc=com,o=internet
objectClass: inetOrgPerson
objectClass: inetSubscriber
objectClass: inetMailRouting
objectClass: inetMailUser
cn: Otis Fanning
sn: Fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@stream.com
uid: fanning
userPassword: secret
inetAuthorizedServices: imap
inetAuthorizedServices: pop3
inetAuthorizedServices: imaps
inetAuthorizedServices: pop3s
inetAuthorizedServices: sunw_webaccess
inetAuthorizedServices: sunw_calendar
inetMailUserVersion: 1.0
inetSubscriberStatus: active
rfc822MailAlias: ofanning@stream.com
mailDeliveryOption: autoreply
mailAutoReplyStartDate: 19990622000000Z
mailAutoReplyExpirationDate: 19990629000000Z
mailAutoReplyTimeout: 6
mailAutoReplySubject: Otis is on vacation
mailAutoReplyText: Otis is out on vacation and will return
 1/2/2000. $ If you need immediate assistance on $SUBJECT than
 please contact Mike Marola at marola@marola.com.
mailAutoReplyTextInternal: Otis out until 1/2/2000.
mailHost: route66.stream.com
mailFolderMap: Sun-MS
dataSource: @(#)mkdirdata.sh 1.10 02/19/99
mailQuota: -1
```

Assigning and Modifying Services for Users

Assign and modify services for users by adding or removing service arguments to the value of the `inetAuthorizedServices` attribute.

CODE EXAMPLE 3-5 LDIF Record for Changing a User's Services.

```
dn: cn=Otis Fanning,ou=People,dc=stream,dc=com,o=internet
objectClass: inetSubscriber
objectClass: inetMailRouting
objectClass: inetMailUser
cn: Otis Fanning
sn: Fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@stream.com
uid: fanning
userPassword: secret
inetAuthorizedServices: imap
inetAuthorizedServices: pop3
inetAuthorizedServices: imaps
inetAuthorizedServices: pop3s
inetAuthorizedServices: sunw_webaccess
inetAuthorizedServices: sunw_calendar
inetMailUserVersion: 1.0
inetSubscriberStatus: active
rfc822MailAlias: ofanning@stream.com
mailDeliveryOption: mailbox
mailHost: route66.stream.com
mailFolderMap: Sun-MS
dataSource: @(#)mkdirdata.sh 1.10 02/19/99
mailQuota: -1
```

Defining New User Aliases

You can assign additional values for the `rfc822MailAlias` to define aliases for a user. The new values must be unique for both users and groups in the domain namespace.

CODE EXAMPLE 3-6 LDIF Record for Defining New User Aliases.

```
dn: cn=Otis Fanning,ou=People,dc=stream,dc=com,o=internet
objectClass: inetOrgPerson
objectClass: inetSubscriber
objectClass: inetMailRouting
objectClass: inetMailUser
cn: Otis Fanning
sn: Fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@stream.com
rfc822MailAlias: Otis.Fanning@us.stream.com
uid: fanning
userPassword: secret
inetAuthorizedServices: imap
inetAuthorizedServices: pop3
inetAuthorizedServices: imaps
inetAuthorizedServices: pop3s
inetAuthorizedServices: sunw_webaccess
inetAuthorizedServices: sunw_calendar
inetMailUserVersion: 1.0
inetSubscriberStatus: active
rfc822MailAlias: Otis.Fanning@stream.com
mailDeliveryOption: mailbox
mailHost: route66.stream.com
mailFolderMap: Sun-MS
dataSource: @(#)mkdirdata.sh 1.10 02/19/99
mailQuota: -1
```

Moving a User From One Server to Another

You can change the user's mail storage server by changing the server assigned to the `mailHost` attribute.

CODE EXAMPLE 3-7 LDIF Record for Moving a User from One Server to Another.

```
dn: cn=Otis Fanning,ou=People,dc=stream,dc=com,o=internet
objectClass: inetOrgPerson
objectClass: inetSubscriber
objectClass: inetMailRouting
objectClass: inetMailUser
cn: Otis Fanning
sn: Fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@stream.com
uid: fanning
userPassword: secret
inetAuthorizedServices: imap
inetAuthorizedServices: pop3
inetAuthorizedServices: imaps
inetAuthorizedServices: pop3s
inetAuthorizedServices: sunw_webaccess
inetAuthorizedServices: sunw_calendar
inetMailUserVersion: 1.0
inetSubscriberStatus: active
rfc822MailAlias: ofanning@stream.com
mailDeliveryOption: mailbox
mailHost: antelope.stream.com
mailFolderMap: Sun-MS
dataSource: @(#)mkdirdata.sh 1.10 02/19/99
mailQuota: -1
```

Activating the Forwarding Address Feature

Set the `mailDeliveryOption` attribute to `forward` to activate the forwarding feature. Set the forwarding address by assigning a valid email address to the `mailForwardingAddress` attribute. If the user wishes to continue receiving mail on his default server and also forward the mail, then set the `mailDeliveryOption` attribute to `mailbox` as well as `forward`. The example below demonstrates this.

CODE EXAMPLE 3-8 LDIF Record for Activating Forwarding Addressing.

```
dn: cn=Otis Fanning,ou=People,dc=stream,dc=com,o=internet
objectClass: inetOrgPerson
objectClass: inetSubscriber
```

CODE EXAMPLE 3-8 LDIF Record for Activating Forwarding Addressing. *(Continued)*

```
objectClass: inetMailRouting
objectClass: inetMailUser
cn: Otis Fanning
sn: Fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@stream.com
uid: fanning
userPassword: secret
inetAuthorizedServices: imap
inetAuthorizedServices: pop3
inetAuthorizedServices: imaps
inetAuthorizedServices: pop3s
inetAuthorizedServices: sunw_webaccess
inetAuthorizedServices: sunw_calendar
inetMailUserVersion: 1.0
inetSubscriberStatus: active
rfc822MailAlias: ofanning@stream.com
mailDeliveryOption: forward
mailDeliveryOption: mailbox
mailForwardingAddress: Otis.Fanning@hawaiian-beach.com
mailHost: route66.stream.com
mailFolderMap: Sun-MS
dataSource: @(#)mkdirdata.sh 1.10 02/19/99
mailQuota: -1
```

Setting Up Additional Delivery Files

You can set up the system such that a user's messages are sent to either a specified file (use the `mailDeliveryFile` attribute) or to a local `sendmail`-style file system mailbox (use the `mailMessageStore` attribute).

`mailDeliveryFile` is the fully qualified pathname of a file to which incoming messages are appended. This file must be accessible for writing from the file system on the user's mail host.

`mailMessageStore` is the file system location for a user's Inbox. This attribute only takes effect when a `mailDeliveryOption` is set to `native` (see "`mailDeliveryOption: mailbox`" on page 32). The IMTA will deliver incoming messages to this file. The filesystem location is in the context of the mail host. If this value is missing and the user's `mailDeliveryOption` is set to `native`, then a default of `/var/mail` is used by the server. This attribute specifies only the name of the directory; to derive the full name of the Inbox, the value of the `uid` attribute (see "`uid: fanning`" on page 30) is appended to the directory name.

The file below creates a user whose mail is delivered to `/var/mail/fanning` and to `/home/fanning/Maibox`.

CODE EXAMPLE 3-9 LDIF Record for Setting Up Delivery Files.

```
dn: cn=Otis Fanning,ou=People,dc=stream,dc=com,o=internet
objectClass: inetOrgPerson
objectClass: inetSubscriber
objectClass: inetMailRouting
objectClass: inetMailUser
cn: Otis Fanning
sn: Fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@stream.com
uid: fanning
userPassword: secret
inetAuthorizedServices: imap
inetAuthorizedServices: pop3
inetAuthorizedServices: imaps
inetAuthorizedServices: pop3s
inetAuthorizedServices: sunw_webaccess
inetAuthorizedServices: sunw_calendar
inetMailUserVersion: 1.0
inetSubscriberStatus: active
rfc822MailAlias: ofanning@stream.com
mailDeliveryFile: /home/fanning/Mailbox
mailMessageStore: /var/mail/
mailDeliveryOption: native
mailHost: route66.stream.com
mailFolderMap: Sun-MS
dataSource: @(#)mkdirdata.sh 1.10 02/19/99
mailQuota: -1
```

Setting User Quotas

A user quota is the amount of space allocated to a user for his mailboxes. Set user quotas by specifying the value of the `mailQuota` attribute to the maximum size (in bytes) of a user's message store. Note that this includes the Inbox and all other mailboxes or folders which the user may have in the message store. A value of -1 or a missing value denotes no limit on the cumulative size of messages in a user's Inbox and / or folder collection. A value of -2 implies that the system or domain default is used. The default unit of bytes may be overridden by using one of the tags listed below prefixed by the size:

<size>K - size is specified in kilobytes

<size>M - size is specified in megabytes

<size>G - size is specified in gigabytes

<size>T - size is specified in terabytes

CODE EXAMPLE 3-10 LDIF Record for Setting a User Quota.

```
dn: cn=Otis Fanning,ou=People,dc=stream,dc=com,o=internet
objectClass: inetOrgPerson
objectClass: inetSubscriber
objectClass: inetMailRouting
objectClass: inetMailUser
cn: Otis Fanning
sn: Fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@stream.com
uid: fanning
userPassword: secret
inetAuthorizedServices: imap
inetAuthorizedServices: pop3
inetAuthorizedServices: imaps
inetAuthorizedServices: pop3s
inetAuthorizedServices: sunw_webaccess
inetAuthorizedServices: sunw_calendar
inetMailUserVersion: 1.0
inetSubscriberStatus: active
rfc822MailAlias: ofanning@stream.com
mailDeliveryOption: mailbox
mailHost: route66.stream.com
mailFolderMap: Sun-MS
dataSource: @(#)mkdirdata.sh 1.10 02/19/99
mailQuota: 9M
```

CHAPTER 4

Creating Groups

This section discusses creating distribution lists (also called groups). A distribution list is a collection of users to which mail can be sent with a single email address. Also discussed will be various tasks for creating group entries. For each task, the attribute will be shown, and an LDIF record example will be shown to illustrate the syntax.

Note – Some of the example code samples contain attribute-value pairs that span more than one line. If this is the case, every line after the first must begin with a blank space. This blank space is shown in the hard copy or PostScript files, but it may not show on the html browser.

Creating Group Entries

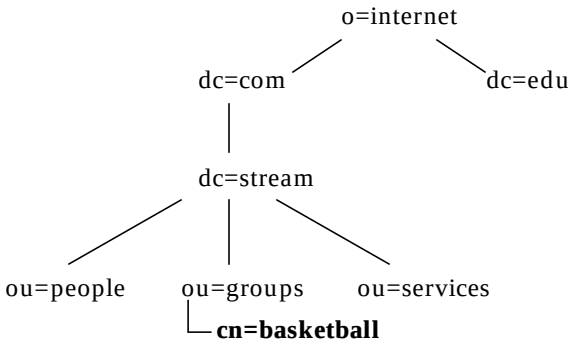


FIGURE 4-1 Creating a Group.

An e-mail distribution list is represented in the directory in the ou=groups container. Information about that group is defined in an entry within that container. An example is shown below.

CODE EXAMPLE 4-1 LDIF Record for Creating a Group.

```
dn: cn=basketball,ou=groups,dc=stream,dc=com,o=internet
objectClass: groupOfUniqueNames
objectClass: inetMailRouting
objectClass: inetMailGroup
cn: basketball
uniqueMember: cn=Kevin Cox (Lighting),ou=people,dc=stream,dc=com,o=internet
rfc822MailMember: camden.miyoko@abalone.com
rfc822MailMember: bryn.yasuko@noodle.net
inetMailGroupVersion: 1.0
inetMailGroupStatus: active
dataSource: Mail Server 4.0
expandable: false
mail: basketball@stream.com
mailHost: buffalo.stream.com
rfc822MailAlias: b-ball_players@stream.com
```

Distribution List Attributes

- `dn: cn=basketball,ou=groups,dc=stream,dc=com,o=internet`

  The distinguished name of the group, `basketball@stream.com`.

- `objectClass: groupOfUniqueNames`
  `objectClass: inetMailRouting`
  `objectClass: inetMailGroup`

  These are the object classes required for all distribution lists.

  `groupOfUniqueNames` object class contains attributes useful for describing a collection of user objects. This object class inherits from `top` and is the structural object class.

  `inetMailRouting` object class contains attributes required for the routing common to all internet email recipients. This class is for entries describing either email users (`inetMailUser`) or groups (`inetMailGroup`).

  `inetMailGroup` object class contains attributes useful for an e-mail distribution list.

groupOfUniqueNames Attributes

- `cn: basketball`

  `cn` (commonname) is the distribution list's common name. There can be more than one cn attribute for a distribution list, however each `cn` must be unique within the domain.

- `uniqueMember: cn=Kevin Cox (Lighting),ou=people,dc=stream,dc=com,o=internet`

  (Required.) This attribute specifies the distinguished names of members of this distribution list.

Note – All distribution lists are required to have at least one unique member.

---

inetMailGroup Attributes

- `rfc822MailMember: camden.kimura@abalone.com`
  `rfc822MailMember: bryn.yasuko@noodle.net`

`rfc822mailmember` stores the e-mail addresses (RFC-822 format) defined for the external members of this list (members without resolvable DNs).

- `inetMailGroupVersion: 1.0`

  `inetmailgroupversion` is a version tag of this object class. This attribute must be set when an entry is created using this object class. The starting version tag is 1.0.

- `inetMailGroupStatus: active`

  `inetmailgroupstatus` specifies the status of a distribution list. The intent of this attribute is to allow the ISP to suspend and reactivate the distribution list. This attribute takes one of three values `active`, `inactive`, `deleted` (marked for deletion). If this attribute is missing, the semantics are the same as if it is `active`.

- `dataSource: Mail Server 4.0`

  `datasource` is free form text entry of the original data source or migration tool for data in the group entry.

- `expandable: false`

  `expandable` specifies whether if the distribution list is expandable or not. If set to true, then someone can read the addresses of the members of the distribution list by using the SMTP command `expn <dl_name>`. If not specified, default is true.

inetMailRouting Attributes

- `mail: basketball@stream.com`

  The group's advertised e-mail address.

- `mailHost: buffalo.stream.com`

  This is the fully qualified hostname of the IMTA where the distribution list is expanded.

- `rfc822MailAlias: b-ball_players@stream.com`

  Stores alternate e-mail aliases (RFC-822 format), if any, defined for the distribution list. Mail to this address will be delivered to the group associated with this entry. The value must be unique for all `mail` and `rfc822MailAlias` attributes in a domain.

Distribution List Tasks

This section describes how to implement common tasks on distribution list entries. The entire LDIF record is shown for most tasks. Usually, however, these tasks require only adding or modifying one or more attributes to an existing distribution list. Instead of using the entire record, use only the lines in italics. For example, to do the task described in the following section using `ldapmodify`, you would do:

```
# ./ldapmodify -D "<SIMS Admin DN>" -w <passwd> -f change.ldif
```

where the contents of `change.ldif` is:

```
dn: cn=basketball,ou=groups,dc=stream,dc=com,o=internet
changetype: modify
add: owner
uniqueMember: cn=Kevin Cox (Lighting),ou=people,dc=stream,dc=com,o=internet
```

Assigning Owners to Groups

Group owners can add or delete members to the distribution list. To change an owner to a group, assign a distinguished name to the `owner` attribute from `objectClass groupOfUniqueNames`. There can be more than one owner for the group, but owners must have valid DNs in the directory where the distribution list is defined. Example code is shown below.

CODE EXAMPLE 4-2 LDIF Record for Creating a Group with an Owner.

```
dn: cn=basketball,ou=groups,dc=stream,dc=com,o=internet
objectClass: groupOfUniqueNames
objectClass: inetMailRouting
objectClass: inetMailGroup
cn: basketball
owner: cn=Kevin Cox (Lighting),ou=people,dc=stream,dc=com,o=internet
uniqueMember: cn=Kevin Cox (Lighting),ou=people,dc=stream,dc=com,o=internet
rfc822MailMember: camden.miyoko@abalone.com
rfc822MailMember: bryn.yasuko@noodle.net
inetMailGroupVersion: 1.0
inetMailGroupStatus: active
dataSource: Mail Server 4.0
expandable: false
mail: basketball@stream.com
mailHost: buffalo.stream.com
rfc822MailAlias: b-ball_players@stream.com
```

Adding Members to a Distribution List

Add internal members (members with resolvable DNs) by assigning their DN to the attribute `uniqueMember`. Add external members by assigning their email address to the attribute `rfc822MailMember`.

CODE EXAMPLE 4-3 LDIF Record for Adding Members to a Group.

```
dn: cn=basketball,ou=groups,dc=stream,dc=com,o=internet
changetype: modify
add: uniqueMember
uniqueMember: cn=Wally Boi,ou=people,dc=stream,dc=com,o=internet
-
add: rfc822MailMember
rfc822MailMember: wilt@abalone.com
```

Making Distribution Lists User Joinable

You can allow members within the directory domain to add or remove themselves from a distribution list by setting the attribute `joinable` to `true`. The values for this task are TRUE and FALSE.

CODE EXAMPLE 4-4 LDIF Record for a Group Joinable.

```
dn: cn=basketball,ou=groups,dc=stream,dc=com,o=internet
objectClass: groupOfUniqueNames
objectClass: inetMailRouting
objectClass: inetMailGroup
cn: basketball
uniqueMember: cn=Kevin Cox (Lighting),ou=people,dc=stream,dc=com,o=internet
rfc822MailMember: camden.miyoko@abalone.com
rfc822MailMember: bryn.yasuko@noodle.net
inetMailGroupVersion: 1.0
inetMailGroupStatus: active
dataSource: Mail Server 4.0
expandable: false
joinable: true
mail: basketball@stream.com
mailHost: buffalo.stream.com
rfc822MailAlias: b-ball_players@stream.com
```

Designating Moderators

A group moderator(s) is someone who first receives a message to the distribution list, reads it, then forwards it to the rest of the members if desired. Any message submitted to the group will go to the moderator instead of the distribution list members. The moderator will then send the message to the distribution list as desired, where it will be delivered to the individual members. Set a valid DN or email address to the attribute `moderator`. Multiple moderators are allowed.

Format of inetMailGroup Attribute Values

There are several `inetMailGroup` attributes—`errorsTo`, `requestsTo`, `moderator`, `authorizedSubmitter`, `unauthorizedSubmitter`—that can contain both RFC-822 mail addresses and DNs of LDAP entries. This is permitted since `inetMailGroup` is

both an LDAP and email entity. When preceded by `ldap:///` the entry is taken as an LDAP entry with the remaining value treated as the distinguished name of the entry. For example:

```
moderator: ldap:///cn=Kevin Cox (White Lightening)
 ,ou=people,dc=stream,dc=com,o=internet
```

When preceded by `mailto:` the entry is interpreted as an RFC-822 address. A missing prefix of `ldap:///` or `mailto:` for the entry is assumed to be an RFC-822 address.

Note that the `moderator` attribute-value pair spans two lines and that the second line must start with a blank space. (If you are reading this on-line, this space may not be displayed in your browser.)

CODE EXAMPLE 4-5 LDIF Record for Creating Group Moderators.

```
dn: cn=basketball,ou=groups,dc=stream,dc=com,o=internet
objectClass: groupOfUniqueNames
objectClass: inetMailRouting
objectClass: inetMailGroup
cn: basketball
moderator: ldap:///cn=Kevin Cox (Lightening)
 ,ou=people,dc=stream,dc=com,o=internet
moderator: camden.miyoko@abalone.com
uniqueMember: cn=Kevin Cox (Lightening),ou=people,dc=stream,dc=com,o=internet
rfc822MailMember: camden.miyoko@abalone.com
rfc822MailMember: bryn.yasuko@noodle.net
inetMailGroupVersion: 1.0
inetMailGroupStatus: active
dataSource: Mail Server 4.0
expandable: false
mail: basketball@stream.com
mailHost: buffalo.stream.com
rfc822MailAlias: b-ball_players@stream.com
```

Creating Posting Restrictions on Distribution Lists

Restrictions can be placed on what submitters or domains can or cannot send mail to the group. The restriction attributes are as follows:

- `authorizedSubmitter` defines the list of addresses that are authorized to send messages to the distribution list. If this attribute is not included in the LDIF record, the list is unrestricted, meaning it will not contain the authorized/

unauthorized submitters, or the authorized/unauthorized domains. The From: header address must match one of the addresses in the permitted list before the IMTA will route the message to a list of members.

- `unauthorizedSubmitter` specifies addresses not permitted to post messages to the list. The sender's address is compared against those in this attribute. If there is a match then the message is rejected.
- `authorizedDomain` specifies the domain names from which users are authorized to post messages to the distribution list. The wildcard character is "*". Using the wildcard character one may optionally replace a sub-domain to authorize the entire DNS hierarchy below a given top or sub-domain.
- `unauthorizedDomain` defines the domain names from which users cannot post messages to the distribution list.

Note – Note that DNs values for `authorizedSubmitter`, `unauthorizedSubmitter` must have the prefix `ldap:///`. Refer to Section , "Format of inetMailGroup Attribute Values," on page 51.

Precedence Rules

The following precedence rules are followed by the IMTA when deciding whether it should accept the message for further processing or not ("From:" address is used in all the rules when looking for match):

1. If `unauthorizedSubmitter` attribute exists in the LDAP entry, the sender's address must not match either the `mail` attribute or `rfc822MailAlias` attribute of any DN listed in the form of a `ldap:///<DN>` address and must not match the RFC-822 address listed in the form of a `mailto:<RFC-822> address`.
2. if `authorizedSubmitter` attribute exists in the LDAP entry, the sender's address must match either the mail attribute or `rfc822MailAlias` attribute of any DN listed in the form of a `ldap:///<DN>` address and must not match the RFC-822 address listed in the form of a `mailto:<RFC-822>` address.
3. if `unauthorizedDomain` exists in the LDAP entry, then sender's domain must not match the domain(s) listed in the `unauthorizedDomain` attribute.

4. If `authorizedDomain` attribute exists in the LDAP entry, then the sender's domain must match the domain(s) listed in the `authorizedDomain` attribute.

CODE EXAMPLE 4-6 LDIF Record for Creating Group Posting Restrictions.

```
dn: cn=basketball,ou=groups,dc=stream,dc=com,o=internet
objectClass: groupOfUniqueNames
objectClass: inetMailRouting
objectClass: inetMailGroup
cn: basketball
uniqueMember: cn=Kevin Cox (Lightening),ou=people,dc=stream,dc=com,o=internet
rfc822MailMember: camden.miyoko@abalone.com
rfc822MailMember: bryn.yasuko@noodle.net
inetMailGroupVersion: 1.0
unauthorizedSubmitter: xxx@porno.com
unauthorizedDomain: spam.net
inetMailGroupStatus: active
dataSource: Mail Server 4.0
expandable: false
mail: basketball@stream.com
mailHost: buffalo.stream.com
rfc822MailAlias: b-ball_players@stream.com
```

Designating Addresses for Requests

You can set the `requestTo` attribute to forward the distribution list subscription requests to a particular address. Note that the `requestsTo` attribute-value pair spans two lines and that the second line must start with a blank space. (If you are reading this on-line, this space may not be displayed in your browser.)

CODE EXAMPLE 4-7 LDIF Record for Creating a `requestTo` Attribute.

```
dn: cn=basketball,ou=groups,dc=stream,dc=com,o=internet
objectClass: groupOfUniqueNames
objectClass: inetMailRouting
objectClass: inetMailGroup
cn: basketball
requestsTo: ldap:///cn=Kevin Cox (Lightening)
 ,ou=people,dc=stream,dc=com,o=internet
rfc822MailMember: camden.miyoko@abalone.com
uniqueMember: cn=Kevin Cox (White Lightening)
 ,ou=people,dc=stream,dc=com,o=internet
rfc822MailMember: camden.miyoko@abalone.com
rfc822MailMember: bryn.yasuko@noodle.net
```

CODE EXAMPLE 4-7 LDIF Record for Creating a `requestTo` Attribute. *(Continued)*

```
inetMailGroupVersion: 1.0
inetMailGroupStatus: active
dataSource: Mail Server 4.0
expandable: false
mail: basketball@stream.com
mailHost: buffalo.stream.com
rfc822MailAlias: b-ball_players@stream.com
```

Setting Error Handling Parameters

Mail delivery error handling is set in one of two ways:

- Delivery errors are reported to the original sender.
- Delivery errors go back to the address specified in the `errorsTo` attribute.

Set the `errorsTo` attribute to the address to which distribution list errors are sent. When a list is expanded, the original return address in the envelope is replaced by this address. The intent is for errors to be sent to the owner of the list, rather than the message originator who generally has no control over the contents of the list. If the `errorsTo` attribute is not specified, errors are sent to the originator.

The Requirements for Internet Hosts [RFC1123] specify that all IMTAs should support a mechanism where a list is expanded, but with the original return address preserved. This is referred to by the RFC as *aliasing*. This can be achieved by omitting the `errorsTo` attribute.

Note that the `errorTo` attribute-value pair spans two lines and that the second line must start with a blank space. (If you are reading this on-line, this space may not be displayed in your browser.)

CODE EXAMPLE 4-8 LDIF Record for Setting `errorTo` Attribute.

```
dn: cn=basketball,ou=groups,dc=stream,dc=com,o=internet
objectClass: groupOfUniqueNames
objectClass: inetMailRouting
objectClass: inetMailGroup
cn: basketball
errorsTo: ldap:///cn=Kevin Cox (White Lightening)
 ,ou=people,dc=stream,dc=com,o=internet
uniqueMember: cn=Kevin Cox (White Lightening)
 ,ou=people,dc=stream,dc=com,o=internet
rfc822MailMember: camden.miyoko@abalone.com
rfc822MailMember: bryn.yasuko@noodle.net
```

CODE EXAMPLE 4-8 LDIF Record for Setting `errorTo` Attribute. *(Continued)*

```
inetMailGroupVersion: 1.0
inetMailGroupStatus: active
dataSource: Mail Server 4.0
expandable: false
mail: basketball@stream.com
mailHost: buffalo.stream.com
rfc822MailAlias: b-ball_players@stream.com
```

CHAPTER 5

Creating SIMS Administrators

A SIMS Administrator is a user with permissions to modify server configuration. To create SIMS administrators, add the object class `inetAdministrator` to the user entry and add the attribute `inetAdministeredServices` with the desired administrative rights in the desired administrative domain.

The format for assigning a value to `inetAdministeredServices` is as follows:

```
inetAdministeredServices: inetVersion=<service_version>,
ou=<service_name>,ou=services, dc=<domain_comp_1>,
...,dc=<domain_comp_N>,o=internet??<scope>
```

where

`service_version` = the version number of the service (e.g. 3.5 or 4.0). Specifying a version limits the administrator to accessing services of only this version. Leaving `service_version` out allows the administrator to access all versions of the service(s).

`service_name` = one of the following names.

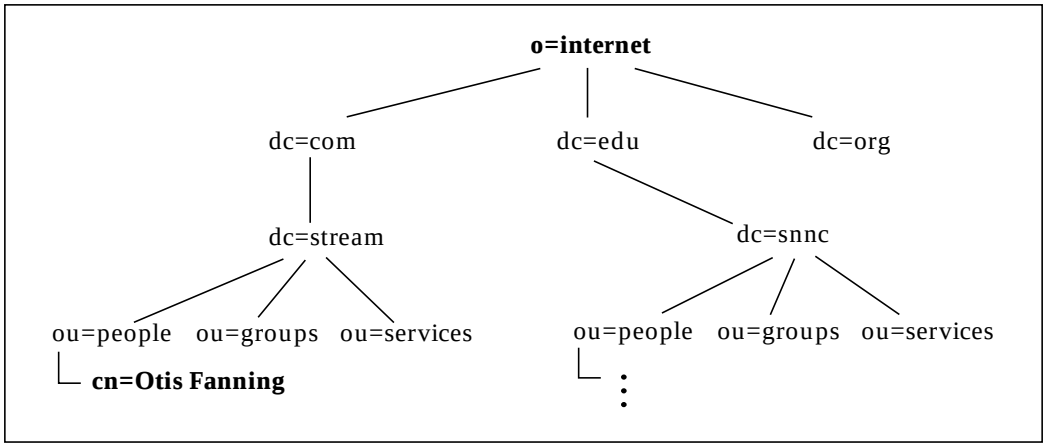
- `imta` - Access to IMTA configuration file changes.
- `msma` - Access to MS/MA administration functions.
- `provisioning` - Ability to perform provisioning tasks (rights to modify the directory). Note that to obtain full provisioning rights, you must also add the user to the appropriate ACLs. See the directory documentation for details.
- `calendar` - Access to calendar administration functions.
- `webaccess` - Access to WebAccess administration functions.
- `admin` - Access to ALL functions of administration server.
- If not specified, then the administrator has privileges for all services.

`ou=services` - this is only needed if the administrator is scoped by service. That is, if `service_name` is explicitly specified.

dc=<domain_comp_1>,...,dc=<domain_comp_N> is domain over which the user has administrative authority for the specified service.

<scope> is the part of the LDAP tree over which the administrative priviliges are granted. A value of sub specifies that administrative rights extend over the subtree beneath the most significant domain component in the DN and all contained LDAP entries. A value of base means administrative rights extends only to users immediately beneath the most significant domain component in the DN. You should consider the fact that users and groups are contained in ou=People and ou=Groups containers under the domain component node. Thus, in order to do a one level search, we have to prefix the search base (domain) with the name of the container (ou=People for users and ou=Groups for groups).

In the example below, Otis Fanning has all administrative privileges in the stream.com domain and its subdomains, as well as message store management privileges (for example, the ability to delete mailboxes using imdeluser()) in snnc.edu and its subdomains.



CODE EXAMPLE 5-1 LDIF Record for Creating a SIMS Administrator.

```
dn: cn=Otis Fanning,ou=People,dc=stream,dc=com,o=internet
objectClass: inetOrgPerson
objectClass: inetSubscriber
objectClass: inetMailRouting
objectClass: inetMailUser
objectClass: inetAdministrator
```

CODE EXAMPLE 5-1 LDIF Record for Creating a SIMS Administrator. *(Continued)*

```
cn: Otis Fanning
sn: Fanning
initials: T
givenName: Otis
mail: fanning@stream.com
uid: fanning
userPassword: secret
inetAdministeredServices: dc=stream,dc=com,o=internet??sub
inetAdministeredServices: ou=msma, ou=services,dc=snnc,dc=edu,o=internet??sub
inetAuthorizedServices: imap
inetAuthorizedServices: pop3
inetAuthorizedServices: imaps
inetAuthorizedServices: pop3s
inetAuthorizedServices: sunw_webaccess
inetAuthorizedServices: sunw_calendar
inetMailUserVersion: 1.0
inetSubscriberStatus: active
rfc822MailAlias: Otis.Fanning@stream.com
mailDeliveryOption: mailbox
mailHost: buffalo.stream.com
mailFolderMap: Sun-MS
dataSource: @(#)mkdirdata.sh 1.10 02/19/99
mailQuota: -1
```

Glossary

ACAP Application Configuration Access Protocol. A protocol which enhances IMAP by allowing the user to set up address books, user options, and other data for universal access.

access control rules Rules specifying user permissions for a given set of directory entries or attributes.

access control list (ACL) A set of data associated with a directory that defines the permissions that users and/or groups have for accessing it.

Administration Console or Admin Console A GUI (graphical user interface) which enables you to configure, monitor, maintain, and troubleshoot the SIMS components.

address mapping See forward address mapping or reverse address mapping.

address token The address element of a rewrite rule pattern.

Administration Services A service daemon that administers components of SIMS through a GUI.

agent In the client-server model, the part of the system that performs information preparation and exchange on behalf of a client or server application. See also *MTA*.

alias An alternate name of an email address.

alias file A file used to set aliases not set in a directory, such as the postmaster alias.

APOP Authenticated Post Office Protocol. Similar to the Post Office Protocol (POP), but instead of using a plaintext password for authentication, it uses an encoding of the password together with a challenge string.

attribute The form of information stored and retrieved by the directory service. Directory information consists of entries, each containing one or more attributes. Each attribute consists of a type identifier together with one or more values. Each directory read operation can retrieve some or all attributes from a designated entry.

attribute index An index, or list, of entries which contains a given attribute or attribute value.

autoreply option file A file used for setting options for autoreply, such as vacation notices.

backbone The primary connectivity mechanism of a distributed system. All systems that have connectivity to an intermediate system on the backbone are connected to each other. This does not prevent you from setting up systems to bypass the backbone for reasons of cost, performance, or security.

bang path An address for sending e-mail via UUCP that specifies the entire route to the destination computer. It separates each host name with an exclamation point, which is also known as a bang. For example, the bang path `midearth!shire!bilbo!jsmith` would go to the `jsmith` user account on the `bilbo` host, which is reached by first going to `midearth` and then `shire`.

CA Certificate Authority. An organization that issues digital certificates (digital identification) and makes its public key widely available to its intended audience.

directory cache A temporary storage of information that has been retrieved from the directory.

Certificate Authority See CA.

channel An interface with another SIMS component, another email system, or a mail user agent.

character set labels A name or label for a character set.

client-server model A computing model in which powerful networked computers provide specific services to other client computers. Examples include the name-server/name-resolver paradigm of the DNS and fileserver/file-client relationships such as NFS and diskless hosts.

cn LDAP alias for common name.

composition The process of constructing a message by the Mail User Agent (MUA). See also *MUA*.

configuration file A file that contains the configuration parameters for a specific component of the SIMS system.

congestion thresholds A disk space limit that can be set by the system administrator that prevents the database from becoming overloaded by restricting new operations when system resources are insufficient.

conversion channel Converts body of messages from one form to another.

cookie Cookies are text-only strings entered into the browser's memory automatically when you visit specific web sites. Cookies are programmed by the web page author. Users can either accept or deny cookies. Accepting the cookies allows the web page to load more quickly and is not a threat to the security of your machine.

ciphertext Text which has been encrypted. Opposite of plaintext.

daemon A UNIX program that is not invoked explicitly, but lies dormant waiting for some condition(s) to occur. The instigator of the condition need not be aware that a daemon is lurking (though often a program will commit an action only because it knows that it will implicitly invoke a daemon). Typical daemons are print spoolers, e-mail handlers, and schedulers that start up another process at a designated time or condition.

data store A store that contains directory information, typically for an entire directory information tree.

DC tree Domain Component tree. A directory information tree that mirrors the DNS network syntax. An example of a distinguished name in an DC tree would be `cn=billbob,dc=bridge,dc=net,o=internet`

defragmentation The Multiple InternetMail Extensions (MIME) feature that enables a large message that has been broken down into smaller messages or fragments to be reassembled. A Message Partial Content-Type header field that appears in each of the fragments contains information that helps reassemble the fragments into one message. See also *fragmentation*.

delegated administrator A person who has the privileges to add, modify, delete, and search for group or user entries at a specified hosted domain.

Delegated Managment Console A web browser-based software console that allows delegated administrators to add and modify users and groups to a hosted domain. Also allows end users to change their password, set message forwarding rules, set vacation rules, and list distribution list subscriptions.

delegated managment server A daemon program that handles access control to the directory by hosted domains.

denial of service attack A situation where an individual intentionally or inadvertently overwhelms your mail server by flooding it with messages. Your server's throughput could be significantly impacted or the server itself could become overloaded and nonfunctional.

dereferencing an alias Specifying, in a bind or search operation, that a directory service translate an alias distinguished name to the actual distinguished name of an entry.

destination channel The last element of a host/domain rewrite rule, in whose queue a message should be placed in for delivery.

directory cache A cache containing the directory information used by the IMTA to deliver mail.

directory context The point in the directory tree information at which a search begins for entries used to authenticate a user and password for Sun Message Store access.

directory entry A set of directory attributes and their values identified by its distinguished name. Each entry contains an object class attribute that specifies the kind of object the entry describes and defines the set of attributes it contains. Also called the *IMTA directory cache.*

directory information tree The tree-like hierarchical structure in which directory entries are organized. Also called a DIT. DITs can be organized along the DNS (DC trees) or Open Systems Interconnect networks (OSI trees).

directory schema The set of rules that defines the data that can be stored in the directory.

directory service A logically centralized repository of information. The component in SIMS that stores user, distribution list, and configuration data.

directory synchronization Because information stored in the directory service is updated as new entries are added, modified and deleted, the information in the IMTA directory cache must be periodically updated with the current information in the directory service. This process is called directory synchronization. Sometimes called a dirsync in reference to the `imta dirsync` command.

dirsync option file A file used to set options for the `dirsync` program which cannot be set through the command line.

disconnected state The mail client connects to the server, makes a cache copy of selected messages, then disconnects from the server.

distinguished name The comma-separated sequence of attributes and values that specify the unique location of an entry within the directory information tree. Often abbreviated as DN.

distribution list A list of email addresses (users) that can be sent a message by specifying one email address. Also called a group. See also *expansion*, *member*, *moderator*, *owner,* and *alias*.

distribution list owner An individual who is responsible for a distribution list. An owner can add or delete distribution list members. See also *distribution list*, *expansion*, *member,* and *moderator.*

DIT See *directory information tree*.

DN Distinguished name.

dn LDAP alias for distinguished name.

DNS Domain Name System. A distributed name resolution software that allows computers to locate other computers on a UNIX network or the Internet by domain name. DNS servers provide a distributed, replicated, data query service for translating hostnames into Internet addresses.

DNS database A database of domain names (host names) and their corresponding IP addresses.

domain A group of computers whose hostnames share a common suffix, the *domain name*. Syntactically, an Internet domain name consists of a sequence of names (labels) separated by periods (dots), for example, `tundra.mpk.ca.us`.

domain quota The amount of space, configured by the system administrator, allocated to a domain for email messages.

domain rewriting rules See also *rewrite rules*.

domain template The part of a rewrite rule that defines how the host/domain portion of an address is rewritten. It can include either a full static host/domain address or a single field substitution string, or both.

dsservd A daemon that operates that accesses the database files that hold the directory information, and communicates with directory clients using the LDAP protocol.

EMAPI Extended MAPI Service Provider. Transparently turns Microsoft Exchange client into an Internet standard IMAP/LDAP client. See also *IMAP, LDAP.*

encryption Scrambling the contents of a message so that its contents cannot be read without the encryption, or code key.

entries User, group, or organizational data used to configure message accounts.

envelope The part of an Internet mail message that contains the delivery information. The envelope contains the originator and recipient information associated with a message.

ESMTP Extended Simple Mail Transfer Protocol. An Internet message transport protocol.

expander Part of an electronic mail delivery system which allows a message to be delivered to a list of addressees. Mail expanders are used to implement mailing lists. Users send messages to a single address (e.g., hacks@somehost.edu) and the mail expander takes care of delivery to the mailboxes in the list. Also called *mail exploders.*

expansion This term applies to the IMTA processing of distribution lists. The act of converting a message addressed to a distribution list into enough copies for each distribution list member.

expunge The act of marking a message for deletion and then permanently removing it from you INBOX.

external channel An interface between the IMTA and either another SIMS component or another component outside the SIMS email system.

failover The automatic transfer of a computuer service from one system to another to provide redundant backup.

Filesharing Transport This type of transport moves messages between the UNIX operating system and the PC running a client through a shared file system available to both platforms. When a channel is configured to use filesharing transport, the shared directory to use for the file exchange must be specified.

firewall A dedicated gateway machine with special security precautions used to service outside network, especially Internet, connections and dial-in lines. The idea is to protect a cluster of more loosely administered machines hidden behind the firewall from unwanted entry from outside the firewall.

folder Named place where mail is stored. Also called a *mailbox*. Inbox is a folder that stores new mail. Users can also have folders where mail can be stored. A folder can contain other folders in a hierarchical tree. Folders owned by a user are called *private folders*. See also *shared folders*.

Folder Check A utility which checks the accessibility of messages and folders and verifies links. This utility is used as part of the regular maintenance of SIMS.

forward address mapping Message envelopes, TO:address, are processed to a mapping table. The result of the mapping is tested. If necessary, the exact form of the envelope is exchanged for another which can then be processed by a different, and perhaps non-compliant RFC 822, mail system.

FQDN See fully qualified domain name.

fragmentation The Multiple Internet Mail Extensions (MIME) feature that allows the breaking up of a large message into smaller messages. See also *defragmentation*.

full static host/domain address The portion of a host/domain address elements set off by decimals as part of the domain template. See also *domain template*.

fully qualified domain name The full name of a system, consisting of its local host name and its domain name. For example, *class* is a host name and *class.sun.edu* is an fully qualified domain name. A fully qualified domain name should be sufficient to determine a unique Internet address for any host on the Internet. The same naming scheme is also used for some hosts that are not on the Internet, but share the same name-space for electronic mail addressing. A host which does not have a fully qualified domain name must be addressed using a bang path.

gateway The terms *gateway* and *application gateway* refer to systems that do translation from one native format to another. Examples include X.400 to/from RFC 822 electronic mail gateways. A machine that connects two or more electronic mail systems (especially dissimilar mail systems on two different networks) and transfers messages between them. Sometimes the mapping and translation can

be complex, and it generally requires a store-and-forward scheme whereby the message is received from one system completely before it is transmitted to the next system after suitable translations.

global log manager A utility that handles log information from each Sun Internet Mail Server component.

group Same as a distribution list.

group folders These contain folders for shared and group folders. See *shared folder.*

header The part of an Internet mail message that is composed of a field name followed by a colon and then a value. Headers include delivery information, summaries of contents, tracing, and MIME information.

hosted domain An email domain that is outsourced by an ISP. That is, the ISP provides email domain hosting for an organization by operating and maintaining the email services for that organization. A hosted domain shares the same SIMS host with other hosted domains. In earlier LDAP-based email systems, a domain was supported one or more email server hosts. With SIMS, many domains can be hosted on a single server. Hosted domains are also called *virtual hosted domains* or *virtual domains*.

host name The logical name assigned to a computer. On the Web, most hosts are named `www`; for example, `www.mycompany.com`. If a site is composed of several hosts, they might be given different names such as `support.mycompany.com` and `sales.mycompany.com`. `support` and `sales` are the host names, `mycompany` is the subdomain name, and `com` is the top-level domain name.

IMAP4 Internet Message Access Protocol. IMAP4 provides advanced disconnected mode client access.

IMTA Internet Message Transfer Agent. IMTA routes, transports, and delivers Internet Mail messages within the email system.

internal channel An interface between internal modules of the IMTA. Internal channels include the reprocessing, conversion, and defragmentation channels. These channels are not configurable.

Internet A collection of networks interconnected by a set of routers that allow them to function as the largest single world-wide virtual network.

internet protocol address A 32-bit address assigned to hosts using TCP/IP. Also called the *IP address* and *internet address.*

invalid user An error condition that occurs during message handling. When this occurs, the message store sends a communication to the Internet Message Transport Agent (IMTA), the message store deletes its copy of the message. The IMTA bounces the message back to the sender and deletes its copy of the message.

ISP Internet Service Provider. A company that provides internet services to its customers including email, electronic calendaring, access to the world wide web, and web hosting.

job controller An IMTA daemon responsible for scheduling message delivery. Job controller also controls channel queues and determines the order of processing. Requests are processed in the order in which they are received by the system.

knowledge information Part of the directory service infrastructure information. The directory server uses knowledge information to pass requests for information to other servers.

LDAP Lightweight Directory Access Protocol. LDAP is a protocol used for the storage, retrieval, and distribution of information, including user profiles, distribution lists, and configuration data.

LDAP referrals An LDAP entry that consists of a symbolic link (referral) to another LDAP entry. An LDAP referral consists of an LDAP host and a distinguished name. LDAP referrals are often used to reference existing LDAP data so that this data does not have to be replicated. They are also used to maintain compatibility for programs that depend on a particular entry that may have been moved.

LDAP Server A software server that maintains an LDAP directory and services queries to the directory. The Sun Directory Services and the Netscape Directory Services are implementations of an LDAP Server.

LDAP server failover A backup feature for LDAP servers. If one LDAP server fails, the system can switch over to another LDAP server.

LDAP filter A way of specifying a set of entries, based on the presence of a particular attribute or attribute value.

LDBM LDAP Data Base Manager.

LDIF LDAP Data Interchange Format. A data format used to represent LDAP entries in text form.

local channel A channel that allows you to determine delivery options of local users and delivers mail to Solaris Operating Environment mailboxes.

lookup Same as a search, using the specified parameters for sorting data.

mailbox A place where messages are stored and viewed. See *folder.*

managed object A collection of configurable attributes, for example, a collection of attributes for the directory service.

mapping tables Two column tables which transform, map, an input string into an output string.

master directory server The directory server that contains the data that will be replicated.

master message catalog Contains message catalogs for the SIMS components.

master program A channel program that initiates a message transfer to another interface on its own.

member A user or group who receives a copy of an email addressed to a distribution list. See also *distribution list*, *expansion*, *moderator,* and *owner.*

Message Access and Store The SIMS components which store user messages and allow for retrieval and processing of messages.

Message Access Services Consists of protocol servers, software drivers, and libraries which support client access to the message store.

message access services The drivers and libraries that support client access to the SIMS message store.

message catalogs The log messages, command line responses, and graphical user interface screen text contained in the SIMS components.

message submission The client Mail User Agent (MUA) transfers a message to the mail server and requests delivery.

MIB Management Information Base. A collection of objects that can be accessed via a network management protocol. See also *SMI.*

MIME Multipurpose Internet Mail Extensions. A format for defining email message content.

moderator A person who first receives all email addressed to a distribution list before A) forwarding the message to the distribution list, B) editing the message and then forwarding it to the distribution list, or C) not forwarding the message to the distribution list. See also *distribution list*, *expansion*, *member,* and *owner.*

MTA Message Transfer Agent. An OSI application process used to store and forward messages in the X.400 Message Handling System. Equivalent to Internet mail agent. See *IMTA*.

MUA Mail User Agent. The client applications invoked by end users to read, submit, and organize their electronic mail.

mx record Mail Exchange Record. A DNS resource record stating a host that can handle electronic mail for a particular domain.

name resolution The process of mapping an IP address to the corresponding name. See also *DNS.*

namespace The space from which an object name is derived and understood. Files are named within the file namespace, domain components are named within the domain namespace.

naming attribute The final attribute in a directory information tree distinguished name. See also *relative distinguished name.*

naming context A specific subtree of a directory information tree that is identified by its DN. In SIMS, specific types of directory information are stored in naming contexts. For example, a naming context which stores all entries for marketing employees in the XYZ Corporation at the Boston office might be called ou=mktg, ou=Boston, o=XYZ, c=US.

NIS A distributed network information service containing key information about the systems and the users on the network. The NIS database is stored on the master server and all the replica or slave servers.

NIS+ A distributed network information service containing hierarchical information about the systems and the users on the network. The NIS+ database is stored on the master server and all the replica servers.

nondelivery report During message transmission, if the IMTA does not find a match between the address pattern and a rewrite rule, the IMTA sends a nondelivery report back to the sender with the original message.

notary messages Text messages sent by the MTA to an email sender indicating delivery or non-delivery status of a sent message.

o LDAP alias for `organization`

object class A template specifying the kind of object the entry describes and the set of attributes it contains. For example, SIMS specifies an `emailPerson` object class which has attributes such as `commonname`, `mail` (email address), `mailHost`, and `mailQuota`.

off-line state The mail client fetches messages from a server system to a client system, which may be a desktop or portable system and may delete them from the server. The mail client downloads the messages where they can be viewed and answered.

on-line state A state in which messages remain on the server and are remotely responded to by the mail client.

option files IMTA option files contain global parameters used to override default values of parameters which apply to IMTA as a whole, such as sizes for various tables into which various configuration and alias files are read.

OSI tree A directory information tree that mirrors the Open Systems Interconnect network syntax. An example of a distinguished name in an OSI tree would be `cn=billt,o=bridge,c=us`

ou LDAP alias for `organizationalUnit`

permanent failure An error condition that occurs during message handling. When this occurs, the message store deletes its copy of an email message. The Internet Message Transport Agent (IMTA) bounces the message back to the sender and deletes its copy of the message.

pipe channel A channel which performs delivery of messages via a per-user-site-supplied program. These programs must be registered in SIMS by the system administrator, and thus do not pose a security risk.

plaintext Unencrypted readable text. The opposite of cypher text

plaintext authentication Authentication that occurs by sending passwords over the network in plaintext. Considered a security problem since plaintext passwords can be easily captured over a network.

POP Post Office Protocol. POP provides remote access support for older mail clients.

populating the directory Entering information for users and distribution lists to the SIMS directory service.

protocol A formal description of messages to be exchanged and rules to be followed for two or more systems to exchange information.

provisioning The process of adding, modifying or deleting entries in the SIMS directory service. These entries include users and groups.

provisioning commands SIMS commands that provide provisioning functions. These commands are prefaced with `imadmin`.

proxy The mechanism whereby one system “fronts for” another system in responding to protocol requests. Proxy systems are used in network management to avoid having to implement full protocol stacks in simple devices, such as modems.

public key encryption A cryptographic method that uses a two-part key (code) that is made up of public and private components. To encrypt messages, the published public keys of the recipients are used. To decrypt the messages, the recipients use their unpublished private keys known only to them.

purge The process of permanently removing messages that have been deleted and are no longer referenced in user and group folders and returning the space to the Sun Message Store file system. See also *backup* and *restore*.

quota See user quota.

referral A process by which the directory server returns an information request to the client that submitted it, with information about the Directory Service Agent (DSA) that the client should contact with the request. See also *knowledge information*.

relaying A message is passed from one mail server to another mail server.

relative distinguished name The final attribute and its value in the attribute and value sequence of the distinguished name. See also *distinguished name*.

replica directory server The directory that will receive a copy of all or part of the data.

reprocessing channel Performs deferred processing. The reprocessing channel is the intersection of all other channel programs. It performs only the operations that are shared with other channels.

restore The process of restoring the contents of folders from a backup device to the Sun Message Store. See also *backup* and *purge*.

reverse address mapping Addresses are processed to a mapping table, with a reversal database, generally substituting a generic address, possibly on a central machine, for an address on a remote or transitory system.

rewrite rules Also known as domain rewriting rules. A tool that the Internet Mail Transport Agent (IMTA) uses to route messages to the correct host for delivery. Rewrite rules perform the following functions: (1) extract the host / domain specification from an address of an incoming message, (2) match the host / domain specification with a rewrite rule pattern, (3) rewrite the host / domain specification based on the domain template, and (4) decide which IMTA channel queue the message should be placed in.

RFC Request For Comments. The document series, begun in 1969, describes the Internet suite of protocols and related experiments. Not all (in fact very few) RFCs describe Internet standards, but all Internet standards are published as RFCs. See `http://www.imc.org/rfcs.html`.

root entry The first entry of the directory information tree (DIT) hierarchy.

router A system responsible for determining which of several paths network traffic will follow. It uses a routing protocol to gain information about the network, and algorithms to choose the best route based on several criteria known as “routing metrics.” In OSI terminology, a router is a Network Layer intermediate system. See also *gateway*.

routability scope Specifications which enable the IMTA to send messages by the most direct route, either to a specific user's folder, a group of folders, or to a mail host.

routing In an email system, the act of delivering a message based on addressing information extracted from the body of the message. The Internet Message Transfer Agent (IMTA) is the component responsible for routing messages.

safe file system A file system performs logging such that if a system crashes it is possible to rollback the data to a pre-crash state and restore all data. An example of a safe file system is Veritas File System, VxFS.

schema A set of rules which sets the parameters of the data stored in a directory. It defines the type of entries, their structure and their syntax.

sendmail This program acts as a mail transport agent for Solaris software. It is responsible for routing mail and resolution of mail addresses.

shared folder or shared mailbox *A* mailbox that can be viewed by members of a *distribution list*. Shared folders have an *owner* who can add or delete members to the group and can delete messages from a the shared folder. The can also have a moderator who can edit, block, or forward incoming messages.

SIMS administrator An individual who has a valid log in and password for the SIMS Admin Console. This person can also use this log in and password to execute the provisioning CLIs.

single field substitution string Part of the domain template that dynamically rewrites the specified address token of the host/domain address. See also *domain template.*

SKIP Simple Key management for IP. A security system that encrypts or scrambles the text of a message so only the receiving mail client or message server can decrypt or unscramble the text.

slave program A channel program that accepts transfers initiated by another interface.

smart host The mail server in a domain to which other other mail servers, forward messages if they do not recognize the recipients.

SMTP Simple Mail Transfer Protocol. The Internet electronic mail protocol. Defined in RFC 821, with associated message format descriptions in RFC 822.

SMTP Dispatcher A multithreaded connection dispatching agent which allows multiple multithreaded servers to share responsibility for a given service, thus allowing several multithreaded SMTP servers to run concurrently and handle one or more active connections.

SMTP intranet or internet channel A channel dedicated to relaying messages between the IMTA and a group of SMTP hosts within, or outside of, your mail network.

SMTP router channel SMTP channel that handles messages between the IMTA and firewall host.

sn LDAP alias for `surname`

SNMP Simple Network Management Protocol. The network management protocol of choice for TCP/IP-based internets.

subordinate reference The naming context that is a child of the naming context held by your directory server. See also *knowledge information*.

Sun Directory Services Sun Microsystems' implementation of an LDAP directory server. Provides storage of, and access to, user profiles, distribution lists, and other SIMS information. The Sun Directory Services is one of the three main SIMS components along with the IMTA and MS/MA.

Sun Internet Mail Server An enterprise-wide, open-standards based, scalable electronic message-handling system.

Sun Message Store The server from which mail clients retrieve and submit messages.

SSL Secure Sockets Layer is an open, non-proprietary security protocol for authenticated and encrypted communication between clients and servers.

synchronization The update of data by a master directory server to a replica directory server.

table lookup With a table consisting of two columns of data, an input string is compared with the data within the table and transformed to an output string.

tailor file An option file used to set the location of various IMTA components.

transient failure An error condition that occurs during message handling. The remote Internet Message Transport Agent (IMTA) is unable to handle the message when it's delivered, but may be able to later. The local IMTA returns the message to the channel queue and schedules it for retransmission at a later time.

transport protocols Provides the means to transfer messages between message stores.

uid User indentification. A unique string identifying a user to a system. Also referred to as a userid.

unsafe file system A file system that does not perform logging. If the system crashes, the state cannot be recreated and some data may be lost. You must also perform `imcheck` before activating message access to these files.

upper reference Indicates the directory server that holds the naming context above your directory server's naming context in the directory information tree (DIT).

user entry or user profile Fields that describe information about each user, required and optional, examples are: distinguished name, full name, title, telephone number, pager number, login name, password, home directory, etc.

user folders A user's email mailboxes.

user quota The amount of space, configured by the system administrator, allocated to a user for email messages.

user redirection The remote Internet Message Transport Agent (IMTA) cannot accept mail for the recipient, but can reroute the mail to a mail server that can accept it.

UUCP UNIX to UNIX Copy Program. A protocol used for communication between consenting UNIX systems.

valid user A condition that occurs during message handling. After the message store sends a communication to the Internet Message Transport Agent (IMTA), the IMTA deletes its copy of the message and it is now the message store's responsibility.

`/var/mail` The UNIX version 7 "`From`" delimited mailbox as implemented in the Solaris operating system.

virtual hosted domains or virtual domains See *hosted domains*.

workgroup Local workgroup environment, where the server performs its own routing and delivery within a local office or workgroup. Interdepartmental mail is routed to a backbone server. See also *backbone*.

X.400 A message handling system standard.

Index

P

R

S

T

U

W