Provisioning Guide

iPlanet Messaging Server

Release 5.0

806-4819-10
February 2001

Contents

List of Figures

List of Tables

List of Code Examples

About This Guide

This manual explains how to provision the iPlanet Messaging Server with users, mailing lists, domains, and administrators using LDAP. This guide is expected to be used with the *iPlanet Schema Reference Manual*.

Topics covered in this chapter include:

- Who Should Read This Guide
- What You Need to Know
- How This Guide is Organized
- Typographical Conventions
- Where to Find Related Information

Who Should Read This Guide

You should read this guide if you want to provision the iPlanet Messaging Server using LDAP. The audience for this guide consists of:

- Messaging system architects who want to develop customized provisioning tools that interface between Messaging Server entries in the iPlanet LDAP directory and their existing source of users, groups, and domains information such as a company database or billing system.
- Site Administrators who want to know how to create domain, user, group, or administrator entries using LDAP.

Readers are expected to have a basic understanding of LDAP, the Netscape Directory Server, and email concepts.

What You Need to Know

This guide assumes that you have a general understanding of the following:

- The Internet and the World Wide Web
- iPlanet Administration Server
- Netscape Directory Server and LDAP
- Email and email concepts
- Netscape Console

How This Guide is Organized

This guide contains the following chapters and appendix:

- About This Guide (this chapter)
- Chapter 1, “Provisioning Concepts and Technologies“
- Chapter 2, “Provisioning Domains“
- Chapter 3, “Provisioning Family Accounts“
- Chapter 4, “Provisioning Users“
- Chapter 5, “Provisioning Mailing Lists“
- Chapter 6, “Provisioning Messaging Server Administrators“
- Appendix A, “Root and Domain ACI Examples“

Typographical Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, code, directories, hostnames, distinguished names, on-screen computer output.	Edit your `msg.conf` file. Use `ls -a` to list all files. `Error: illegal port #`

Typeface or Symbol	Meaning	Example
AaBbCc123	User entered text.	% **cd madonna**
<AaBbCc123>	Command-line place holder or variable. Replace with a real name or value.	# **<***InstanceRoot***>/start-msg**
AaBbCc123	Book titles, new words or terms, or words to be emphasized.	*iPlanet Messaging Server Provisioning Guide*

Command Line Prompts

Command line prompts (for example, % for a C-Shell, or $ for a Korn or Bourne shell) are not displayed in the examples. Depending on which operating system environment you are using, you will see a variety of different command line prompts. However, you should enter the command as it appears in the document unless specifically noted otherwise.

Where to Find Related Information

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

`http://docs.iplanet.com/docs/manuals/messaging.html`

Listed below are the additional documents that are available:

- iPlanet Messaging Server Administrator's Guide
- iPlanet Messaging Server Reference Manual
- iPlanet Messaging Server Schema Reference (on-line only)
- iPlanet Messaging Server Provisioning Guide (on-line only)
- iPlanet Messaging Server Delegated Administrator Guide

Chapter 1

Provisioning Concepts and Technologies

This chapter describes the concepts and technology of provisioning the iPlanet Message Server. It contains the following sections:

Provisioning the iPlanet Message Server

Provisioning is the adding, modifying or deleting of iPlanet Messaging Server user, mailing list, system administrator, and domain entries in the directory server. The messaging server queries the directory for information about these elements as needed.

There are four provisioning interfaces in the iPlanet Messaging Server:

- The iPlanet Delegated Administrator for Messaging console
- The iPlanet Delegated Administrator for Messaging command line utilities
- The iPlanet Messaging Server Administration Console
- The Messaging Server LDAP directory

This guide describes how to provision through LDAP. Reference may be made to other methods of provisioning, but the focus of this manual is provisioning through LDAP.

iPlanet Messaging Server Namespace

As installed, the iPlanet Messaging Server namespace consists of two directory information trees (DIT), an Organization Tree and a Domain Component Tree (DC Tree). Organization Trees contain the user and group entries. The DC Tree mirrors the local DNS structure and is used by the system as an index to the Organization Tree(s) containing the data entries (see Figure 1-2). The DC Tree also contains the domain's operating parameters such as smart hosts, routing hosts, domain disk quota, and so on.

The sections below describe the two-tree mechanism—how it works and why it was chosen. Further information, including how to port existing DITs to the iPlanet Messaging Server's two-tree mechanism, is provided in the *iPlanet Messaging Server Migration Guide*.

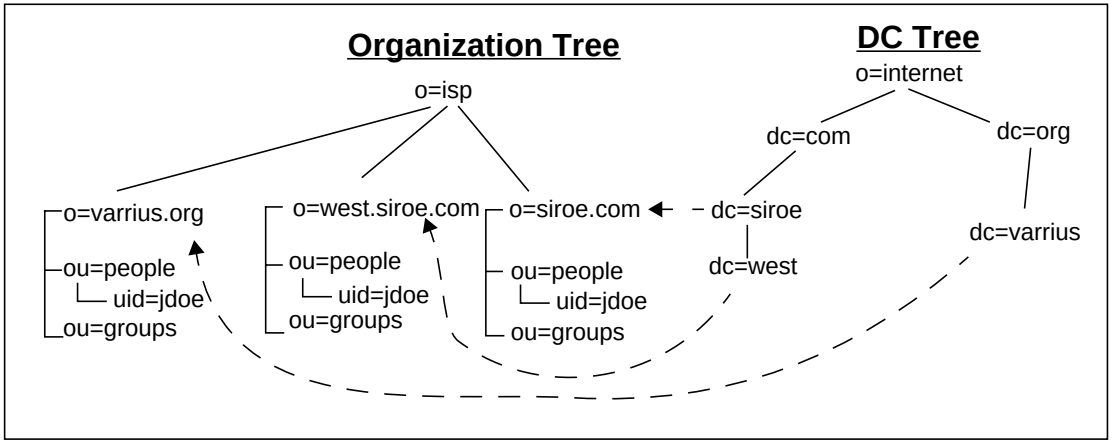
How the Two-tree Namespace Mechanism Works

This section describes how the iPlanet Messaging Server uses the two-DIT mechanism.

When the iPlanet Messaging Server searches for user / group entries, it first looks at the user / group's domain node in the DC Tree and extracts the value of the `inetDomainBaseDN` attribute. This attribute holds a DN reference to the organization subtree containing the actual user / group entry.

Using this model, the iPlanet Messaging Server can support entries stored in any type of directory Tree, provided that a domain component node in the DC Tree points to the node in the Organization Tree under which the users for that domain can be found. This relationship is reflected in the example shown in Figure 1-1 where the dotted line indicates the value of `inetDomainBaseDN`. Note that the node names in the Organization Tree do not have to match those in the DC Tree.

Figure 1-1 iPlanet Messaging Server Directory Structure—Example



In this example, data entries are added and modified under the Organization Tree, but the message server actually references the DC Tree. Let's consider three users:

	User 1	**User 2**	**User 3**
Name:	John Doe	John Doe	Jane Doe
Domain:	siroe.com	west.siroe.com	varrius.org
UID:	jdoe	jdoe	jdoe
Login:	jdoe@siroe.com	jdoe@west.siroe.com	jdoe@varrius.org

The login is derived from the UID and domain. In each of these cases, the server looks at the domain part of the login (the value after the @ sign) and retrieves the DN from the `inetdomainbasedn` attribute of the corresponding DC node. It then searches the subtree pointed at by the DN for a user entry where the UID equals the local part of the login (the value before the @ sign).

John Doe in `west.siroe.com` logs into the server using the login `jdoe@west.siroe.com`. The server follows the DN reference (`inetdomainbasedn`) in the DC node for `west.siroe.com` over to the subtree `o=west.siroe.com, o=isp`. It then searches for a user entry where the `uid=jdoe` in this subtree.

At installation, the iPlanet Messaging Server creates a default DC Tree and Organization Tree that is mapped to your existing DNS. When directory domain nodes are added using the Delegated Administrator command `imadmin domain create`, corresponding nodes are created in both the DC Tree and the Organization Tree. If you create nodes using the LDAP interface, you must create a node for the domain in the DC Tree, and a domain for the data in the Organization Tree. This is described in the *iPlanet Messaging Server Migration Guide*.

NOTE In order for mail to be delivered you must make sure that the domain has an MX record in the DNS.

Why Two Directory Information Trees?

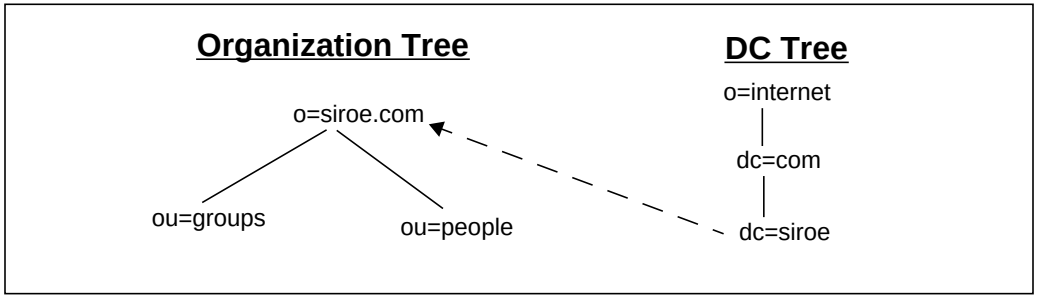
Although the iPlanet Messaging Server supports a single DC Tree containing configuration and user/group data, as installed, the iPlanet Messaging Server creates both a DC Tree and Organization Tree. This dual-tree mechanism provides the following enhancements:

- The ability to adopt existing directory deployments into iPlanet Messaging Server by creating a DC Tree that maps to existing DITs.
- The partitioning of data for organization-specific access control. That is, each organization can have a separate subtree in the DIT where user and group entries are located. Access to that data can be limited to users in that part of the subtree. This allows localized applications, such as iPlanet Delegated Administrator for Messaging to operate securely.
- The ability to have a distinct namespace for subdomains. For example, `west.siroe.com` and `siroe.com` may be mapped to separate organization subtrees allowing the creation of user entries with the same UID in each one of them.

Mapping an Existing DIT to iPlanet Messaging Server

The two-tree mechanism allows mapping existing DITs to the iPlanet Messaging Server. This is shown in the figure below which depicts an existing NMS DIT (o=siroe.com) mapped to a DC Tree in the iPlanet Messaging Server. This process is described in detail in the *iPlanet Messaging Server Migration Guide*.

Figure 1-2 Mapping Existing DIT to iPlanet Messaging Server—Example

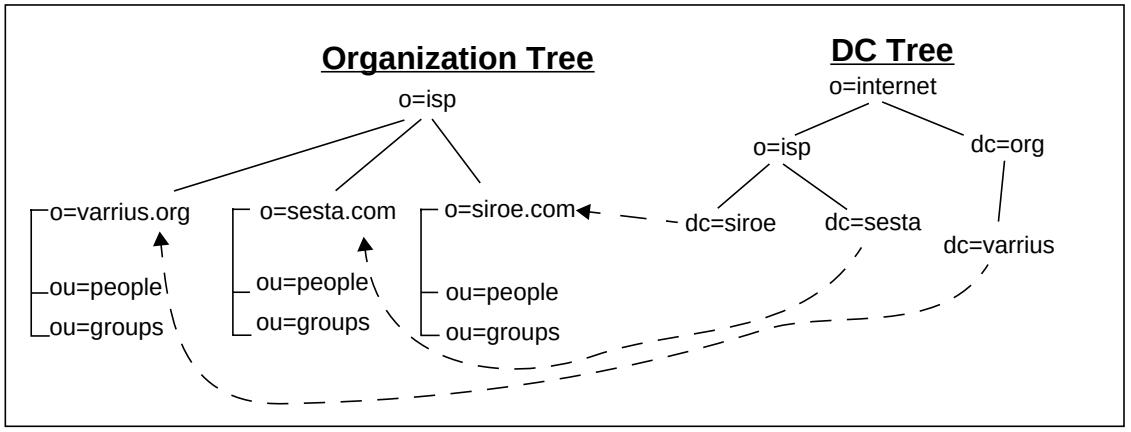


Partitioning Data for Access Control

The two-tree mechanism provides for data partitioning and access control within each partition. This feature is important in a multi-tenant directory since it is a strict requirement that one customer cannot access data pertaining to other customers stored in the same directory tree. Data partitioning and access control permit outsourcing of applications to distinct customer organizations in a secure manner such that user / group data for each organization can be stored separately from other organizations. An example of this type of application is the iPlanet Delegated Administrator for Messaging.

For this reason, outsourced messaging customers are represented in a well-defined subtree (typically a domain) within the overall DIT. This subtree is used to store all service data belonging to the customer. This concept is depicted in Figure 1-3.

Figure 1-3 Partitioning Data for TEST Access Control—Example



Providing Distinct Namespaces for Subdomains

The two-tree mechanism provides distinct namespaces for subdomains. This allows the same login name to be used in subdomains, for example, `jdoe@siroe.com` and `jdoe@west.siroe.com` can be two separate and legal email addresses. This is shown in Figure 1-1.

iPlanet Messaging Server Data Model

The basic data model of the iPlanet Messaging Server object classes is to extend LDAP entry *types* (for example, user, group, domain) created by *core object classes* by overlaying them with *shared classes* (object classes can be shared by more than one service) and *service-specific object classes* (classes specific to a certain type of server). This relationship is depicted in the table below.

Table 1-1 Entry types and Corresponding Object Classes

Class / Type	Core Classes	Shared Classes	Messaging Server Classes
DC Tree Domain	domain, inetdomain		mailDomain, nsManagedDomain, icsCalendarDomain
Org. Tree Domain	organization		nsManagedDomain
Email User	person, inetUser, organizationalPerson, inetOrgPerson	ipUser, userPresenceProfile	inetMailUser, inetLocalMailRecipient, nsManagedPerson
Group	groupOfUniqueNames		inetMailGroup, inetLocalRecipient, inetMailGroupManagement, nsManagedMailList
Family Account	inetManagedGroup		nsManagedDept

Using *email user* type as an example, the following object classes provide the following types of attributes:

`person` provides attributes for describing a person.

`organizationalPerson` provides attributes for describing a person belonging to an organization.

`inetOrgPerson` provides basic internet user attributes.

`ipUser` holds the personal address book attribute, the class of service template, and the DN of the family account as applicable.

`inetUser` represents a user account and is used in conjunction with `inetMailUser` and `ipUser` for creating a mail account.

`inetSubscriber` is an optional object class that represents a subscriber account. It provides account ID and challenge/response attributes.

`inetMailUser` represents a mail account and provides most of the user specific mail account attributes.

`inetLocalMailRecipient` represents a local (intra-organizational) email recipient specifying the recipient's email address(es), and providing routing information pertinent to the recipient.

ACI Architecture

Access Control Information instructions (ACIs) control user access to the directory. There are several types of messaging server users which require different levels of access to the directory. Some of these user types are as follows:

- *Normal email user*. This user type simply sends and receives email and requires such permissions as modifying password and starting vacation mode.
- *Top-level administrator* has permission to do anything to any entry in the directory.
- *Message Store Administrator* has permissions to view mailboxes and manage the message store for the system or domain.
- *Domain Administrator* can create, modify and delete mail user, mailing list, and family account entries in a domain.
- *Domain Organization Administrator* creates, modifies and deletes mail user and mailings list entries in a domain organization.
- *Family Group Administrator* has permission to add and remove family members in a family group entry.

Each of these user types have specific ACIs assigned to them at the root or domain level of the DC and Organization Trees (Figure 1-4). By assigning ACIs in the root and domain entries instead of for each user, access can be scoped to a domain or to the entire system. Thus, ACIs specified on the root node will apply to entries in the entire system, while ACIs specified in a domain apply only to entries in that domain. (For detailed ACI information see the *iPlanet Directory Administration Guide*.)

ACIs for the various administrators are conferred upon specific groups. To create an administrator you simply add a user to the group and add a group back pointer attribute (`memberof`) in the user entry. For example, at installation a group called `cn=Domain Administrators, ou=groups, <DN of domain>` is created with specific ACI privileges. To create a Domain Administrator, simply add a user to the group and add the `memberof` attribute to the user's entry.

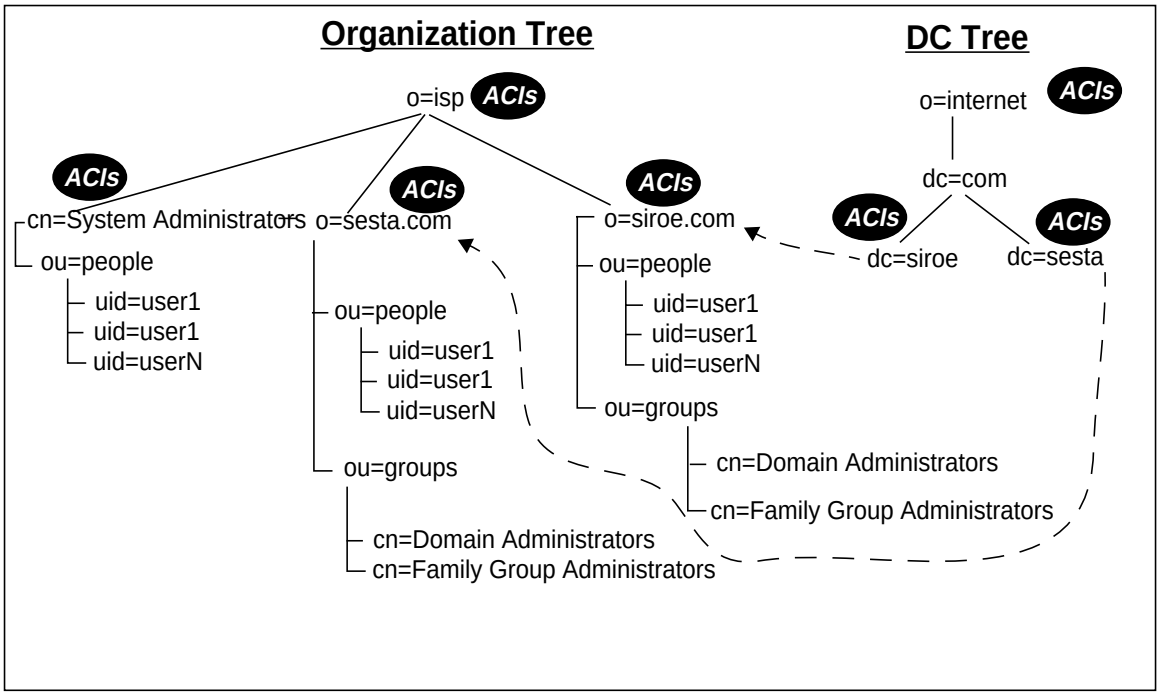
To create a Family Group Administrator, a user is added to the mailing list `cn=Family Group Administrators, ou=groups, <DN of domain>`
For complete instructions on creating administrators, see Chapter 6, "Provisioning Messaging Server Administrators."

In the configuration shown below. ACIs will be specified in the following entries:

```
o=isp
o=sesta.com,o=isp
o=siroe.com,o=isp
o=internet
dc=siroe, dc=com, o=internet
dc=sesta, dc=com, o=internet
```

The ACIs installed for domains and root entry nodes are shown in Figure 1-4.

Figure 1-4 ACI Example



Class of Service

The class of service (COS) feature allows you to create a named set of fixed features and attributes that can be applied to specified users. The class of service feature allows you to create a template of attributes which can be conferred upon user entries with a single attribute. For example, if you are an ISP, you could create two levels of mail service called *Hall of Fame* and *All-Star*. The Hall of Fame class of

service could provide users with IMAP, secure IMAP, POP3, and HTTP (Web mail) mail services as well as 5 gigabytes of message store disk space. The All-Star class of service could provide POP3 mail services along with five megabytes of message store disk space.

NOTE LDAP search requests containing a filter that references an attribute defined by class of service will not be serviced. For example, you cannot successfully search on the attribute `mailquota` if `mailquota` is only defined in a class of service template and not in user entries. The server will respond with an *unwilling to perform* error message when presented with such a request.

Setting Up Class of Service for the iPlanet Messaging Server

The basic procedures for adding the class of service feature is as follows:

1. Add the COS plug-in to `slapd.ldbm.conf`.
2. Create a COS mail scheme entry. The COS mail scheme defines the following:
   - Location of COS template definitions in the directory.
   - Directories containing the user entries to which the class of service can be applied.
   - Name of the attribute (`inetCOS`) used to specify the class of service template applied to a user entry.
   - A list of attributes to be used in a template.
3. Create the class of service template entries.
4. Assign a class of service to user entries.

These procedures are described in detail at the following web site:

http://docs.iplanet.com/docs/manuals/deladmin/45/html/06_cos.htm#25217

Specific iPlanet Messaging Server class of service issues and an example are described in the next section. As you implement the class of service feature in your system, use the following parameter values when you reach the step on *Managing COS Schemes*:

Table 1-2 iPlanet Messaging Server Class of Service Parameter Values

Parameter	Value
DN of container for class of service schemes and templates	ou=COS, *<domain's DN>*
DN for Mail Scheme entry	cn=mail scheme, ou=COS, *<domain's DN>*
DN of class of service container (cosTemplateDn)	ou=MailSchemeTemplates,ou=COS,*<domain's DN>*
Attribute for assigning a class of service to entries (cosSpecifier)	inetCOS

Class of Service Example

We will use the example described in the previous section by creating two classes of services called *Hall of Fame* and *All-Star* mail service for the hosted domain called sesta.com. The Hall of Fame class of service will provide users with IMAP, secure IMAP, POP3, and HTTP (Web mail) mail services as well as 5 gigabytes of message store disk space. The All-Star class of service could provide POP3 mail services along with 5 megabytes of message store disk space.

1. Install the COS plug-in on the Directory Server. Refer to:
   `http: //home.netscape.com/eng/server/directory/DSRK/4.1/cos.htm`
2. Create a mail scheme entry using the example LDIF entry below:

```
dn: cn=mail scheme,ou=COS,o=sesta.com, o=isp
objectclass: top
objectclass: cosDefinition
cosTemplateDn: ou=MailSchemeTemplates,ou=COS,o=sesta.com, o=isp
cosTargetTree: ou=People,o=sesta.com, o=isp
cosSpecifier: inetCOS
cosAttribute: mailQuota
cosAttribute: mailAllowedServiceAccess
```

- `dn: cn=mail scheme,ou=COS,o=sesta.com, o=isp`

  COS mail scheme entry DN.

- `objectclass: cosDefinition`

  Object class that defines the class of service scheme entry.

- `cosTemplateDn: ou=MailSchemeTemplates,ou=COS,o=sesta.com, o=isp`

  Multi-valued attribute that contains the subtree(s) under which the COS template entries for this scheme are stored.

- `cosTargetTree: ou=People,o=sesta.com, o=isp`

  Multi-valued attribute that contains the subtree to which the COS scheme applies.

- `cosSpecifier: inetCOS`

  Name of the attribute used to specify the COS template applied to a user entry.

- `cosAttribute: mailQuota`
  `cosAttribute: mailAllowedServiceAccess`

  Attributes to be used in a template entry.

3. Create COS template entries.

Below is the LDIF for two template entries for the Hall of Fame and All-Star templates.

```
dn: uid=All-Star,ou=MailSchemeClasses,ou=COS,o=sesta.com, o=isp
objectclass: top
objectclass: inetUser
objectclass: inetMailUser
mailQuota: 5000000
mailAllowedServiceAccess: +pop3:*
```

```
dn: uid=Hall of Fame,ou=MailSchemeClasses,ou=COS,o=sesta.com, o=isp
objectclass: top
objectclass: inetUser
objectclass: inetMailUser
mailQuota: 5000000000
mailAllowedServiceAccess: +imap, imaps, pop3, http:*
```

- `dn: uid=All-Star,ou=MailSchemeClasses,ou=COS,o=sesta.com, o=isp`
  `dn: uid=Hall of Fame,ou=MailSchemeClasses,ou=COS,o=sesta.com, o=isp`

DN of the COS templates

- ```
  objectclass: top
  objectclass: inetUser
  objectclass: inetMailUser
  mailQuota: 5000000
  mailAllowedServiceAccess: +pop3:*
  ```

  Attributes and object classes in the All-Star template.

- ```
  objectclass: top
  objectclass: inetUser
  objectclass: inetMailUser
  mailQuota: 5000000
  mailAllowedServiceAccess: +imap, imaps, pop3:*
  ```

  Attributes and object classes in the Hall of Fame template.

4. Add class of service templates to users.

```
dn: uid=Havlicek,ou=People,o=sesta.com, o=isp
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: userPresenceProfile
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
cn: John Havlicek
sn: Havlicek
initials: JH
givenName: John
mail: john.havlicek@sesta.com
mailAlternateAddress: Havlicek@sesta.com
mailHost: mail.siroe.com
uid: Havlicek
dataSource: iPlanet Messaging Server
userPassword: secret
inetUserStatus: active
mailUserStatus: active
mailMsgQuota: 100
inetCos: Hall of Fame

dn: uid=Hornicek,ou=People,o=sesta.com, o=isp
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: userPresenceProfile
```

```
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
cn: Jeff Hornicek
sn: Hornicek
initials: JH
givenName: Jeff
mail: jeff.hornicek@sesta.com
mailAlternateAddress: Hornicek@sesta.com
mailDeliveryOption: mailbox
mailHost: mail.siroe.com
uid: Hornicek
dataSource: iPlanet Messaging Server 5.0
userPassword: secret
inetUserStatus: active
mailUserStatus: active
mailMsgQuota: 100
inetCos: All-Star
```

Chapter 2

Provisioning Domains

This chapter describes how to create the domains and organizational units needed to provision the iPlanet Message Server. Note that some of these units are created during installation or when using the Delegated Administrator. However, we show these procedures for educational purposes. Attribute descriptions in this guide are overviews. Complete descriptions are available in the *iPlanet Schema Reference Manual*. This chapter consists of the following sections:

Domain ACIs

ACIs are required for all root and domain entries in both the DC and Organization Trees. The ACIs control access to the directory by the various types of users and administrators as well as the iPlanet Delegated Administrator for Messaging tool. For a more complete discussion of Messaging Server ACI design for user access, see “ACI Architecture,” on page 24.

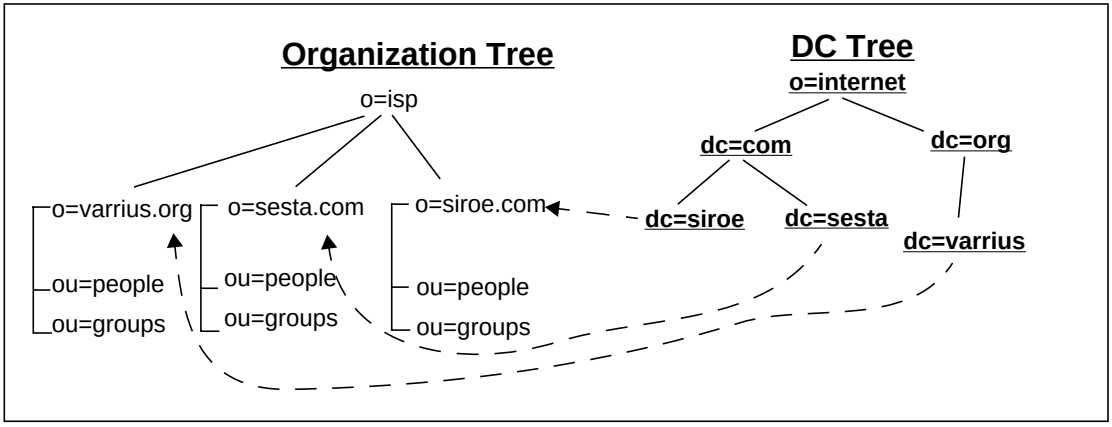
Upon installation ACIs are installed for the root-level of both the DC and Organization Trees, as well as the top-level domains of the Organization Tree (see Appendix A, “Root and Domain ACI Examples“). These ACIs apply to the node as well as all sub-nodes. Thus, ACIs created on the root node will apply to all the domains under the root. ACI rules can be viewed by doing an `ldapsearch` for a specific domain. The rules are displayed after the attributes of the domain entry and have the following format:

```
# Anonymous access control
#
# Allow anonymous read and search access to user entries
#
aci: (targetattr != "userPassword")
 (targetfilter=(objectClass=nsManagedPerson))
 (version 3.0; acl "Anonymous access to User entries";
 allow (read,search)
 userdn="ldap:///anyone";)
#
```

This ACI rule allows anyone to search and read everything but `userPassword` in any entry containing the object class `nsManagedPerson`. Pound signs (#) are non-executable comments. `targetattr` specifies the attributes upon which to act. `targetfilter` is the object class for which to search. `version` is a user defined version number and comment. `allow` lists the privileges allowed (read, write, search, delete, modify). `userdn` specifies who may act upon these attributes.

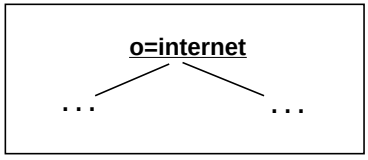
For a more complete discussion of directory ACIs, refer to the *Netscape Directory Server 4.1 Administrator’s Guide*.

Creating a DC Tree



The DC Tree nodes contain the operating parameters of a specific domain (for example routing host, disk quota, etc.) as well as a pointer to the sub-tree containing the user and mailing list entries for the domain. The DC Tree mirrors the DNS structure and a default DC Tree is created upon installation. As new hosted domains or domain organizations are created, however, new domain and domain organization nodes must be created in both the DC and Organization Trees. This can be done using the iPlanet Delegated Administrator for Messaging command `imadmin domain create` (see the command description in the *iPlanet Messaging Server Reference Manual*) or the iPlanet Delegated Administrator for Messaging console (see the *iPlanet Delegated Administrator Guide*). In this section, however, we will only explain how to create them using LDAP. Note that MX records also need to be added for new domains.

Creating the DC Tree Root Domain Entry



The root entry is the top level node of a directory tree. In the DC Tree the convention is to set the root to `o=internet`. If you specify the root to something other than `o=internet`, make sure that `service.dcroot` in the messaging server configuration matches the DN of the DC Tree root. The LDIF record for creating the DC root is shown in Code Example 2-1. Note that the root nodes for the DC and Organization Trees are created during installation, but we show the LDIF for instructional purposes. The default ACIs are shown in Appendix A, “Root and Domain ACI Examples.

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

Code Example 2-1 LDIF Record for a DC Tree Root

```
dn: o=internet
objectClass: organization
o: internet
description: Root level node in the Domain Component (DC) tree
```

DC Tree Root Entry Attributes and Object Classes

- `dn: o=internet`

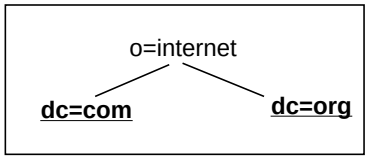
  The distinguished name (`dn`) uniquely identifies the directory entry in the tree. The default root node in the DC Tree is `o=internet`.

- `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, but only “o” is required. `o` must have the same value as set in the `dn` of this entry.

NOTE All object classes inherit the `top` object class, so it is not necessary to include the line `objectClass: top` in the LDIF code for creating entries.

Creating DC Tree Top-level Domain Entries



The top-level domain entries are just below root and must mirror the domain components of the DNS domains. The LDIF record for creating the top-level nodes is shown in Code Example 2-2. Note that a root, top-level, and default domain nodes are created during installation. We show the LDIF for instructional purposes. ACIs are also created (see Appendix A, “Root and Domain ACI Examples).

Code Example 2-2 LDIF Record for Top-level Nodes

```
dn: dc=com, o=internet
objectClass: domain
dc: com
description: top level .com domain in the DC Tree

dn: dc=org, o=internet
objectClass: domain
dc: org
description: top level .org domain in the DC Tree
```

DC Tree Top-level Node Attributes and Object Classes

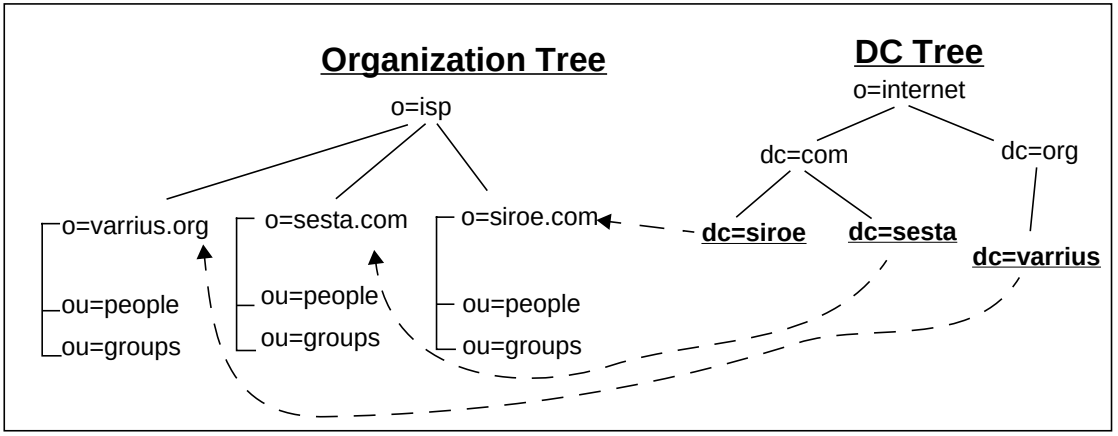
- `dn: dc=com,o=internet`
  `dn: dc=org,o=internet`

  `dn` specifies the top-level domain node entries.

- `objectClass: domain`

  The `domain` object class is used to create all the container entries (except for the root entry) in the DC Tree.

Creating a DC Tree Hosted Domain Entry



For each hosted domain in the Organization Tree, a corresponding hosted domain node must also be created in the DC Tree. The LDIF code for creating a hosted domain in the DC Tree is shown in below. See Appendix A, “Root and Domain ACI Examples“ for ACI information.

Code Example 2-3 LDIF Record for Creating a Hosted Domain Node in the DC Tree

```
dn: dc=sesta,dc=com,o=internet
objectClass: domain
objectClass: inetDomain
objectClass: mailDomain
objectClass: nsManagedDomain
objectClass: icsCalendarDomain
description: DC node for sesta.com hosted domain
dc: sesta
inetDomainBaseDN: o=sesta.com,o=isp
inetDomainStatus: active
mailDomainStatus: active
mailDomainAllowedServiceAccess: +imap, pop3, http:*
mailRoutingHosts: manatee.siroe.com
preferredMailHost: manatee.siroe.com
mailDomainDiskQuota: 100000000
mailDomainMsgQuota: -1
mailClientAttachmentQuota: 5
```

DC Tree Hosted Domain Attributes and Object Classes

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

  The `dn` uniquely identifies the domain entry in the tree. Note that each hosted domain component must match the DNS node for the hosted domain.

- ```
  objectClass: domain
  objectClass: inetDomain
  objectClass: mailDomain
  objectClass: nsManagedDomain
  objectClass: icsCalendarDomain
  ```

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

  `inetDomain` provides attributes for describing the additional properties of a hosted domain. This object class is associated with directory entries which correspond to a DNS domain.

  `mailDomain` represents a hosted domain account and is used in conjunction with `mailDomain` and, optionally, `inetDomainAuthInfo`, for creating hosted domain nodes suitable for mail services for the hosted organization. This object class must be used for all hosted domain entries.

  `nsManagedDomain` stores information for the iPlanet Delegated Administrator for Messaging.

NOTE If your user and group data is stored in the DC Tree and not in an Organization Tree, then you will need to include the `nsManagedDomain` object class and associated attributes. See “Creating a Hosted Domain Entry in the Organizational Tree,” on page 40.

domain Attributes

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

  Free form text description. Usually the full name of the organization that is associated with the value of the attribute `organizationName` for this entry.

- `dc: sesta`

  Required attribute. `dc` is the associated DNS domain component for this node.
  ```

inetDomain Attributes

- `inetDomainBaseDN: o=sesta.com,o=isp`

  Single valued attribute. DN of the sub-tree where all user and group entries for this hosted domain are contained.

- `inetDomainStatus: active`

  Current status of the hosted domain. Valid values: `active`, `inactive`, `deleted`. This attribute is the global domain status. Missing value implies status is `active`. An illegal value is treated as `inactive`. For mail domains, the other status attribute is `mailDomainStatus`.

mailDomain Attributes

- `mailDomainStatus: active`

  Current status of the mail domain. May be one of the following values: `active`, `inactive`, `deleted`, `hold`.

- `mailDomainAllowedServiceAccess: +imap, pop3, http:*`

  Stores service access filters. If no filters are specified, then user is allowed access to all services from all clients.

- `mailRoutingHosts: manatee.siroe.com`

  Fully qualified host name of the MTA responsible for making routing decisions for users in this domain and all its sub-domains. If empty or missing, then all MTAs route user / group messages in this domain and its sub-domains.

- `preferredMailHost: manatee.siroe.com`

  Used by the iPlanet Delegated Administrator for Messaging and Console to set the `mailHost` attribute of users and groups in this mail domain.

- `mailDomainDiskQuota: 100000000`

  Disk quota, in bytes, for all users in the domain. -1 signifies no quota. Though provisioned by the Delegated Administrator, this value is not enforced; however, it may be useful for reporting purposes.

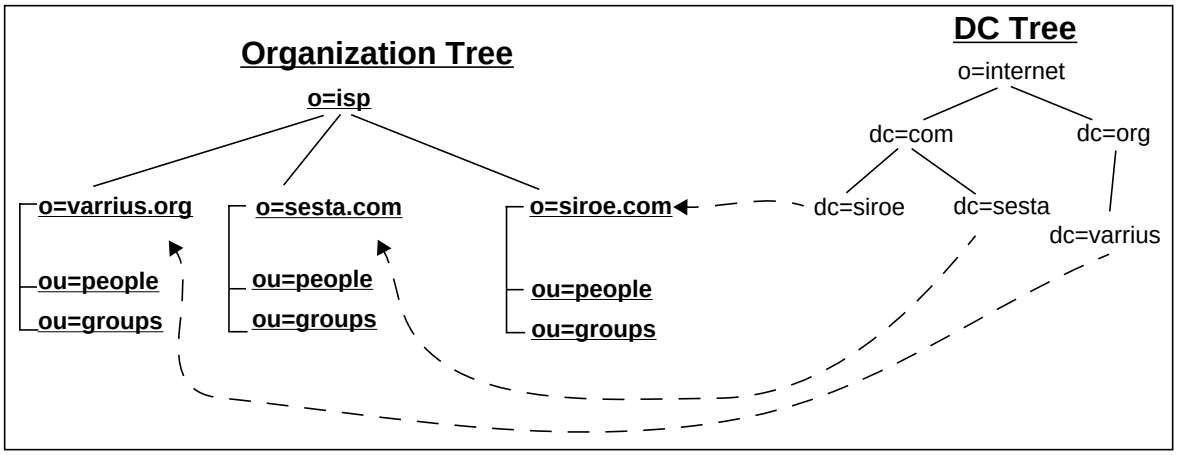
- `mailDomainMsgQuota: -1`

  Quota of number of messages permitted for all users in this domain. -1 signifies no quota. Though provisioned by the Delegated Administrator, this value is not enforced; however it may be useful for reporting purposes.

- `mailClientAttachmentQuota: 5`

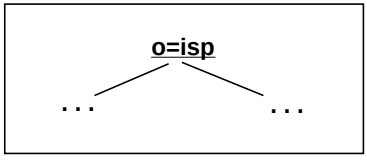
The number of attachments Messenger Express clients are allowed for this domain. -1 means no limit on attachments.

Creating an Organization Tree



The Organization Tree contains the user and group entries. The server references the DC Tree, which points to the Organization Tree. As new hosted domains are created, new domain nodes in the DC and Organization Trees are created. This can be done using the iPlanet Delegated Administrator for Messaging command `imadmin domain create` or the iPlanet Delegated Administrator for Messaging GUI tool.

Creating the Organizational Tree Root Domain Entry



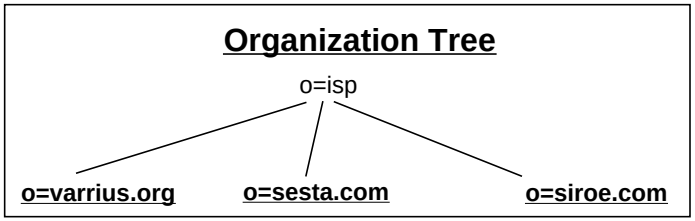
The Organization Tree root entry is created upon installation just like the DC Tree root entry. The organizational tree root name can be specified by the installer. Although this node is created during installation, we show the LDIF for instructional purposes.

Code Example 2-4 LDIF Record for a Organizational Tree Root

```
dn: o=isp
objectClass: organization
o: internet
description: Root level node in the Organizational tree
```

In this example we use `o=isp`, as the root entry DN. For a description of the attributes see “Creating the DC Tree Root Domain Entry,” on page 33. See Appendix A, “Root and Domain ACI Examples“ for ACI information.

Creating a Hosted Domain Entry in the Organizational Tree



Hosted domains in the Organization Tree are created below the root (`o=isp`). The DN of the hosted domain can be any arbitrary name and does not need to reflect the name used in the DC Tree, but the `inetDomainBaseDN` attribute in the domain node of the DC Tree must point to its corresponding node in the Organization Tree. Example LDIF code for creating a hosted domain in the Organization Tree is shown in Code Example 2-5.

Code Example 2-5 Example LDIF Code for a Hosted Domain in the Organization Tree

```
dn: o=sesta.com,o=isp
objectclass: organization
objectclass: nsManagedDomain
o: sesta.com
nsNumMailLists: 5
nsMaxMailLists: 1000
nsNumUsers: 20
nsMaxUsers: 1000
```

Organization Tree Hosted Domain Attributes and Object Classes

This section provides brief descriptions of the hosted domain attributes. For more complete descriptions of the attributes refer to the *iPlanet Schema Reference Manual*.

- `dn: o=sesta.com, o=isp`

  `dn` uniquely identifies the hosted domain entry in the tree.

- `objectClass: organization`
  `objectclass: nsManagedDomain`

  Object classes required by hosted domain entries in the Organization Tree. `organization` is the core object class and is used to describe an organization. `nsManagedDomain` stores information for the iPlanet Delegated Administrator for Messaging.

organization Attributes

- `o: sesta.com`

  `o` is an alias for `organizationName` and is a required attribute.

nsManagedDomain Attributes

- `nsNumMailLists: 5`

  The number of mail lists that have been created by users in the domain. This count is cumulative for all nested sub-domains. The Delegated Administrator maintains this counter and enforces it.

- `nsMaxMailLists: 1000`

  The maximum number of mail lists that users in the domain can create. This limit applies to all nested sub-domains.

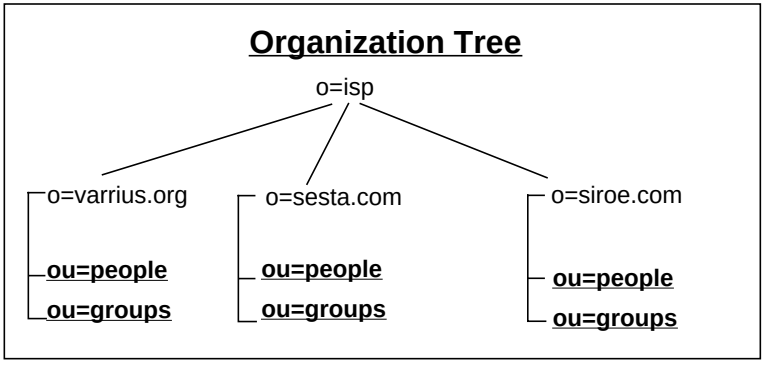
- `nsNumUsers: 20`

The number of user accounts in use in delegated administrator.

- `nsMaxUsers: 1000`

  The maximum number of user accounts that may be created.

Creating the Required Containers for Hosted Domains



Each hosted domain must have a container for user entries called *people*, and a container for mailing list entries called *groups*. Example code is shown below.

Code Example 2-6 LDIF Code for Hosted Domain Containers

```
dn: ou=People,o=sesta.com,o=isp
objectClass: organizationalUnit
ou: People

dn: ou=Groups,o=sesta.com,o=isp
objectClass: organizationalUnit
ou: Groups
```

Organization Tree Hosted Domain Container Attributes and Object Classes

- `dn: ou=People,o=sesta.com,o=isp`
  `dn: ou=Groups,o=sesta.com,o=isp`

  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 mailing list entries for the hosted domain.

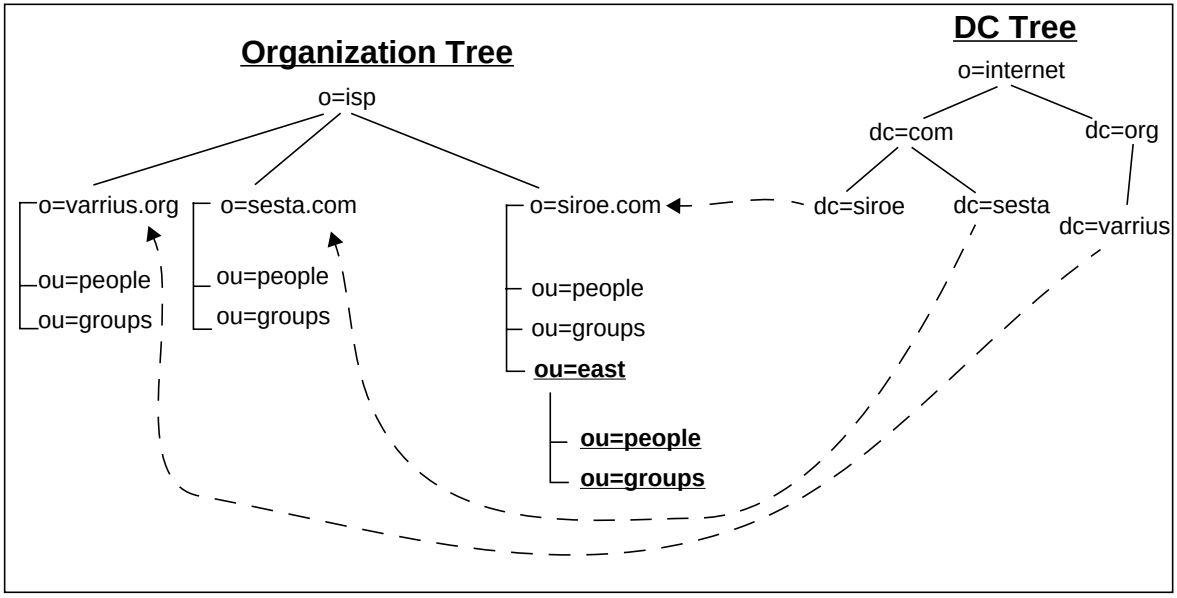
- `objectClass: organizationalUnit`

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

- `ou: People`
  `ou: Groups`

  These are the required `organizationalUnit` entries.

Creating a Domain Organization



A domain organization is a node with users and groups within a domain node (see above). When viewed from the Delegated Administrator for Messaging, a domain organization looks like another organization node. Domain organizations are useful for companies that wish to organize and delegate their user and group entries along departmental or functional lines. When a user entry is put in a domain organization, the user still retains the same email address.

Domain organizations can only be created through LDAP. Currently, there is no mechanism in the iPlanet Console or the iPlanet Delegated Administrator for Messaging to create organizations; however, the Delegated Administrator console does support the creation, deletion and modification of users and groups in existing domain organizations.

A domain organization entry without its ACIs is shown below. (See Appendix A, “Root and Domain ACI Examples“ for information on ACIs.) See the previous section, “Creating the Required Containers for Hosted Domains,” on page 42 for instructions on how to create containers for the domain organization.

Code Example 2-7 LDIF Record for a Domain Organization in the Organization Tree

```
dn: ou=east,o=siroe.com,o=isp
objectclass: nsManagedOrgUnit
objectclass: organizationalUnit
objectclass: inetdomainOrg
ou: east
nsdamodifiableby: cn=Domain Organization Administrators,ou=east,
 o=siroe.com,o=isp
domOrgMaxUsers: 1000
domOrgNumUsers: 3
```

Deleting a Domain Organization

Domain organizations are deleted as follows:

1. Delete all users and groups using the `imadmin delete` commands.
2. Purge all users and groups using the `imadmin purge` commands.
3. Remove all remaining containers such as ou=users and ou=groups.

Creating a Vanity Domain

A vanity domain or custom domain is a domain name attached to individual user entries—not to a domain entry. Vanity domains are useful for individuals or small organizations desiring a customized domain name without the administrative overhead of supporting hosted domain. For example, a small company with six employees called *Florizel* has an email account with Siroe ISP. They would like to receive their email at *florizel.com*. Instead of creating a hosted domain, a vanity domain can be registered and set up so that each member of Florizel will receive mail at *username@florizel.com*.

A vanity domain does not have an LDAP entry for the domain name, but is instead specified on a per user basis by adding the `MailAlternateAddress` attribute (object class `inetLocalMailRecipient`) and the `msgVanityDomain` attribute (object class `msgVanityDomainUser`) in a user entry. An example is shown below.

Code Example 2-8 Example User Entry with Vanity Domain

```
dn: uid=kong,ou=people,o=siroe.com,o=isp
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
objectClass: msgVanityDomainUser
cn: Kelly Kong
sn: Kong
initials: KK
givenName: Kelly
mail: Kelly.Kong@siroe.com
mailAlternateAddress: kelly.kong@florizel.com
mailAlternateAddress: @florizel.com
msgVanityDomain: florizel.com
mailDeliveryOption: mailbox
mailHost: manatee.siroe.com
uid: kong
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: password
mailAllowedServiceAccess: +imap, imaps, pop3, http:*
inetUserStatus: active
mailUserStatus: active
mailQuota: -1
mailMsgQuota: 100
```

In this example, `msgVanityDomain` specifies a domain name used by the MTA for routing purposes. Mail addressed to kelly.kong@siroe.com or kelly.kong@florizel.com will be sent to the message store for Kelly Kong.

The line `mailAlternateAddress: @florizel.com` makes kelly.kong@florizel.com a catch-all domain address, that is, an address that will receive mail to the domain florizel.com if the MTA does not find an exact address match.

NOTE 1) Users with vanity domains must log in to the messaging server using the real domain name. In the example above, Kelly Kong must access her mail through the domain siroe.com and not florizel.com.

2) Vanity domain names must be unique registered internet domain names, and an MX record must be registered for each such name also.

Domain Tasks

This section describes how to perform 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. These attributes are shown in an LDIF update statement and appear in bold in the complete LDIF record. This section consists of the following subsections.

Creating a Domain Alias

A *domain alias* is a domain entry that points to another domain. Domain aliases are created in the DC Tree. A domain aliases must be a unique registered internet domain name. In order for a domain to receive mail, an MX record must be registered for it as well. In the LDIF example below, the domain `florizel.com` can be an alias to `sesta.com`.

Code Example 2-9 Creating a Domain Alias

```
dn: dc=florizel, dc=com, o=internet
objectclass: alias
objectclass: inetDomainAlias
aliasedObjectName: dc=sesta, dc=com, o=internet
dc: florizel
```

Adding a Smart Routing Host for a Domain

A *smart routing host* or *smart host* is an MTA host that is considered to be the authoritative source of routing information for all users in a domain. If a local MTA does not find a user in its local directory, it will forward the message to the smart host. Specify a smart host by adding the fully qualified host name of the routing host to the `mailRoutingSmartHost` attribute of the domain entry. The following LDIF record sets `smarthost1.siroe.com` as the routing host for domain `sesta.com`.

Code Example 2-10 Modify Statement for Adding a Smart Routing Host

```
dn: dc=sesta, dc=com, o=internet
changetype: modify
add: mailRoutingSmartHost
mailRoutingSmartHost: smarthost1.siroe.com
```

Code Example 2-11 LDIF Record for Hosted Domain with Smart Routing Host

```
dn: dc=sesta,dc=com,o=internet
objectClass: domain
objectClass: inetDomain
objectClass: mailDomain
description: DC node for sesta.com hosted domain
dc: sesta
inetDomainBaseDN: o=sesta.com,o=isp
```

Code Example 2-11 LDIF Record for Hosted Domain with Smart Routing Host

```
inetDomainStatus: active
mailDomainStatus: active
mailDomainAllowedServiceAccess: +imap, pop3, http:*
mailRoutingHosts: manatee.sesta.com
preferredMailHost: manatee.sesta.com
mailDomainDiskQuota: 100000000
mailDomainMsgQuota: -1
mailClientAttachmentQuota: 5
mailRoutingSmartHost: smarthost1.siroe.com
```

Adding a New Routing Host for a Domain

A *routing host* is the MTA host that is permitted route mail for addresses in a domain and its sub-domains. A missing `mailRoutingHosts` attribute in a domain record means all MTAs with access to the directory in the system are permitted to route mail for that domain. The example LDIF record below shows how to designate one or more specific MTAs as responsible for mail routing for the domain.

Code Example 2-12 Modify Statement for Adding Routing Hosts

```
dn: dc=sesta, dc=com, o=internet
changetype: modify
add: mailRoutingHosts
mailRoutingHosts: sestarouter1.siroe.com
mailRoutingHosts: sestarouter2.siroe.com
```

Code Example 2-13 LDIF Record for Hosted Domain with Routing Host

```
dn: dc=sesta,dc=com,o=internet
objectClass: domain
objectClass: inetDomain
objectClass: mailDomain
description: DC node for sesta.com hosted domain
dc: sesta
inetDomainBaseDN: o=sesta.com,o=isp
inetDomainStatus: active
mailDomainStatus: active
mailDomainAllowedServiceAccess: +imap, pop3, http:*
mailRoutingHosts: manatee.sesta.com
mailRoutingHosts: sestarouter1.siroe.com
mailRoutingHosts: sestarouter2.siroe.com
preferredMailHost: manatee.sesta.com
```

Code Example 2-13 LDIF Record for Hosted Domain with Routing Host

```
mailDomainDiskQuota: 100000000
mailDomainMsgQuota: -1
mailClientAttachmentQuota: 5
```

Setting a Welcome Message for a Domain

If domains want a welcome message delivered to new users in their domain, they can set the welcome message in the attribute `mailDomainWelcomeMessage`. The following LDIF record sets a welcome message for domain `sesta.com`. A $$ is replaced by a carriage return.

Code Example 2-14 Modify Statement for Adding a Domain Welcome Message

```
dn: dc=sesta, dc=com, o=internet
changetype: modify
add: mailDomainWelcomeMessage
mailDomainWelcomeMessage: Subject: Welcome to Sesta! $$ Welcome!
```

Code Example 2-15 LDIF Record for Hosted Domain with a Welcome Message

```
dn: dc=sesta,dc=com,o=internet
objectClass: domain
objectClass: inetDomain
objectClass: mailDomain
description: DC node for sesta.com hosted domain
dc: sesta
inetDomainBaseDN: o=sesta.com,o=isp
inetDomainStatus: active
mailDomainStatus: active
mailDomainAllowedServiceAccess: +imap, pop3, http:*
mailRoutingHosts: manatee.siroe.com
preferredMailHost: manatee.siroe.com
mailDomainDiskQuota: 100000000
mailDomainMsgQuota: -1
mailClientAttachmentQuota: 5
mailDomainWelcomeMessage: Subject: Welcome to Sesta! $$ Welcome!
```

Limiting Number of Attachments for Messenger Express Clients

To limit the number of attachments a Messenger Express client can send on a single message, set the attribute `mailClientAttachmentQuota` as shown below.

Code Example 2-16 Modify Statement Limiting Messenger Express Attachments to 2 Per Message

```
dn: dc=sesta, dc=com, o=internet
changetype: modify
replace: mailClientAttachmentQuota
mailClientAttachmentQuota: 2
```

Code Example 2-17 LDIF Record for Limiting WebMail Attachments to 2 Per Message

```
dn: dc=sesta,dc=com,o=internet
objectClass: domain
objectClass: inetDomain
objectClass: mailDomain
description: DC node for sesta.com hosted domain
dc: sesta
inetDomainBaseDN: o=sesta.com,o=isp
inetDomainStatus: active
mailDomainStatus: active
mailRoutingHosts: manatee.siroe.com
preferredMailHost: manatee.siroe.com
mailDomainDiskQuota: 100000000
mailDomainMsgQuota: -1
mailClientAttachmentQuota: 2
```

Setting the Domain State

A mail domain can have one of the following states:

- `active` - the mail service in the domain is fully operational.
- `inactive` - the mail service is suspended.
- `deleted` - mail service is suspended, and the domain is marked for deletion by the `imadmin domain purge` command (refer to the *iPlanet Messaging Server Reference Manual*).

- `hold` - tells the MTA to hold all messages for users in that domain in the destination mail server's HOLD queue. This may be used when users are being transitioned from one server to another, and you wish to hold the messages while the user is being moved.

These states are set in the attribute `mailDomainStatus`. States may be changed from one value to another. A missing value implies that the status is `active`. An illegal value is treated as `inactive`. The following LDIF sets the state to `hold`.

Code Example 2-18 Modify Statement Setting Domain State to Hold

```
dn: dc=sesta, dc=com, o=internet
changetype: modify
replace: mailDomainStatus
mailDomainStatus: hold
```

Code Example 2-19 LDIF Record for Hosted Domain on Hold

```
dn: dc=sesta,dc=com,o=internet
objectClass: domain
objectClass: inetDomain
objectClass: mailDomain
description: DC node for sesta.com hosted domain
dc: sesta
inetDomainBaseDN: o=sesta.com,o=isp
inetDomainStatus: active
mailDomainStatus: active
mailRoutingHosts: manatee.sesta.com
mailDomainDiskQuota: 100000000
mailDomainMsgQuota: -1
mailClientAttachmentQuota: 5
mailDomainStatus: hold
```

Chapter 3

Provisioning Family Accounts

A *family account* is a group of email users in the same domain. The family account is under the administrative control of a *family group administrator*, and has one user designated as the *billable user* who is responsible for paying the email account bills for this group of users. The family group administrator is a user responsible for creating and removing users from the group as well as elevating another user to be this family's administrator. *Members* are users who are under the administrative control of the family group administrator and whose email account is paid by the billable user. Family accounts are typically families, but can also be organizational departments where local administrative and billing responsibilities are desired. This chapter contains the following sections:

Creating a Family Account

Family accounts can be managed and created using the `imadmin` command line interface (see the *iPlanet Messaging Server Reference Manual*), the iPlanet Delegated Administrator for Messaging or by using LDAP. This section describes how to provision with LDAP.

Two steps are required for provisioning users in family accounts using LDAP:

1. Create a family account entry (Code Example 3-1).
2. Add the attribute line "`memberOfManagedGroup:` <*FamilyGroupDN*>" to each user entry in the family account (Code Example 3-2).

Code Example 3-1 shows an example of a family account entry.

Code Example 3-1 Family Account Entry

```
dn: cn=gsWarriors, ou=groups,o=sesta.com,o=ISP
objectclass: inetManagedGroup
objectclass: nsManagedDept
mnggrpbillableuser: uid=attles,ou=People,o=sesta.com,o=ISP
mnggrpmailquota: 1024000
mnggrpcurrentusers: 0
mnggrpdeletionpolicy: delete
cn: gsWarriors
mnggrpstatus: active
mnggrpmaxusers: 1000
nsdamodifiableby: cn=Family Group
Administrators,cn=gsWarriors,ou=groups,o=sesta.com,o=ISP
```

- `dn: cn=gsWarriors, ou=groups,o=sesta.com,o=ISP`

  The distinguished name of the family account.

- `objectclass: inetManagedGroup`
  `objectclass: nsManagedDept`

  `inetManagedGroup` represents a family account. `nsManagedDept` stores information used by the iPlanet Delegated Administrator for Messaging.

- `mnggrpbillableuser: uid=attles,ou=People,o=sesta.com,o=ISP`

  DN of the user who is responsible for paying the bills for this group of users.

- `mnggrpmailquota: 1024000`

  Cumulative disk quota allowed for all users in the group.

- `mnggrpcurrentusers: 0`

  Current count of users in the group.

- `cn: gswarriors`

  Common name of the family account.

- `mnggrpstatus: active`

  Current status of the group—`active`, `inactive` or `deleted`. `inactive` temporarily suspends operation. `deleted` marks the entry for deletion, but does not mark the users for deletion. Missing value implies status is `active`. An illegal value is treated as `inactive`.

- `mnggrpmaxusers: 1000`

  Number of users allowed in the group.

- `nsdamodifiableby: cn=Domain Organization Adminstrators, cn=gsWarriors,ou=groups,o=sesta.com,o=ISP`

  Specifies the groups whose members can administer this family group. Refer to “Creating a Family Group Administrator,” on page 55.

Once the family account entry is created, members are added by setting the `memberOfManagedGroup` attribute in the user’s entry to the family account DN. An example is shown below.

Code Example 3-2 LDIF Record for a Member of a Family Group

```
dn: uid=Antwan,ou=people,o=sesta.com,o=ISP
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
cn: Antwan
sn: James
initials: AJ
givenName: Ant
mail: aj@sesta.com
mailAlternateAddress: ant@sesta.com
mailDeliveryOption: mailbox
mailHost: manatee.siroe.com
uid: Antwan
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: {SHA}aluWfd0LYY9ImsJb3h4afrI4AXk=
mailAllowedServiceAccess: +imap, imaps, pop3, smtp, http:*
inetUserStatus: active
mailUserStatus: active
mailQuota: -1
mailMsgQuota: 100
memberOfManagedGroup: cn=gsWarriors, ou=groups,o=sesta.com,o=ISP
```

Creating a Family Group Administrator

Delegated Admin Task Utility: `imadmin family-admin add`
Task Privilege: Top-level Administrator, Domain Administrator, or Family Group Administrator

The Family Group Administrator is a user responsible for creating and removing users from the family group. When a family group is created with the Delegated Administrator, another group called Family Group Administrators is also created below the DN of the family group entry. For example, if the family group is

`cn=gsWarriors,ou=groups,o=sesta.com,o=isp`

then a Family Administrator's group is also created. Its DN is:

```
cn=Family Group
Administrators,cn=gsWarriors,ou=groups,o=sesta.com,o=isp
```

Members of this group have administrative privileges for the family group. The example below demonstrates how to provision a Family Group Administrator.

1. Make sure a group called Family Group Administrators is created below the DN of the family group entry and add the DN of the Family Group Administrator. This is automatically created when a Family Group is created with the Delegated Administrator.

Code Example 3-3 Family Administrator Group Entry

```
dn: cn=Family Group Administrators,cn=gsWarriors,ou=groups,o=sesta.com,o=isp
objectclass: groupOfUniqueNames
objectclass: nsManagedDept
cn: Organization Administrators
nsNumUsers: 1
nsMaxUsers: Unlimited
uniqueMember: uid=Dave,ou=people,ou=sesta.com,o=isp
```

2. Add `memberof` attribute to the new Family Group Administrator's entry:

Code Example 3-4 Entry for a Family Group Administrator

```
dn: uid=Dave,ou=people,o=sesta.com,o=isp
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
cn: Dave Cowins
sn: cowins
initials: DC
```

Code Example 3-4 Entry for a Family Group Administrator

```
givenName: Dave
mail: Dave.Cowins@sesta.com
mailAlternateAddress: dcowins@florizel.com
mailDeliveryOption: mailbox
mailHost: manatee.siroe.com
uid: Dave
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: {SHA}aluWfd0LYY9ImsJb3h4afrI4AXk=
mailAllowedServiceAccess: +imap, imaps, pop3, http:*
inetUserStatus: active
mailUserStatus: active
mailQuota: -1
mailMsgQuota: 100
memberOf: cn=Family Group Administrators,cn=gsWarriors,ou=groups,o=sesta.com,o=isp
```

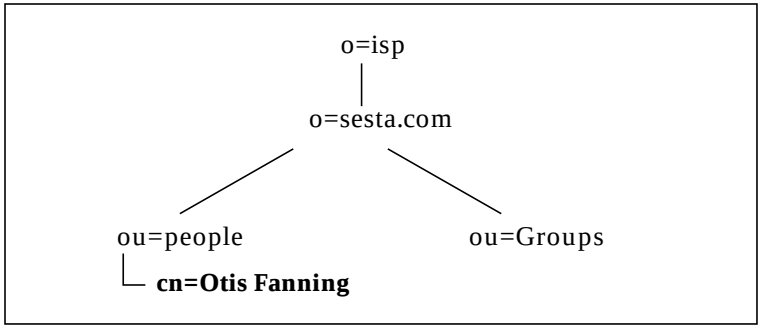
3. ACIs are set at top level on root suffix. See Appendix A, “Root and Domain ACI Examples."

Chapter 4

Provisioning Users

This section describes how to create user entries for the iPlanet Messaging Server. As installed, user entries are created in the Organization Tree and our examples will reflect this. Note that attribute descriptions in this guide are overviews. Full attribute descriptions are available in the *iPlanet Schema Reference Manual*. This chapter consists of the following sections:

Creating User Entries



sesta.com email users are in `ou=people,o=sesta.com,o=isp` in the Organization Tree. Each entry contains attribute information about the user. An LDIF example is shown below.

Code Example 4-1 Example User Entry

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
cn: Otis Fanning
sn: fanning
initials: OTF
givenName: Otis
pabURI: ldap://ldap.siroe.com:389/ou=fanning,ou=people,o=sesta.com,o=isp,o=pab
mail: Otis.Fanning@sesta.com
mailAlternateAddress: ofanning@sesta.com
mailDeliveryOption: mailbox
mailHost: manatee.siroe.com
uid: fanning
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: password
inetUserStatus: active
mailUserStatus: active
mailQuota: -1
mailMsgQuota: 100
```

User Entry Object Classes and Attributes

- `dn: uid=fanning,ou=people,o=sesta.com,o=ISP`

  The distinguished name of the user Otis Fanning.

- ```
  objectClass: person
  objectClass: organizationalPerson
  objectClass: inetOrgPerson
  objectClass: inetUser
  objectClass: ipUser
  objectClass: inetMailUser
  objectClass: inetLocalMailRecipient
  objectClass: nsManagedPerson
  objectClass: userPresenceProfile
  ```

  These are the user entry object classes. `inetOrgPerson` is the core object class, and it inherits from `organizationalPerson` and `person` (these two classes are shown for clarification even though it is not necessary to include them in an LDAP add statement if you are provisioning).

  `inetUser`, `inetMailUser` and `ipUser` provide attributes for creating a mail user account. They provide attributes such as UID, user password, public address book information, and mail delivery information.

  `inetLocalMailRecipient` provides local routing information.

  `userPresenceProfile` supports vacation attributes.

  `nsManagedPerson` provides iPlanet Delegated Administrator for Messaging support attributes.

inetOrgPerson

- `cn: Otis Fanning`

  `cn` (`commonname`) is the user's full name and is inherited from the `person` object class.

- `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@sesta.com`

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

inetUser

- `uid: fanning`

  `uid` is the user's login identification. This attribute must be unique within the domain.

- `userPassword: secret`

  `userPassword` is the string representing the user's password for accessing his account to read and send mail.

- `inetUserStatus: active`

  Global status of the user for all directory-enabled server. Values: `active`, `inactive`, `deleted`. Missing value implies status is `active`. An illegal value is treated as `inactive`.

ipUser

This core object class supports internet services like mail and calendar. It provides attributes for the personal address book store and class of service features.

- `pabURI: ldap://ldap.siroe.com:389/ou=fanning,ou=people, o=sesta.com,o=isp,o=pab`

  LDAP URI specifying the container of the personal address book entries for this user. Typically this is set by the server when the user creates personal address book entries; however if you want to force the server to use a different LDAP server for the address book, then you would use this attribute and provision the appropriate values.

inetMailUser

- `dataSource: iPlanet Messaging Server 5.0 @(#)ims50users.sh`

  Text field to store a tag or identifier.

- `mailDeliveryOption: mailbox`

Mail delivery option(s) to be used for the recipient. `mailDeliveryOption` can be multi-valued. Possible values are `mailbox`, `native|unix`, `autoreply`, `program`, and `forward`.

- `mailUserStatus: active`

  Single valued attribute. Stores one of the following mail user states: `active`, `inactive`, `deleted`, `hold`. `hold` specifies that all mail for the user's inbox is sent to the hold queue and all access to the mailbox over IMAP and POP is disallowed. Missing value implies status is `active`. An illegal value is treated as `inactive`.

- `mailQuota: -1`

  Disk space allowed for the user's mailbox. Value of 0 (or not specified) means system default quota (defined through iPlanet Console) and -1 means no limit on space usage.

- `mailMsgQuota: 100`

  Maximum number of messages permitted for a user. This is a cumulative count for all folders in the store. Value of 0 (or not specified) means system default quota and -1 (defined through iPlanet Console or iPlanet Delegated Administrator for Messaging) means no limit on number of messages.

inetLocalMailRecipient

- `mailHost: manatee.siroe.com`

  Fully-qualified hostname of the MTA that is the final SMTP destination of messages to this group.

- `mailAlternateAddress: ofanning@sesta.com`

  Alternate RFC 822 e-mail address of this user.

userPresenceProfile

Supports vacation attributes. See *iPlanet Schema Reference Manual* for details.

nsManagedPerson

Supports attributes for iPlanet Delegated Administrator for Messaging.

Mail User Tasks

This section consists of the following subsections:

Activating/Deactivating Users

The iPlanet Messaging Server honors two different status attributes for user activation / deactivation, `inetUserStatus` and `mailUserStatus`. The `inetUserStatus` attribute holds the global user status for all services (mail, calendar, etc.) and overrides the status set in `mailUserStatus`. `mailUserStatus` sets the status for mail service only. It may have the values `active`, `inactive`, or `deleted`. Setting `inetUserStatus` to `deleted` marks the entry for deletion during next purge cycle.

`mailUserStatus` sets the status for individual users with one of the following mail user states: `active`, `inactive`, `deleted`, `hold`. `mailUserStatus=hold` implies ALL mail for the users inbox is sent to the hold queue and all access to the mailbox over IMAP and POP is disallowed. Setting `mailUserStatus` to `delete` marks the entry for removal.

The following LDIF update statement disables a user’s ability to use any services:

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
changetype: modify
replace: inetUserStatus
inetUserStatus: inactive
```

The following LDIF update statement disables only the mail service for that user. If other services, like calendar, are provisioned for that user, they continue to be active.

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
changetype: modify
replace: mailUserStatus
mailUserStatus: inactive
```

The following update statement redirects messages for the user into the hold queue. This may be used when moving users from one mail server to another. Inbound messages for users are redirected to the hold queue. To start mail delivery for the user again, `mailUserStatus` must be set back to `active`.

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
changetype: modify
replace: mailUserStatus
mailUserStatus: hold
```

Changing a User Password

You can change the user's password by changing the `userPassword` attribute. This can be changed with the following LDIF:

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
changetype: modify
replace: userPassword
userPassword: PAssWoRd
```

Code Example 4-2 LDIF Record after Changing a User's Password

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
cn: Otis Fanning
sn: fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@sesta.com
mailAlternateAddress: ofanning@sesta.com
mailDeliveryOption: mailbox
```

Code Example 4-2 LDIF Record after Changing a User's Password

```
mailHost: manatee.siroe.com
uid: fanning
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: PAssWoRd
inetUserStatus: active
mailUserStatus: active
mailQuota: -1
mailMsgQuota: 100
```

Setting a User Vacation Message

Vacation messages are set using attributes from object class `userPresenceProfile` (attributes: `vacationStartDate` and `vacationEndDate`) and `inetMailUser` (attributes: `mailAutoreplyMode`, `mailAutoreplyTimeout`, `mailAutoreplySubject`, `mailAutoreplyText`, `mailAutoreplyTextInternal` and `mailDeliveryOption`).

Please refer to the *iPlanet Schema Reference Manual* for the detailed semantics for each attribute.

The example LDIF file below sets a vacation message for a user starting 15th February 2001 and ending 20th February 2000, with auto reply time-out of 2 days and a vacation message subject of *I am on vacation* and a body text (both internal to the same domain and external to all other users) of *Please contact me later*.

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
changetype: modify
add: mailDeliveryOption
mailDeliveryOption: autoreply
-
replace: mailAutoreplyMode
mailAutoreplyMode: reply
-
replace: vacationStartDate
vacationStartDate: 20010215000000Z
-
replace: vacationEndDate
vacationEndDate: 20010220000000Z
-
replace: mailAutoreplyTimeout
mailAutoreplyTimeout: 48
-
replace: mailAutoreplySubject
mailAutoreplySubject: I am on vacation
```

```
-
replace: mailAutoreplyTextInternal
mailAutoreplyTextInternal: Please contact me later.
-
replace:  mailAutoreplyText
mailAutoreplyText: Please contact me later.
```

Note that the date settings are Universal Time Coordinated (UTC), the international time standard formerly known as Greenwich Mean Time, or GMT. If you are in a different time zone, you need to set the UTC equivalent time. For example, subtract 8 hours if you want to set the vacation to midnight 2/15/2001 Pacific Standard Time.

Code Example 4-3 LDIF Record after Setting and Activating a User's Auto-reply Vacation Message

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
cn: Otis Fanning
sn: fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@sesta.com
mailAlternateAddress: ofanning@sesta.com
mailDeliveryOption: mailbox
mailHost: manatee.siroe.com
uid: fanning
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: password
inetUserStatus: active
mailUserStatus: active
mailQuota: -1
mailMsgQuota: 100
mailDeliveryOption: autoreply
mailAutoreplyMode: reply
vacationStartDate: 20010215000000Z
vacationEndDate: 20010220000000Z
mailAutoreplyTimeout: 48
mailAutoreplySubject: I am on vacation
mailAutoreplyTextInternal: Please contact me later.
mailAutoreplyText: Please contact me later.
```

Adding/Removing Allowed Mail Services

Allowed mail services are set with the `mailAllowedServiceAccess` attribute. The allowed values are `imap`, `imaps`, `pop3`, `http`. The format is as follows:

`mailAllowedServiceAccess: +<`*Allowed,Services*`>:*`

The following LIDF modify code adds IMAP, secure IMAP, POP3 and HTTP (Messenger Express) support for Otis Fanning.

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
changetype: modify
replace: mailAllowedServiceAccess
mailAllowedServiceAccess: +imap, imaps, pop3, http:*
```

Code Example 4-4 LDIF Record after Changing a User's Mail Services

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
cn: Otis Fanning
sn: fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@sesta.com
mailAlternateAddress: ofanning@sesta.com
mailDeliveryOption: mailbox
mailHost: manatee.siroe.com
uid: fanning
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: password
mailAllowedServiceAccess: +imap, imaps, pop3, http:*
inetUserStatus: active
mailUserStatus: active
mailQuota: -1
mailMsgQuota: 100
```

Adding or Changing Incoming Mail Delivery Options

The `mailDeliveryOption` attribute controls whether incoming messages are sent to the message store (`mailbox`), `/var/mail` (`native`), to an autoreply facility (`autoreply`), forwarded to another mail address (`forward`), or to an approved mail processing program (`program`). Note that this attribute can take multiple values. If `mailDeliveryOption` is set to `program`, then `mailProgramDeliveryInfo` has to be set (see the *iPlanet Message Server Administration Guide* for details on program delivery).

The LDIF data below sends messages to `/var/mail` as well as the message store.

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
changetype: modify
add: mailDeliveryOption
mailDeliveryOption: native
```

Code Example 4-5 LDIF Record for Changing a User's Mail Delivery Options

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
cn: Otis Fanning
sn: fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@sesta.com
mailAlternateAddress: ofanning@sesta.com
mailDeliveryOption: native
mailDeliveryOption: mailbox
mailHost: manatee.siroe.com
uid: fanning
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: password
inetUserStatus: active
mailUserStatus: active
mailQuota: -1
mailMsgQuota: 100
```

Setting User Message Filters

The iPlanet Messaging Server supports the SIEVE mail filtering language for creating message filters that can block, store, or redirect mail messages containing a specified string or exceeding specified lengths. Users can specify filtering rules and actions through the iPlanet Delegated Administrator for Messaging and also through LDAP using the attribute `mailSieveRuleSource`. Note, however, that message filters created by adding the `mailSieveRuleSource` with LDAP cannot be modified using iPlanet Delegated Administrator for Messaging. The iPlanet Delegated Administrator for Messaging can view and delete these LDAP created message filters, but it cannot edit them. The iPlanet Messaging Server supports the SIEVE rules as specified in the Internet draft "Sieve: A Mail Filtering Language", T. Showalter, draft 9 (http:/ / docs.iplanet.com / docs / sieve). Below is an example of a mail filter using SIEVE.

NOTE The iPlanet Delegated Administrator for Messaging supplies templates of the most common useful mail filters. For information on creating or modifying the iPlanet Delegated Administrator for Messaging mail filters and templates, refer to the *iPlanet Message Server Administration Guide*.

In this example, the value of `mailSieveRuleSource` is stored in a file and the value is encoded by entering the `-b` flag in the `ldapmodify` command.

- The modify entry file `modfanning.ldif` contains the following:

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
changetype: modify
add: mailSieveRuleSource
mailSieveRuleSource:  /export/example/mailsievesource.txt
```

- `mailsievesource.txt` contains the following:

```
require ["fileinto", "reject"];
if header :contains "Subject" "New Rules Suggestion"
   {redirect "rules@sesta.com" # Forward message }
elsif header :contains "Sender" "porn.com"
   {discard text:
Your message has been rejected. Please remove this address from
your mailing list.    # Reject message, send reply message
.
    ;}
elsif size :over 1M
   { reject text:
Please do not send me large attachments.
```

```
Put your file on a server and send me the URL.
Thank you. # Discard message, send reply message
.
    ;}
elsif header :contains "Sender" "barkley@sesta.com"
   { fileinto "complaints.refs"; # file message}
```

- The modify command is as follows:

```
ldapmodify -D "cn=Directory Manager" -w password -b -f
modfanning.ldif
```

Code Example 4-6 LDIF Record after Setting a User's Message Filters (displayed using ldapsearch with -o flag)

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
cn: Otis Fanning
sn: fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@sesta.com
mailAlternateAddress: ofanning@sesta.com
mailDeliveryOption: mailbox
mailHost: manatee.siroe.com
uid: fanning
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: password
inetUserStatus: active
mailUserStatus: active
```

Code Example 4-6 LDIF Record after Setting a User's Message Filters (displayed using ldapsearch with -o flag)

```
mailQuota: -1
mailMsgQuota: 100
mailSieveRuleSource:
require ["fileinto", "reject"];
if header :contains "Subject" "New Rules Suggestion"
   {redirect "rules@sesta.com" # Forward message }
elsif header :contains "Sender" "porn.com"
   {discard text:
Your message has been rejected. Please remove this address from
your mailing list.    # Reject message, send reply message
.
    ;}
elsif size :over 1M
   { reject text:
Please do not send me large attachments.
Put your file on a server and send me the URL.
Thank you. # Discard message, send reply message
.
    ;}
elsif header :contains "Sender" "barkley@sesta.com"
   { fileinto "complaints.refs"; #file message}
```

Mail and Message Quotas

A user's mailbox disk space quota is set by specifying the `mailQuota` attribute. A setting of 0 or no value means system default quota (set via the iPlanet Console or the variable `store.defaultmailboxquota`) and -1 means no limit on space usage.

The maximum number of messages a user is permitted is set with `mailMsgQuota`. Value of 0 (or not specified) means system default quota (set via the iPlanet Console or the variable `store.defaultmessagequota`) and -1 means no limit on number of messages.

The following LDIF modify code sets this user's mail quota to unlimited space and message quota to 2000.

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
changetype: modify
replace: mailQuota
mailQuota: -1
-
replace: mailMsgQuota
mailMsgQuota: 2000
```

Code Example 4-7 LDIF Record for Setting a User's Mailbox and Message Quota

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
cn: Otis Fanning
sn: fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@sesta.com
mailAlternateAddress: ofanning@sesta.com
mailDeliveryOption: mailbox
mailHost: manatee.siroe.com
uid: fanning
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: password
inetUserStatus: active
mailUserStatus: active
mailQuota: -1
mailMsgQuota: 2000
```

Mail Forwarding

To forward mail set `mailDeliveryOption` to `forward` and set `mailForwardingAddress` to the desired email address. `mailForwardingAddress` can have multiple values. The following example sends mail to the user's mailbox as well as forwards mail to `fastjonny@varrius.com` and `John@florizel.net`. Note that the message will also be delivered to the users inbox because `mailDeliveryOption` is also set to `mailbox`.

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
changetype: modify
add: mailDeliveryOption
mailDeliveryOption: forward
-
replace: mailForwardingAddress
mailForwardingAddress: fastjonny@varrius.com
mailForwardingAddress: John@florizel.net
```

Code Example 4-8 LDIF Record for Adding User Mail Forwarding Addresses

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
cn: Otis Fanning
sn: fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@sesta.com
mailAlternateAddress: ofanning@sesta.com
mailDeliveryOption: mailbox
mailDeliveryOption: forward
mailForwardingAddress: fastjonny@varrius.com
mailForwardingAddress: john@florizel.com
mailHost: manatee.siroe.com
uid: fanning
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: password
inetUserStatus: active
mailUserStatus: active
mailQuota: -1
mailMsgQuota: 100
```

New Mail Aliases

A user can have multiple alternate email addresses for a mail account by setting these alternate names to `mailAlternateAddress`. A catch-all mail address (an address that will receive any message to an address in a domain if the MTA does not find an exact match for the address) can be created by setting the attribute to `@domain_name>.com`.

The modify statement below adds an alternative address for Otis Fanning.

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
changetype: modify
replace: mailAlternateAddress
mailAlternateAddress: ofanning@sesta.com
```

Code Example 4-9 LDIF Record for Adding User Mail Aliases

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
cn: Otis Fanning
sn: fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@sesta.com
mailAlternateAddress: ofanning@sesta.com
mailDeliveryOption: mailbox
mailHost: manatee.siroe.com
uid: fanning
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: password
inetUserStatus: active
mailUserStatus: active
mailQuota: -1
mailMsgQuota: 100
```

Changing A User's Mail Server

Change the user's mail server (fully-qualified hostname of the MTA that is the final SMTP destination of messages to this recipient) by specifying the new server to `mailhost`.

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
changetype: modify
replace: mailhost
mailhost: buffalo.siroe.com
```

Code Example 4-10 LDIF Record for Changing a User's Mail Server

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
```

Code Example 4-10 LDIF Record for Changing a User's Mail Server

```
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
cn: Otis Fanning
sn: fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@sesta.com
mailAlternateAddress: ofanning@sesta.com
mailDeliveryOption: mailbox
mailHost: buffalo.siroe.com
uid: fanning
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: password
inetUserStatus: active
mailUserStatus: active
mailQuota: -1
mailMsgQuota: 100
```

Adding Mailing List Creation Privileges

Add the attribute value-pair `nsDACapability: mailListCreate` to a user's entry to allow mailing list creation.

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
changetype: modify
add: nsDACapability
nsDACapability: yes
```

Code Example 4-11 LDIF Record for Changing a User's Mail Server

```
dn: uid=fanning,ou=people,o=sesta.com,o=ISP
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
cn: Otis Fanning
sn: fanning
initials: OTF
givenName: Otis
mail: Otis.Fanning@sesta.com
mailAlternateAddress: ofanning@sesta.com
```

Code Example 4-11 LDIF Record for Changing a User's Mail Server

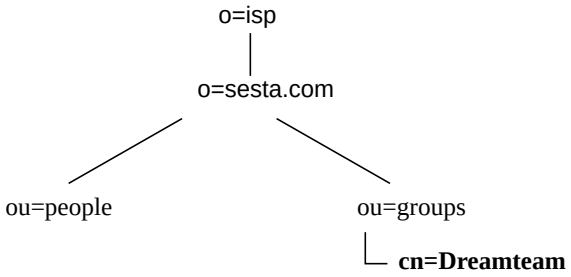
```
mailDeliveryOption: mailbox
mailHost: manatee.siroe.com
nsDACapability: mailListCreate
uid: fanning
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: password
inetUserStatus: active
mailUserStatus: active
mailQuota: -1
mailMsgQuota: 100
```

Chapter 5

Provisioning Mailing Lists

This section describes how to create mailing list entries in the iPlanet Messaging Server. As installed, mailing list entries are created in the Organization Tree. If you system is using a SIMS still DC Tree, then mailing list entries will be in the DC Tree. Note that attribute descriptions in this guide are overviews. Full attribute descriptions are available in the *iPlanet Schema Reference Manual*.

Creating Mailing List Entries



Mailing list entries are created in the ou=group containers of the organization tree. An example LDIF record for a mailing list is shown below.

Code Example 5-1 LDIF Record for a Mailing List

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
cn: Dreamteam
objectClass: groupOfUniqueNames
objectClass: inetMailGroup
objectClass: inetLocalMailRecipient
objectClass: inetMailGroupManagement
objectClass: nsManagedMailList
uniqueMember: uid=baylor,ou=People,o=sesta.com,o=isp
uniqueMember: uid=bird,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jordan,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jabbar,ou=People,o=sesta.com,o=isp
uniqueMember: uid=magic,ou=People,o=sesta.com,o=isp
mgrpRFC822MailMember: west@florizel.com
mgrpRFC822MailMember: robertson@florizel.com
mail: dreamteam@sesta.com
mailAlternateAddress: thegreatest@sesta.com
mailHost: manatee.siroe.com
dataSource: @(#)ims50users.sh 1.5a 02/3/00
inetMailGroupStatus: active
nsNumUsers: 7
nsMaxUsers: 1000
```

Mail List Attributes

- `dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp`

  The distinguished name of the mailing list.

- `objectClass: groupOfUniqueNames`
  `objectClass: inetMailGroup`
  `objectClass: inetLocalMailRecipient`
  `objectClass: inetMailGroupManagement`
  `objectClass: nsManagedMailList`

  `groupOfUniqueNames` is the core object class for all mailing list entries. Overlaying with the mail service object classes `inetMailGroup`, `inetLocalMailRecipient`, and the Delegated Administrator service object classes `inetMailGroupManagement` makes the entry a mailing list for use by the messaging server.

  `inetMailGroup` specifies attributes for mailing lists.

  `inetLocalMailRecipient` provides internal routing attributes. This object class is intended to support SMTP message transfer agents in routing RFC 822-based email within a private enterprise only and is not to be used in the process of routing email across the public Internet.

  `inetMailGroupManagement` specifies attributes for managing a mailing list.

  `nsManagedMailList` provides iPlanet Delegated Administrator for Messaging support attributes for mailing lists.

groupOfUniqueNames

- `cn: Dreamteam`

  `cn` (`commonname`) is the mailing list's name.

- `uniqueMember: uid=baylor,ou=People,o=sesta.com,o=isp`
  `uniqueMember: uid=bird,ou=People,o=sesta.com,o=isp`
  `uniqueMember: uid=jordan,ou=People,o=sesta.com,o=isp`
  `uniqueMember: uid=jabbar,ou=People,o=sesta.com,o=isp`
  `uniqueMember: uid=magic,ou=People,o=sesta.com,o=isp`

  These are the members of the mailing list that can be resolved to a user in this directory.

inetMailGroup

- `inetMailGroupStatus: active`

Current status of the mailing list: `active`, `inactive` or `deleted`. Missing value implies status is `active`. An illegal value is treated as `inactive`.

- `dataSource: @(#)ims50users.sh 1.5a 02/3/00`

  Text field to store a tag or identifier.

- `mgrpRFC822MailMember: west@florizel.com`
  `mgrpRFC822MailMember: robertson@florizel.com`

  External members of the mailing list.

inetLocalMailRecipient

- `mail: dreamteam@sesta.com`

  The mailing list's advertised email address (RFC 822 format).

- `mailAlternateAddress: thegreatest@sesta.com`

  Alternate RFC822 email address of this mailing list.

- `mailHost: manatee.siroe.com`

  Fully-qualified hostname of the MTA server that is the final SMTP destination of messages to this mailing list.

nsManagedMailList

- `nsNumUsers: 7`

  Current number of user entries.

- `nsMaxUsers: 1000`

  Maximum user entries.

Format of Attribute Values

There are several attributes such as `moderator` and `mailDeliveryURL` that require user addresses or filenames specified as URLs. 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:///uid=cox,ou=people,o=sesta,o=isp
```

Note that if attribute-value pairs span two lines, the second line must start with a blank space.

When preceded by a `mailto: ofanning@sesta.com` the entry is interpreted as a mail address.

When preceded by `file:///` the entry is interpreted as a file. Example:

```
mailDeliveryURL: file:///home/dogboy/dr_j/mail_archive.htm
```

Mailing List Tasks

Assigning Mailing List Owners

Mailing list owners can add or delete members to the list. To change an owner of a mailing list, assign a DN to the `owner` attribute. There can be more than one owner for the mailing list, but owners must have DNs in the same directory as the mailing list. An example modification statement and LDIF record is shown below.

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
changetype: modify
add: owner
owner:  uid=baylor,ou=People,o=sesta.com,o=isp
```

Code Example 5-2 LDIF Record for a Mailing List with an Owner

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
cn: Dreamteam
objectClass: groupOfUniqueNames
objectClass: inetMailGroup
objectClass: inetLocalMailRecipient
objectClass: inetMailGroupManagement
objectClass: nsManagedMailList
uniqueMember: uid=baylor,ou=People,o=sesta.com,o=isp
uniqueMember: uid=bird,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jordan,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jabbar,ou=People,o=sesta.com,o=isp
uniqueMember: uid=magic,ou=People,o=sesta.com,o=isp
mgrpRFC822MailMember: west@florizel.com
mgrpRFC822MailMember: robertson@florizel.com
mail: dreamteam@sesta.com
mailAlternateAddress: thegreatest@sesta.com
mailHost: manatee.siroe.com
dataSource: @(#)ims50users.sh 1.5a 02/3/00
inetMailGroupStatus: active
nsNumUsers: 7
nsMaxUsers: 1000
owner:  uid=baylor,ou=People,o=sesta.com,o=isp
```

Adding Members

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 `mgrpRFC822MailMember`. The example LDIF code below shows how to add an internal and external user.

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
changetype: modify
add: uniqueMember
uniqueMember:  uid=russell,ou=People,o=sesta.com,o=isp
-

add: mgrpRFC822MailMember
mgrpRFC822MailMember:  chamberlain@varrius.com
```

Code Example 5-3 LDIF Record for a Mailing List with Added Members

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
cn: Dreamteam
objectClass: groupOfUniqueNames
objectClass: inetMailGroup
```

Code Example 5-3 LDIF Record for a Mailing List with Added Members

```
objectClass: inetLocalMailRecipient
objectClass: inetMailGroupManagement
objectClass: nsManagedMailList
uniqueMember: uid=baylor,ou=People,o=sesta.com,o=isp
uniqueMember: uid=bird,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jordan,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jabbar,ou=People,o=sesta.com,o=isp
uniqueMember: uid=magic,ou=People,o=sesta.com,o=isp
mgrpRFC822MailMember: west@florizel.com
mgrpRFC822MailMember: robertson@florizel.com
uniqueMember:  uid=russell,ou=People,o=sesta.com,o=isp
mgrpRFC822MailMember:  chamberlain@varrius.com
mail: dreamteam@sesta.com
mailAlternateAddress: thegreatest@sesta.com
mailHost: manatee.siroe.com
dataSource: @(#)ims50users.sh 1.5a 02/3/00
inetMailGroupStatus: active
nsNumUsers: 7
nsMaxUsers: 1000
owner:  uid=baylor,ou=People,o=sesta.com,o=isp
```

Creating Posting Restrictions on Mailing Lists

Incoming mail to a mailing list can be restricted by domain or user. The restriction attributes are as follows:

- `mgrpAllowedBroadcaster` specifies addresses authorized to send messages to the mailing list. If not included in the LDAP entry, the list is unrestricted and anyone can submit. The envelope `From:` address must match one of the addresses in the permitted list before the MTA will route the message to a list of members.
- `mgrpDisallowedBroadcaster` specifies addresses restricted from posting 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.
- `mgrpAllowedDomain` specifies the domain names from which users are authorized to post messages to the mailing list. The wildcard character is "*". Using the wildcard character you may optionally replace a sub-domain to authorize the entire DNS hierarchy below a given top or sub-domain.

- `mgrpDisallowedDomain` defines the domain names from which users cannot post messages to the mailing list.

NOTE DN values for `mgrpAllowedBroadcaster`, `mgrpDisallowedBroadcaster` must have the prefix `ldap:///` or `mailto:` Refer to “Format of Attribute Values,” on page 82.

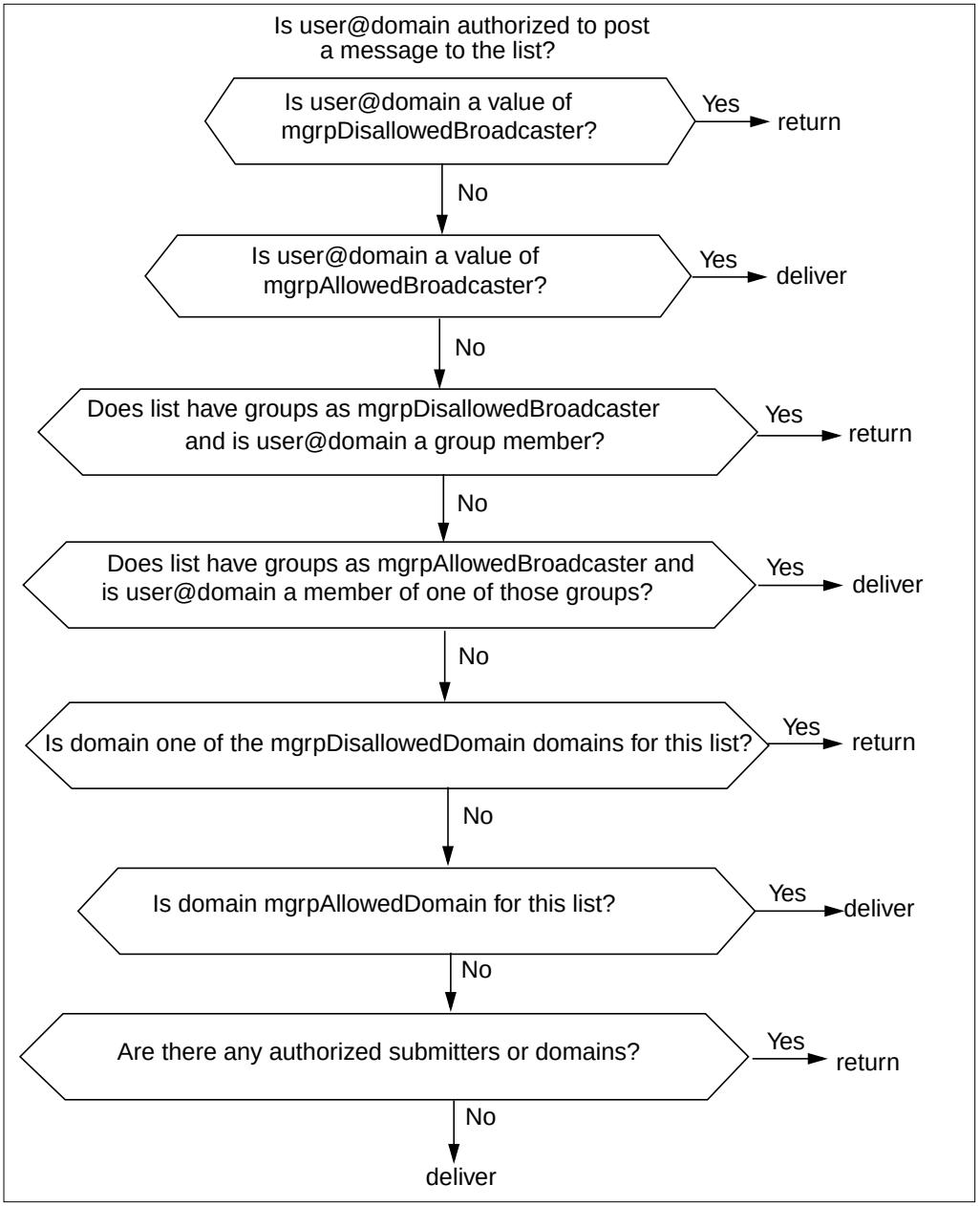
Precedence Rules

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

1. If `mgrpDisallowedBroadcaster` is set, there must not be a match between this value and the sender's `mail` attribute or `mailAlternateAddress` attribute of any DN listed in the form of a `ldap:///`<*DN*> address, or there must not be a match with the RFC-822 address listed in the form of a `mailto:`<*RFC-822*> address.
2. If `mgrpAllowedBroadcaster` attribute exists in the LDAP entry, the sender's address must match either the `mail` attribute or `mailAlternateAddress` attribute of any DN listed in the form of a `ldap:///`<*DN*> address or must match the RFC-822 address listed in the form of a mailto:<*RFC-822*> address.
3. If `mgrpDisallowedDomain` exists in the LDAP entry, then sender's domain must not match the domain(s) listed in the `mgrpDisallowedDomain` attribute.
4. If `mgrpAllowedDomain` attribute exists in the LDAP entry, then the sender's domain must match the domain(s) listed in the `mgrpAllowedDomain` attribute.

The diagram below shows the access control process.

Figure 5-1 Access Control Process



The following LDIF code allows users on the domain sesta.com to send messages to the mail list, but blocks users on all other domains to send messages. It also blocks internal mail from barry@sesta.com but allows mail from barkley@florizel.com.

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
changetype: modify
add: mgrpAllowedDomain
mgrpAllowedDomain: sesta.com
-
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
changetype: modify
add: mgrpAllowedBroadcaster
mgrpAllowedBroadcaster: mailto: barkley@florizel.com
-
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
changetype: modify
add: mgrpDisallowedBroadcaster
mgrpDisallowedBroadcaster:
 ldap:///cn=barry,ou=people,o=sesta.com,o=isp
```

Code Example 5-4 Mailing List LDIF Record with Delivery Restrictions

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
cn: Dreamteam
objectClass: groupOfUniqueNames
objectClass: inetMailGroup
objectClass: inetLocalMailRecipient
objectClass: inetMailGroupManagement
objectClass: nsManagedMailList
uniqueMember: uid=baylor,ou=People,o=sesta.com,o=isp
uniqueMember: uid=bird,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jordan,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jabbar,ou=People,o=sesta.com,o=isp
uniqueMember: uid=magic,ou=People,o=sesta.com,o=isp
mgrpRFC822MailMember: west@florizel.com
mgrpRFC822MailMember: robertson@florizel.com
mail: dreamteam@sesta.com
mailAlternateAddress: thegreatest@sesta.com
mailHost: manatee.siroe.com
dataSource: @(#)ims50users.sh 1.5a 02/3/00
inetMailGroupStatus: active
nsNumUsers: 7
nsMaxUsers: 1000
mgrpAllowedDomain:  sesta.com
mgrpAllowedBroadcaster:  mailto: barkley@florizel.com
mgrpDisallowedBroadcaster: ldap:///cn=barry,ou=people,o=sesta.com,o=isp
```

Mailing List Moderators

A mailing list moderator is a user who receives a mailing list message before all other members, then forwards it to the rest of the members if desired. Any message submitted to the mailing list will go to the moderator instead of the mailing list members. Set a valid DN or email address to the attribute `moderator`. Multiple moderators are allowed.

Moderators are created by setting `mgrpModerator` to an RFC 822 email address or a DN in URL format.

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
changetype: modify
add: mgrpModerator
mgrpModerator: ldap:///uid=baylor,ou=People,o=sesta.com,o=isp
```

Code Example 5-5 Mailing List LDIF Record with Moderator

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
cn: Dreamteam
objectClass: groupOfUniqueNames
objectClass: inetMailGroup
objectClass: inetLocalMailRecipient
objectClass: inetMailGroupManagement
objectClass: nsManagedMailList
uniqueMember: uid=baylor,ou=People,o=sesta.com,o=isp
uniqueMember: uid=bird,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jordan,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jabbar,ou=People,o=sesta.com,o=isp
uniqueMember: uid=magic,ou=People,o=sesta.com,o=isp
mgrpRFC822MailMember: west@florizel.com
mgrpRFC822MailMember: robertson@florizel.com
mail: dreamteam@sesta.com
mailAlternateAddress: thegreatest@sesta.com
mailHost: manatee.siroe.com
dataSource: @(#)ims50users.sh 1.5a 02/3/00
inetMailGroupStatus: active
nsNumUsers: 7
nsMaxUsers: 1000
mgrpModerator: ldap:///uid=baylor,ou=People,o=sesta.com,o=isp
```

Enable/Disable/Delete Mailing Lists

Mailing lists can be enabled, temporarily disabled, or deleted by setting `inetMailGroupStatus` to `active`, `inactive`, or `deleted`. A disabled mailing list can be activated by resetting `inetMailGroupStatus` to `active`. Missing value implies status is `active`. An illegal value is treated as `inactive`.

The following LDIF code disables the mailing list.

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
changetype: modify
add: inetMailGroupStatus
inetMailGroupStatus: inactive
```

Code Example 5-6 LDIF Record with Mailing List Disabled

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
cn: Dreamteam
objectClass: groupOfUniqueNames
objectClass: inetMailGroup
objectClass: inetLocalMailRecipient
objectClass: inetMailGroupManagement
objectClass: nsManagedMailList
uniqueMember: uid=baylor,ou=People,o=sesta.com,o=isp
uniqueMember: uid=bird,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jordan,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jabbar,ou=People,o=sesta.com,o=isp
uniqueMember: uid=magic,ou=People,o=sesta.com,o=isp
mgrpRFC822MailMember: west@florizel.com
mgrpRFC822MailMember: robertson@florizel.com
mail: dreamteam@sesta.com
mailAlternateAddress: thegreatest@sesta.com
mailHost: manatee.siroe.com
dataSource: @(#)ims50users.sh 1.5a 02/3/00
nsNumUsers: 7
nsMaxUsers: 1000
inetMailGroupStatus: inactive
```

Archiving Messages to a File

Archive mailing list messages by setting `mailDeliveryFileURL` to a URL file. The example below saves messages to `dreamteam@sesta.com` to `/home/dreamteam/mail_archive.log`

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
changetype: modify
add: mailDeliveryFileURL
mailDeliveryFileURL: file:///home/dreamteam/mail_archive.log
```

Code Example 5-7 Mailing List LDIF Record with Archive Attribute

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
cn: Dreamteam
objectClass: groupOfUniqueNames
```

Code Example 5-7 Mailing List LDIF Record with Archive Attribute

```
objectClass: inetMailGroup
objectClass: inetLocalMailRecipient
objectClass: inetMailGroupManagement
objectClass: nsManagedMailList
uniqueMember: uid=baylor,ou=People,o=sesta.com,o=isp
uniqueMember: uid=bird,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jordan,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jabbar,ou=People,o=sesta.com,o=isp
uniqueMember: uid=magic,ou=People,o=sesta.com,o=isp
mgrpRFC822MailMember: west@florizel.com
mgrpRFC822MailMember: robertson@florizel.com
mail: dreamteam@sesta.com
mailAlternateAddress: thegreatest@sesta.com
mailHost: manatee.siroe.com
dataSource: @(#)ims50users.sh 1.5a 02/3/00
inetMailGroupStatus: active
nsNumUsers: 7
nsMaxUsers: 1000
mailDeliveryFileURL: file:///home/dreamteam/mail_archive.log
```

Request Addresses

Specify the mailing list subscription request address with the `mgrpRequestsTo` attribute. An example of a subscription request is `dreamteam-request@sesta.com` Only internal addresses are allowed, and these must be in URL format.

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
changetype: modify
add: mgrpRequestsTo
mgrpRequestsTo: uid=baylor,ou=People,o=sesta.com,o=isp
```

Code Example 5-8 Mailing List LDIF Record with Subscription Request Attribute

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
cn: Dreamteam
objectClass: groupOfUniqueNames
objectClass: inetMailGroup
objectClass: inetLocalMailRecipient
objectClass: inetMailGroupManagement
objectClass: nsManagedMailList
uniqueMember: uid=baylor,ou=People,o=sesta.com,o=isp
uniqueMember: uid=bird,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jordan,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jabbar,ou=People,o=sesta.com,o=isp
uniqueMember: uid=magic,ou=People,o=sesta.com,o=isp
mgrpRFC822MailMember: west@florizel.com
mgrpRFC822MailMember: robertson@florizel.com
mail: dreamteam@sesta.com
```

Code Example 5-8 Mailing List LDIF Record with Subscription Request Attribute

```
mailAlternateAddress: thegreatest@sesta.com
mailHost: manatee.siroe.com
dataSource: @(#)ims50users.sh 1.5a 02/3/00
inetMailGroupStatus: active
nsNumUsers: 7
nsMaxUsers: 1000
mgrpRequestsTo: uid=baylor,ou=People,o=sesta.com,o=isp
```

Visibility of Mailing List Members

Mailing list members are typically visible through the iPlanet Console, iPlanet Delegated Administrator for Messaging, or the SMTP `EXPN` command. Visibility can be limited by setting `mgmanMemberVisibility` to `ANYONE` (anyone in the world can view), `ALL` (anyone in the directory can view), `NONE` (only owner can view).

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
changetype: modify
add: mgmanMemberVisibility
mgmanMemberVisibility: ALL
```

Code Example 5-9 Mailing LIst LDIF Record with Archive Attribute

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
cn: Dreamteam
objectClass: groupOfUniqueNames
objectClass: inetMailGroup
objectClass: inetLocalMailRecipient
objectClass: inetMailGroupManagement
objectClass: nsManagedMailList
uniqueMember: uid=baylor,ou=People,o=sesta.com,o=isp
uniqueMember: uid=bird,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jordan,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jabbar,ou=People,o=sesta.com,o=isp
uniqueMember: uid=magic,ou=People,o=sesta.com,o=isp
mgrpRFC822MailMember: west@florizel.com
mgrpRFC822MailMember: robertson@florizel.com
mail: dreamteam@sesta.com
mailAlternateAddress: thegreatest@sesta.com
mailHost: manatee.siroe.com
dataSource: @(#)ims50users.sh 1.5a 02/3/00
inetMailGroupStatus: active
nsNumUsers: 7
nsMaxUsers: 1000
mgmanMemberVisibility: ALL
```

Making Mailing Lists Joinable

You can specify who may join a mailing list by setting the attribute `mgmanJoinability`. The possible values for this task are `ANYONE` (anyone in the world can join), `ALL` (anyone in the directory can join), `NONE` (no additional members can join).

```
dn: cn=dreamteam,ou=groups,o=sesta.com,o=isp
changetype: modify
add: mgmanJoinability
mgmanJoinability:  ALL
```

Code Example 5-10 LDIF Record for a Joinable Mailing List

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
cn: Dreamteam
objectClass: groupOfUniqueNames
objectClass: inetMailGroup
objectClass: inetLocalMailRecipient
objectClass: inetMailGroupManagement
objectClass: nsManagedMailList
uniqueMember: uid=baylor,ou=People,o=sesta.com,o=isp
uniqueMember: uid=bird,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jordan,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jabbar,ou=People,o=sesta.com,o=isp
uniqueMember: uid=magic,ou=People,o=sesta.com,o=isp
mgrpRFC822MailMember: west@florizel.com
mgrpRFC822MailMember: robertson@florizel.com
mail: dreamteam@sesta.com
mailAlternateAddress: thegreatest@sesta.com
mailHost: manatee.siroe.com
dataSource: @(#)ims50users.sh 1.5a 02/3/00
inetMailGroupStatus: active
nsNumUsers: 7
nsMaxUsers: 1000
mgmanJoinability:  All
```

Creating Dynamic Mailing Lists

The iPlanet Message Server supports both static and dynamic mailing lists. Unlike static mailing lists, where members of the list are specified by using `uniquemember` and `mgrprfc822mailmember` attributes, dynamic mailing list members are specified using an LDAP search filter (RFC-2254). The LDAP filter is set in the `mgrpDeliverTo` attribute in the `inetMailGroup` objectclass.

The example below shows a mailing list consisting of static members and members determined with an LDAP search filter. The filter below includes as members in `o=sesta.com,o=isp` who also have the attribute value-pair `city=tokyo`. You should make sure that the attributes used in the LDAP search filter are indexed; otherwise, the evaluating membership of dynamic lists will be both time consuming and stress the directory server.

Code Example 5-11 LDIF Record for a Dynamic Mailing List

```
dn: cn=Dreamteam,ou=groups,o=sesta.com,o=isp
cn: Dreamteam
objectClass: groupOfUniqueNames
objectClass: inetMailGroup
objectClass: inetLocalMailRecipient
objectClass: inetMailGroupManagement
objectClass: nsManagedMailList
uniqueMember: uid=baylor,ou=People,o=sesta.com,o=isp
uniqueMember: uid=bird,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jordan,ou=People,o=sesta.com,o=isp
uniqueMember: uid=jabbar,ou=People,o=sesta.com,o=isp
uniqueMember: uid=magic,ou=People,o=sesta.com,o=isp
mgrpRFC822MailMember: west@florizel.com
mgrpRFC822MailMember: robertson@florizel.com
mail: dreamteam@sesta.com
mailAlternateAddress: thegreatest@sesta.com
mgrpdeliverto: ldap:///o=sesta.com,o=isp??sub?
 (&(objectclass=inetMailUser)(city=tokyo)
mailHost: manatee.siroe.com
dataSource: @(#)ims50users.sh 1.5a 02/3/00
inetMailGroupStatus: active
nsNumUsers: 7
nsMaxUsers: 1000
mgmanJoinability:  All
```

NOTE iPlanet Messaging Server also supports dynamic lists based on the attribute `memberURL` from the objectclass `groupofurls`. Netscape Directory Server 4.x allows creating of dynamic groups using this attribute and messaging server can take advantage of any groups that may have already been defined using `memberURL`.

Chapter 6

Provisioning Messaging Server Administrators

This chapter describes how to provision the different types of Messaging Server Administrators (Table 6-1). It contains the following sections:

Administrator Types

iPlanet Message Server administrators are classified by two sets of privileges:

- Privileges to configure messaging server (Server Administrators).
- Privileges to add, modify, and delete users and groups in the system (Messaging Directory Administrators).

Table 6-1 Messaging Server Administrators and Privileges *(1 of 2)*

Administrator	Description/Scope of Privileges	Permissions/Creation
Server Administrators:		
Configuration Administrator	Can configure all servers and modify all directory data in the entire topology. Has system-level access to modify the MTA. Unrestricted access to all resources in the Console. Can provide server access to other administrators.	Config Admin user ID is automatically created when messaging server is first installed. For more information see *Managing Servers with Netscape Console*. **Permissions granted by ACIs at**: o=NetscapeRoot **Admin Account**: uid=admin,ou=adminstrators, ou=topologymanagement,o=NetscapeRoot) **Group DN**: cn=configuration administrators, ou=groups, ou=topologymanagment, o=NetscapeRoot
Directory Manager	Can modify anything in directory. Can configure directory. For security, the Configuration Administrator should not be the same as the Directory Manager.	Directory Manager user ID is created when the Directory Server is installed. Directory Manager credentials are stored in the directory server configuration file `slapd.conf`. *Typical Account:* cn = Directory Manager
Message Store Administrator	System level admins can view mailboxes & specify access control. Using proxy authorization rights, can log in as any user. Can specify partition for a mailbox and run message store utilities. Domain-level admins can't do partitions. Have limited access to message store utilities.	This administrator is created by the Messaging Server Console or command line utilities. **System-wide MS Admin Group DN**: Specified in `store.serviceAdminGroupDN` **Domain MS Admin Group DN**: cn=Store Administrators,ou=Groups, *<OrgTreeDomainSuffix>* **Server MS Administrator** is specified in server configuration variable `store.admin`
Messaging Directory Administrators:		
Top-level Administrator (Also called Service Administrator.)	Creates / modifies / deletes mail users, mailing lists, family accounts, and domains in an entire Messaging Server namespace via DA GUIs or CLIs. Automatically gets all message store privileges for all servers in the topology.	Top-level Administrator is automatically created at installation time. ACIs stored on root node. **Group DN**: cn = Service Administrators, ou=groups,*<OrgTreeRoot>*

Table 6-1 Messaging Server Administrators and Privileges *(2 of 2)*

Administrator	Description/Scope of Privileges	Permissions/Creation
Domain Administrator	Creates / modifies / deletes mail users, mailing lists, and family accounts in a hosted domain via DA GUI or CLIs. By default, is a message store admin for the hosted domain.	Top-level Administrator can create Domain Administrator. ACIs in OrgTree root and DC root and the OrgTree domain node. **Group DN**: cn = Domain Administrators, ou=groups,<*OrgTreeDomain*>
Domain Organization Administrator	Creates / modifies / deletes mail users and mailing lists in a domain organization via DA GUI or CLIs.	Top-level or Domain Administrator can create Domain Organization Administrator. ACIs in root and Domain Organization node. **Group DN**: cn = Organization Administrator, <*DomainOrgDN*>
Family Group Administrator	Adds and removes family members in a family group. Can grant administrative access to other members of group. See "Creating a Family Group Administrator," on page 55	Top-level & Domain Admin can create Family Group Administrator. Permissions stored in LDAP. **Group DN**: cn=Family Group Administrators, <*FamilyGrpDN*>
Mail List Owner	Two sets of rights: ability to create & ability to add / remove members to mailing list.	Top-level, Domain, or Domain Organization Admin can grant permissions to mailing list owner. `nsDACapability` grants creation privileges (see "Adding Mailing List Creation Privileges," on page 76). `owner` grants management privileges (see "Assigning Mailing List Owners," on page 83).

NOTE The Netscape Console documentation at (`http://docs.iplanet.com/docs/manuals/console.html`) provides detailed information on using the console.

Creating a Configuration Administrator

A Configuration Administrator is automatically created at installation time. Additional Configuration Administrators can be created by other Configuration Administrators through the Console. See the he Netscape Console documentation at (`http://docs.iplanet.com/docs/manuals/console.html`) for more information.

Creating Message Store Administrators

Message Store Administrators have *privileges* and *scope*. Privileges are as follows:

- View and monitor user mailboxes through IMAP.
- Specify access control for a message store through IMAP.
- Execute message store command line utilities requiring proxy authentication (for example, MoveUser)
- Using proxy authorization rights, can log in as any user.
- Specify partition for a mailbox.

The scope of the administrator's privileges can be:

- For a single domain (in addition, domain-level admins can't specify partitions and have limited access to certain message store commands).
- For a single message store (that is, the message store of a single messaging server).
- For all the message stores in a mail system topology.
- Top-level Administrators automatically have system-wide message store privileges.
- Messaging Server Administrators created during installation automatically have message store privileges for the installed server.
- Top-level Administrators created during installation or at the console automatically have message store privileges for the entire topology.
- Domain Administrators created on the iPlanet Delegated Administrator for Messaging automatically have message store privileges for the users in the domain on which they are installed.

To Create a Message Store Administrator for a Specific Messaging Server

Privileges required: Configuration Administrator or access to the `mailsrv` account on the Messaging Server machine.

Note that Configuration Administrators automatically receive Message Store privileges on the installed server. Server-specific Message Store Administrators can be created by Console (see the *iPlanet Message Server Administration Guide*) or by command line:

```
configutil -o store.admin -v "adminlist"
```

where `configutil` is a utility that enables you to change configuration options, `store.admins` is the Message Store Administrator parameter, and `adminlist` is a space separated list of fully-qualified UIDs (if in the default domain) or <*uid*>@<*domain*> if in a hosted domain. Refer to the *iPlanet Messaging Server Reference Manual* for details.

To Create a Message Store Administrator for the Entire Mail System Topology

Privileges required: Top-level Administrator or access to the `mailsrv` account on the Messaging Server machine.

By *"entire mail system topology"* we mean all the message stores for all the messaging servers under a common user / group directory root. By default topology-wide message store administrative privileges are only granted to members of the group `cn=Service Administrators,ou=groups,`<*OrgTreeRoot*>. However, it is possible to change these message store privileges to another group by resetting the configuration value `store.serviceAdminGroupDN`. Note that if you do this, members of `cn=Service Administrators,ou=groups,`<*OrgTreeRoot*> will no longer have message store privileges unless they are also added to the new group.

In the example below, we will change the system-wide Message Store Administrator group from `cn=Service Administrators,ou=groups,o=isp` to `cn=System-wide Store Administrators,ou=groups,o=isp` and we'll add Biff as an administrator.

1. Create System-wide Store Administrators Group and Add a Member.

First create a group called *System-wide Store Administrators* and add a member using the `uniqueMember` attribute.

Code Example 6-1 Creating the System-wide Message Store Administrators Group

```
dn: cn=System-wide Store Administrators,ou=groups,o=isp
objectclass: groupOfUniqueNames
cn: System-wide Store Administrators
uniqueMember: uid=Biff,ou=people,o=sesta.com,o=isp
```

2. Set `store.serviceAdminGroupDN` to the DN of the System-wide Message Store Administrators Group.

   ```
   configutil -o store.serviceAdminGroupDN -v "cn=System-wide Store
   Administrators,ou=groups,o=isp"
   ```

   This must be done on each server in the system.

3. Set `memberof` attribute in the user entry.

Code Example 6-2 Example User Entry for a System-wide Message Store Administrator

```
dn: uid=Biff,ou=people,o=sesta.com,o=isp
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
cn: Biff Fanning
sn: fanning
initials: BTF
givenName: Biff
mail: Biff.Fanning@sesta.com
mailAlternateAddress: bfanning@florizel.com
mailDeliveryOption: mailbox
mailHost: manatee.siroe.com
uid: biff
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: {SHA}aluWfd0LYY9ImsJb3h4afrI4AXk=
mailAllowedServiceAccess: +imap, imaps, pop3, smtp, http:*
inetUserStatus: active
mailUserStatus: active
```

Code Example 6-2 Example User Entry for a System-wide Message Store Administrator

```
mailQuota: -1
mailMsgQuota: 100
memberOf: cn=System-wide Store Administrators,ou=groups o=sesta.com,o=isp
```

To Create a Message Store Administrator for a Specific Domain

Privileges required: Domain Administrator, or Top-level Administrator

Domain Message Store Administrators can be created as follows:

- By using the iPlanet Delegated Administrator for Messaging GUI to convert a user into a Delegated Administrator.
- By provisioning through LDAP.

The following example grants the user Biff message store privileges in sesta.com through LDAP.

1. Create Store Administrators Group and add a member.

   Create a group called *Store Administrators* in the domain node of the Organization Tree. Add the `inetMailAdministrator` object class and set the attribute `mailAdminRole` to `storeadmin` to the group entry. Add a member using the `uniqueMember` attribute. See the LDIF data below.

   Note that the ACIs are created automatically at installation, and this group is created whenever a domain is created with the Delegated Administrator or Console.

Code Example 6-3 Creating the Store Administrator Group

```
dn: cn=Store Administrators,ou=Groups,o=sesta.com,o=isp
objectclass: groupOfUniqueNames
objectclass: inetMailAdministrator
cn: Store Administrators
mailAdminRole: storeadmin
uniqueMember: uid=Biff,ou=People,o=sesta.com,o=isp
```

- `objectclass: groupOfUniqueNames`
  objectclass: `inetMailAdministrator`

  The `groupOfUniqueNames` object class contains attributes for describing a collection of directory entries (namely users and other groups).

  `inetMailAdministrator` specifies attributes that confer administrative privileges to this group.

- `cn: Store Administrators`

  This is the common name of the group of which Message Store Administrators must be a member.

- `mailAdminRole: storeadmin`

  The type of administrative privileges conferred on this group.

- `uniqueMember: uid=Biff,ou=People,o=sesta.com,o=isp`

  DN of a member. In this example there is only one member in this group.

2. Specify the `memberOf` attribute in the user's entry to `cn=Store Administrators,ou=groups,o=sesta.com,o=isp`

Code Example 6-4 Example User Entry for a Domain Administrator

```
dn: uid=Biff,ou=people,o=sesta.com,o=isp
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
cn: Biff Fanning
sn: fanning
initials: BTF
givenName: Biff
mail: Biff.Fanning@sesta.com
mailAlternateAddress: bfanning@florizel.com
mailDeliveryOption: mailbox
mailHost: manatee.siroe.com
uid: biff
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: {SHA}aluWfd0LYY9ImsJb3h4afrI4AXk=
mailAllowedServiceAccess: +imap, imaps, pop3, smtp, http:*
inetUserStatus: active
mailUserStatus: active
```

Code Example 6-4 Example User Entry for a Domain Administrator

```
mailQuota: -1
mailMsgQuota: 100
memberOf: cn=Store Administrators,ou=groups o=sesta.com,o=isp
```

- `dn: uid=Biff,ou=People,o=sesta.com,o=isp`

  The DN of the user designated to be a Message Store Administrator to this group.

- `memberOf: cn=Store Administrators,ou=groups,o=sesta.com,o=isp`

  DN of a group to which Biff belongs.

Creating Top-level Administrators

Task Privilege: Top-level Administrator

A Top-level administrator has directory and message store privileges to the entire messaging system. A default Top-level Administrator is created at installation, but additional Top-level Administrators can created by adding users to the following group:

`cn=Service Administrators,ou=Groups,o=<`*OrgTreeRoot*`>`

and by specifying the `memberOf` attribute in the user's entry to `cn=Service Administrators,o=groups,o=<`*OrgTreeRoot*`>`

The example below makes Biff Fanning a Top-level Administrator. Note that the installer creates the appropriate ACIs for this entry. If you are creating the directory from scratch, see Appendix A, "Root and Domain ACI Examples."

Code Example 6-5 The Top-level Administrator Group

```
dn: cn=Service Administrators,ou=Groups,o=isp
objectclass: groupOfUniqueNames
objectclass: nsManagedDept
cn: Service Administrators
nsNumUsers: 1
nsMaxUsers: Unlimited
uniqueMember: uid=Biff,ou=People,o=sesta.com,o=isp
```

Code Example 6-6 Example User Entry for a Top-level Administrator

```
dn: uid=Biff,ou=people,o=sesta.com,o=isp
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
cn: Biff Fanning
sn: fanning
initials: BTF
givenName: Biff
mail: Biff.Fanning@sesta.com
mailAlternateAddress: bfanning@florizel.com
mailDeliveryOption: mailbox
mailHost: manatee.siroe.com
uid: biff
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: {SHA}aluWfd0LYY9ImsJb3h4afrI4AXk=
mailAllowedServiceAccess: +imap, imaps, pop3, http:*
inetUserStatus: active
mailUserStatus: active
mailQuota: -1
mailMsgQuota: 100
memberOf: cn=Service Administrators,ou=groups,o=isp
```

Creating Domain Administrators

Delegated Admin Utility: `imadmin admin add`
Task Privilege for Provisioning: Top-level Administrator

A *domain administrator* is a user who has privileges to add, delete, and modify users and groups in a particular domain using the Delegated Administrator or the command line utilities. Only Top-level Administrators can create Hosted Domain Administrators.

Once the Domain Administrator's group has been created and the ACI rules have been set, it no longer has to be done again. To create new administrators, simply add them to the group. The following LDIF examples create a Domain Administrators group and add Biff as a member of this group.

1. Create a Domain Administrators group and add a user to the group.

   Create a group called *Domain Administrators* in the hosted domain node of the Organization Tree and add the DN of the user designated to be a Domain Administrator to this group. Also, add the object class `inetMailAdministrator` and the attribute value pair `mailadminrole: storeadmin`. (Note that this group with ACIs is automatically created when a domain is created with the Delegated Administrator.) Specify the `uniqueMember` attribute in the Domain Administrator's Group to the DN of the new Domain Administrator. This is shown below.

Code Example 6-7 Creating the Domain Administrator Group

```
dn: cn=Domain Administrators,ou=groups,o=sesta.com,o=isp
objectclass: groupOfUniqueNames
objectClass: nsManagedDept
objectClass: inetMailAdministrator
mailadminrole: storeadmin
cn: Domain Administrators
uniqueMember: uid=Biff,ou=People,o=sesta.com,o=isp
```

- `objectclass: groupOfUniqueNames`
  `objectClass: nsManagedDept`
  `objectClass: inetMailAdministrator`

  The `groupOfUniqueNames` object class contains attributes for describing a collection of directory entries (namely users and other groups).

- `cn: Domain Administrators`

  This is common name of the group of which domain administrators must be a member.

- `mailadminrole: storeadmin`

  Grants message store administrator privileges to members of this group.

- `uniqueMember: uid=Biff,ou=People,o=sesta.com,o=isp`

  `uniqueMember` specifies the distinguished names of the members of this list. In this example there is only one member in this group.

2. Verify Domain Administrators ACI Rules.

Domain administrator ACI rules are created automatically when you create a hosted domain using the Delegated Administrator or command line utilities like `imadmin domain create`. If you are provisioning hosted domains using LDAP, you will need to add ACI rules. An example is shown Appendix A, “Root and Domain ACI Examples."

3. Add `memberOf` to User Entry.

Specify the `memberOf` attribute to `cn=Domain Administrators,o=groups,o=sesta.com,o=isp` in the user’s entry.

Code Example 6-8 Example User Entry for a Domain Administrator

```
dn: uid=Biff,ou=people,o=sesta.com,o=isp
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
cn: Biff Fanning
sn: fanning
initials: BTF
givenName: Biff
mail: Biff.Fanning@sesta.com
mailAlternateAddress: bfanning@florizel.com
mailDeliveryOption: mailbox
mailHost: manatee.siroe.com
uid: fanning
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: password
mailAllowedServiceAccess: +imap, imaps, pop3, http:*
inetUserStatus: active
mailUserStatus: active
mailQuota: -1
mailMsgQuota: 100
memberOf: cn=Domain Administrators,o=groups,o=sesta.com,o=isp
```

- `dn: uid=Biff,ou=People,o=sesta.com,o=isp`

  The DN of the user designated to be a domain administrator for this domain.

- `memberOf: cn=Domain Administrators,o=sesta.com,o=isp`

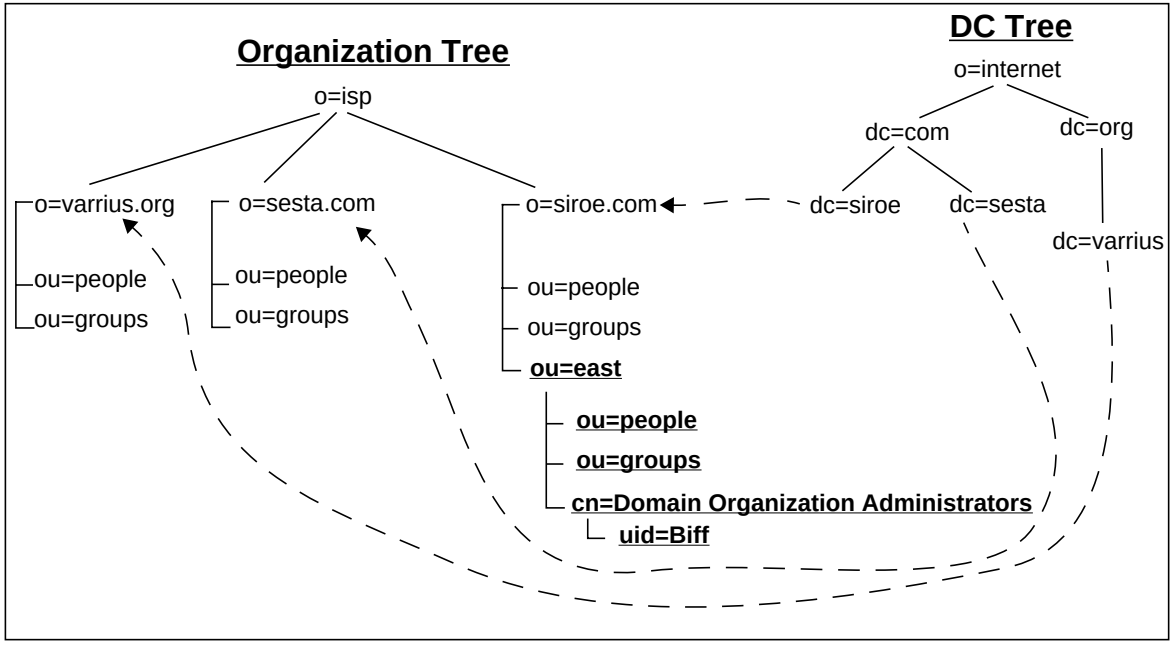
  DN of the group to which this user belongs.

Creating a Domain Organization Administrator

A *Domain Organization Administrator* is a user of an organization who has privileges to add, delete, and modify users and groups in a particular organization using the Delegated Administrator or the command line utilities. Multiple Domain Organization Administrators can be contained in a hosted domain, and Domain Organization Administrators can be nested. Only Top-level Administrators can create Organization Administrators.

Once the Organization Administrator's group has been created and the ACI rules have been set, it no longer has to be done again. To create new administrators, simply add them to the group. The example below shows how to create an Organization Administrator, Biff, for ou=east,o=siroe.com,o=isp.

Figure 6-1 Creating a Domain Organization Administrator



See "Creating a Domain Organization," on page 43 for how to create a domain organization.

1. Create a group called *Domain Organization Administrators* in the domain organization node of the organization tree and add the DN of the Domain Organization Administrator of this group.

Code Example 6-9 Creating the Organization Administrator Group

```
dn: cn=Domain Organization Administrators,ou=east,o=siroe.com,o=isp
objectclass: nsManagedDept
objectclass: inetAdmin
objectclass: groupOfUniqueNames
cn: Domain Organization Administrators
uniqueMember: uid=Biff,ou=people,ou=east,o=siroe.com,o=isp
```

- `dn: cn=Organization Administrators,ou=groups,ou=east,o=siroe.com,o=isp`

  Name of Organization Administrator's group.

- `objectclass: nsManagedDept`
  `objectclass: inetAdmin`
  `objectclass: groupOfUniqueNames`

  `nsManagedDept` attributes to support Delegated Administrator. `inetAdmin` provides attributes to support administration. The `groupOfUniqueNames` object class contains attributes for describing a collection of directory entries (namely users and other groups).

- `cn: Organization Administrators`

  This is the common name of the group of which organization administrators must be a member

- `uniqueMember: uid=Biff,ou=People,o=east.siroe.com,o=isp`

  `uniqueMember` specifies the distinguished names of the members of this list. In this example there is only one member in this group.

2. Add Domain Organization Administrator ACI Rules.

   You must add and modify the appropriate ACI rules to the domain organization. In this example that would be `ou=east,o=siroe.com,o=isp`. An example is shown Appendix A, "Root and Domain ACI Examples."

3. Specify the `memberOf` attribute in the Domain Organization Administrator's entry.

   Specify the `memberOf` attribute to `cn=Domain Organization Administrators,o=east.siroe.com,o=isp` in `uid=Biff,ou=people,o=sesta.com,o=isp`

Code Example 6-10 Example User Entry for a Domain Administrator

```
dn: uid=Biff,ou=people,o=sesta.com,o=isp
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: inetUser
objectClass: ipUser
objectClass: inetMailUser
objectClass: inetLocalMailRecipient
objectClass: nsManagedPerson
objectClass: userPresenceProfile
cn: Otis Fanning
sn: fanning
initials: BTF
givenName: Biff
mail: Biff.Fanning@sesta.com
mailAlternateAddress: bfanning@florizel.com
mailDeliveryOption: mailbox
mailHost: manatee.siroe.com
uid: fanning
dataSource: iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword: {SHA}aluWfd0LYY9ImsJb3h4afrI4AXk=
mailAllowedServiceAccess: +imap, imaps, pop3, http:*
inetUserStatus: active
mailUserStatus: active
mailQuota: -1
mailMsgQuota: 100
memberOf: cn=Domain Organization Administrators,ou=east,o=siroe.com,o=isp
```

- `dn: uid=Biff,ou=People,o=eng.siroe.com,o=isp`

  The DN of the user designated to be a domain organization administrator to this group.

- `memberOf: cn=Organization Administrators,ou=groups,ou=east,o=siroe.com,o=isp`

  DN of the group to which this user belongs.

Root and Domain ACI Examples

The ACIs listed in this Appendix are the default ACIs installed when a domain or root node is created in the directory information tree. These ACIs can be modified for your system needs. You can also view these ACIs on-line by doing an LDAP search on the root and domain entries. Note that domain organization ACIs must be added using LDAP when domain organizations are created. This Appendix contains the following sections:

NOTE If you are using a DC Tree for domain, user, and group entries (that is, you do not have an Organization Tree), then all the ACIs for the Organization Tree described in this Appendix are not needed. In this case, where "<*OrgRoot*>" appears in ACIs for DC Tree, change them to the value of <*DCRoot*>.

Variable Definitions in ACI Example

<*OrgRoot*> - Root of the Organization Tree. This is where the user and group entries are created in a default installation.

<*DCRoot*> - Root of the Domain Component Tree. This is where domain entries are created.

<*OrgNodeDN*> - Domain node in the Organization Tree. This is where the user and group entries for a domain reside.

<*DCNodeDN*> - Domain node in the DC Tree. This is where the user and group entries for a domain reside.

<*DomainOrgNodeDN*> - Root of the Domain Component Tree. This is where domain entries are created.

Organization Tree Root Node ACIs

The ACIs below grant required access to Top-level Administrators, Domain Administrators, Domain Organization Administrators, Family Group Administrators, Mail List Owners, and End Users. Where necessary, additional ACIs are set on domain nodes and domain organization nodes further down the tree. If you are setting up namespace from scratch (that is, you are not using the iPlanet Message Server installer for preparing the namespace), then you need to set the ACIs on the Organization Tree Root Node.

Code Example A-1 Organization Tree Root Node ACIs

```
dn: <OrgRoot>
changetype: modify
add: aci
#
#-----------------------------------
# iDA User access control
#
# Allow read and search access to all attributes in all entries
#
aci: (targetattr="*") (version 3.0; acl "NDAUser access -
 product=ims5.0,class=nda,num=1,version=1"; allow (read,search)
 userdn="ldap:///uid=NDAUser,ou=config,<OrgRoot>";)
#
# Allow write access to nsNum* attributes of all domain entries
#
aci: (targetattr="nsNumUsers||nsNumDepts||nsNumMailLists||nsNumDomains")
 (version 3.0; acl "NDAUser access - product=ims5.0, class=nda,num=2,
 version=1"; allow (write) userdn="ldap:///uid=NDAUser,ou=config,
 <OrgRoot>";)
#
#-----------------------------------
# Service Administrator access control
#
# Allow read and search access to all DCROOT nodes
#
aci: (targetattr="*") (version 3.0; acl "SA root node access -
 product=ims5.0,class=nda,num=3,version=1"; allow (all)
```

Code Example A-1 Organization Tree Root Node ACIs

```
 groupdn="ldap:///cn=Service Administrators,ou=Groups,<OrgRoot>";)
#
#-----------------------------------
# Domain Administrator control.
#
# Deny write and delete access to any domain container node.
#
aci: (targetfilter="objectclass=nsManagedDomain") (version 3.0; acl
 "Domain Admin domain container access -
 product=ims5.0,class=nda,num=5,version=1"; deny (delete,write)
 userdn="ldap:///<DCRoot>??sub?(memberOf=cn=Domain Administrators*)" or
 userdn="ldap:///<OrgRoot>??sub?(memberOf=cn=Domain Administrators*)";)
#
#-----------------------------------
# User access control
#
# Allow read and search access to self
#
aci: (targetattr="*") (targetfilter=(objectClass=inetOrgPerson)) (version
 3.0; acl "User self search and read - product=ims5.0,class=nda,num=6,
 version=1"; allow (read,search) userdn="ldap:///self";)
#
# Allow write access to self
#
aci: (targetattr="*") (version 3.0; acl "Allow self entry modification -
 product=ims5.0,class=nda,num=7,version=1";
 allow (write) userdn = "ldap:///self";)
#
# Deny write access to self for uid, ou, owner,
# nsDAModifiableBy, nsDACapability, mail, mailAlternateAddress,
# memberOf, and nsDADomain attributes
#
aci: (targetattr="uid||ou||owner||nsDAModifiableBy||nsDACapability||
 mail||mailAlternateAddress||memberOf||nsDADomain||inetuserstatus||
 mailuserstatus||memberOfManagedGroup||mailQuota||mailMsgQuota||
 inetSubscriberAccountId||dataSource||mailhost||mailAllowedServiceAccess
 ||pabURI||inetCOS") (targetfilter=(objectClass=nsManagedPerson))
 (version 3.0; acl "User self modification - product=ims5.0,class=nda,
 num=8,version=1"; deny (write) userdn = "ldap:///self" and
 userdn != "ldap:///<OrgRoot>??sub?(memberOf=cn=Domain Administrators*)"
 and userdn !=
 "ldap:///<DCRoot>??sub?(memberOf=cn=Domain Administrators*)"
 and groupdn != "ldap:///cn=Service Administrators,ou=groups,<OrgRoot>";)
#
# Deny delete access to self
#
aci: (targetfilter=(objectClass=inetOrgPerson)) (version 3.0; acl
 "User self deletion - product=ims5.0,class=nda,num=9,version=1";
 deny (delete) userdn="ldap:///self";)
#
#-----------------------------------
# Mail List access control
#
# Allow designated users to create mail lists
#
```

Code Example A-1 Organization Tree Root Node ACIs

```
aci: (targetattr="*")(targetfilter=(objectClass=inetMailGroupManagement))
 (version 3.0; acl "Mail list create access - product=ims5.0,class=nda,
 num=10,version=1"; allow (add)
 userdn="ldap:///<OrgRoot>??sub?(nsDACapability=mailListCreate)";)
#
# Allow maillist owner read, search, write, and delete access
# to the maillists s/he owns except for the nsMaxUsers attr
#
aci: (targetattr="*") (targetfilter=(objectClass=inetMailGroupManagement))
 version 3.0; acl "Mail list owner access - product=ims5.0,class=nda,num=11,
 version=1"; allow (read,search,write,delete)
 groupdnattr="ldap:///<OrgRoot>?owner";)
#
#-----------------------------------
# Family Group Administrator access control
#
# family group read access
#
aci: (targetattr="*") (targetfilter=(objectClass=inetManagedGroup))
 (version 3.0; acl "Family Group Adm group read & search access -
  product=ims5.0,class=nda,num=12,version=1"; allow (read,search)
 userdn="ldap:///<OrgRoot>??sub?(memberOf=cn=Family Group
 Administrators*)" and groupdnattr="ldap:///<OrgRoot>?nsDAModifiableBy";)
#
# family group write access for 'description' attribute
#
aci: (targetattr="description")
 (targetfilter=(objectClass=inetManagedGroup))
 (version 3.0; acl "Family Group Adm description write access -
 product=ims5.0,class=nda,num=13,version=1"; allow (write)
 userdn="ldap:///<OrgRoot>??sub?(memberOf=cn=Family Group
 Administrators*)" and groupdnattr="ldap:///<OrgRoot>?nsDAModifiableBy";)
#
# family group write access for 'mnggrpCurrentUsers' attribute
#
aci: (targetattr="mnggrpCurrentUsers")
 (targetfilter=(objectClass=inetManagedGroup)) (version 3.0; acl "Family
 Group Adm description write access - product=ims5.0,class=nda,num=14,
 version=1"; allow (write)
 userdn="ldap:///<OrgRoot>??sub?(memberOf=cn=Family Group
 Administrators*)" and groupdnattr="ldap:///<OrgRoot>?nsDAModifiableBy";)
#
# family member create,delete,modify permissions
#
aci: (targetattr="*") (targetfilter=(objectClass=nsManagedPerson))
 (version 3.0;acl "Family Group Adm member access - product=ims5.0,
 class=nda, num=15,version=1"; allow (add,read,search,write,delete)
 userdn="ldap:///<OrgRoot>??sub?(memberOf=cn=Family Group
 Administrators*)" and groupdnattr="ldap:///<OrgRoot>?nsDAModifiableBy";)
#
# access to add,remove family admins of the same admin group
#
aci: (targetattr="uniquemember")
 (targetfilter=(&(|(objectClass=nsManagedDept)
```

Code Example A-1 Organization Tree Root Node ACIs

```
 (objectClass=nsManagedDeptAdminGroup))(cn=Family Group
 Administrators*))) (version 3.0;acl "Family Group Adm admin write
 access - product=ims5.0,class=nda,num=16,version=1"; allow (write)
 userdn="ldap:///<OrgRoot>??sub?(memberOf=cn=Family Group
 Administrators*)" and groupdnattr="ldap:///<OrgRoot>?uniquemember";)
#
# access to add,remove memberof attribute
#
aci: (targetattr="memberOf") (targetfilter=(objectClass=nsManagedPerson))
 (version 3.0;acl "Family Adm user access -
 product=ims5.0,class=nda,num=17,version=1"; allow (write)
 userdn="ldap:///<OrgRoot>??sub?(memberOf=cn=Family Group
 Administrators*)" and groupdnattr="ldap:///<OrgRoot>?nsDAModifiableBy";)
#
#-----------------------------------
# Domain Organization Administrator
#
# access to the Domain Organization nodes.
#
aci: (targetattr="*") (targetfilter=(objectClass=inetdomainorg))(version
 3.0; acl "Domain Organization Administrator - Dom Org node read & search
 access - product=ims5.0,class=nda,num=21,version=1"; allow (read,search)
 groupdnattr="ldap:///<OrgRoot>?nsDAModifiableBy";)
#
# write access for selected attribute
#
aci: (targetattr="description||domOrgMaxUsers")
 (targetfilter=(objectClass=inetdomainorg)) (version 3.0; acl "Domain
 Organization Administrator - Dom Org node write access -
 product=ims5.0,class=nda,num=22,version=1"; allow (write)
 groupdnattr="ldap:///<OrgRoot>?nsDAModifiableBy";)
```

DC Tree Root Node ACIs

The ACIs below grant required access to Top-level Administrators, Domain Administrators, Domain Organization Administrators, Family Group Administrators, Mail List Owners, and End Users. Where necessary, additional ACIs are set on domain nodes and domain organization nodes further down the tree. If you are setting up namespace from scratch (that is, you are not using the iPlanet Message Server installer for preparing the namespace), then you need to set the ACIs on the DC Tree Node.

Code Example A-2 DC Tree Root Node ACIs

```
dn: <DCRoot>
changetype: modify
add: aci
#-----------------------------------
#
# iDA User access control
#
# Allow read and search access to all attributes in all entries
#
aci: (targetattr="*") (version 3.0; acl "NDAUser access -
 product=ims5.0,class=nda,num=1,version=1"; allow (read,search)
 userdn="ldap:///uid=NDAUser,ou=config,<OrgRoot>";)
#
# Allow write access to nsNum* attributes of all domain entries
#
aci: (targetattr="nsNumUsers||nsNumDepts||nsNumMailLists||nsNumDomains")
 (version 3.0; acl "NDAUser access - product=ims5.0,class=nda,num=2,
 version=1"; allow (write) userdn="ldap:///uid=NDAUser,
 ou=config,<OrgRoot>";)
#
#-----------------------------------
# Service Administrator access control
#
# Allow read and search access to all DCROOT nodes
#
aci: (targetattr="*") (version 3.0; acl "SA root node access -
 product=ims5.0,class=nda,num=3,version =1"; allow (all)
  groupdn="ldap:///cn=Service Administrators,ou=Groups,<OrgRoot>";)
#
#-----------------------------------
# Domain Administrator control.
#
# Access to dcroot to search for domain components
#
aci: (targetattr="*") (version 3.0; acl "Domain Admin dc root access -
 product=ims5.0,class=nda,num=4 ,version=1"; allow (read,search)
 userdn="ldap:///<DCRoot>??sub?(memberOf=cn=Domain Administrators*)" or
 userdn="ldap:///<OrgRoot>??sub?(memberOf=cn=Domain Administrators*)";)
#
# Deny write and delete access to any domain container node.
#
aci: (targetfilter="objectclass=nsManagedDomain") (version 3.0; acl
 "Domain Admin domain container access -
 product=ims5.0,class=nda,num=5,version=1"; deny (delete,write)
 userdn="ldap:///<DCRoot>??sub?(memberOf=cn=Domain Administrators*)" or
 userdn="ldap:///<OrgRoot>??sub?(memberOf=cn=Domain Administrators*)";)
#
#-----------------------------------
# User access control
#
# Allow read and search access to self
#
```

Code Example A-2 DC Tree Root Node ACIs

```
aci: (targetattr="*") (targetfilter=(objectClass=inetOrgPerson)) (version
 3.0; acl "User self search and read - product=ims5.0,class=nda, num=6,
 version=1"; allow (read,search) userdn="ldap:///self";)
#
# Allow write access to self
#
aci: (targetattr = "*") (version 3.0; acl "Allow self entry modification
 - product=ims5.0,class=nda,num=7,version=1"; allow (write) userdn =
 "ldap:///self";)
#
# Deny write access to self for uid, ou, owner,
# nsDAModifiableBy, nsDACapability, mail, mailAlternateAddress,
# memberOf, and nsDADomain attributes
#
aci: (targetattr="uid||ou||owner||nsDAModifiableBy||nsDACapability||
 mail||mailAlternateAddress||memberOf||nsDADomain||inetuserstatus||
 mailuserstatus||memberOfManagedGroup||mailQuota||mailMsgQuota||
 inetSubscriberAccountId||dataSource||mailhost||mailAllowedServiceAccess
 ||pabURI||inetCOS") (targetfilter=(objectClass=nsManagedPerson))
 (version 3.0; acl "User self modification - product=ims5.0,class=nda,
 num=8, version=1"; deny (write) userdn = "ldap:///self" and userdn
 != "ldap:///<DCRoot>??sub?(memberOf=cn=Domain Administrators*)" and
 userdn != "ldap:///<OrgRoot>??sub?(memberOf=cn=Domain Administrators*)"
 and groupdn != "ldap:///cn=Service Administrators,ou=groups,<OrgRoot>";)
#
# Deny delete access to self
#
aci: (targetfilter=(objectClass=inetOrgPerson)) (version 3.0; acl "User
 self deletion - product=ims5.0,class=nda,num=9,version=1"; deny (delete)
 userdn="ldap:///self";)
#
#----------------------------------
# Mail List access control
#
# Allow designated users to create mail lists
#
aci: (targetattr="*")  (targetfilter=(objectClass=inetMailGroupManagement))
 (version 3.0; acl  "Mail list create access - product=ims5.0,class=nda,
 num=10, version=1";  allow (add)
 userdn="ldap:///<DCRoot>??sub?(nsDACapability=mailListCreate)";)
#
# Allow maillist owner read, search, write, and delete access
# to the maillists s/he owns except for the nsMaxUsers attr
#
aci: (targetattr="*")  (targetfilter=(objectClass=inetMailGroupManagement))
 (version 3.0; acl  "Mail list owner access -
 product=ims5.0,class=nda,num=11,version=1";  allow (read,search,write,delete)
 groupdnattr="ldap:///<DCRoot>?owner";)
#
#----------------------------------
# Family Group Administrator access control
#
# family group read access
```

Code Example A-2 DC Tree Root Node ACIs

```
#
aci: (targetattr="*") (targetfilter=(objectClass=inetManagedGroup))
 (version 3.0; acl "Family Group Adm group read & search access -
 product=ims5.0 ,class=nda,num=12,version=1"; allow (read,search)
 userdn="ldap:///<DCRoot>??sub?(memberOf=cn=Family Group
 Administrators*)" and groupdnattr="ldap:///<DCRoot>?nsDAModifiableBy";)
#
# family group write access for 'description' attribute
#
aci: (targetattr="description")
 (targetfilter=(objectClass=inetManagedGroup)) (version 3.0; acl "Family
 Group Adm description write access -
 product=ims5.0,class=nda,num=13,version=1"; allow (write)
 userdn="ldap:///<DCRoot>??sub?(memberOf=cn=Family Group
 Administrators*)" and groupdnattr="ldap:///<DCRoot>?nsDAModifiableBy";)
#
# family group write access for 'mnggrpCurrentUsers' attribute
#
aci: (targetattr="mnggrpCurrentUsers")
 (targetfilter=(objectClass=inetManagedGroup)) (version 3.0; acl "Family
 Group Adm description write access -
 product=ims5.0,class=nda,num=14,version=1"; allow (write)
 userdn="ldap:///<DCRoot>??sub?(memberOf=cn=Family Group
 Administrators*)" and groupdnattr="ldap:///<DCRoot>?nsDAModifiableBy";)
#
# family member create,delete,modify permissions
#
aci: (targetattr="*") (targetfilter=(objectClass=nsManagedPerson))
 (version 3.0;acl "Family Group Adm member access -
 product=ims5.0,class=nda,num=15,version=1"; allow
 (add,read,search,write,delete)
 userdn="ldap:///<DCRoot>??sub?(memberOf=cn=Family Group
 Administrators*)" and groupdnattr="ldap:///<DCRoot>?nsDAModifiableBy";)
#
# access to add,remove family admins of the same admin group
#
aci: (targetattr="uniquemember")
 (targetfilter=(&(|(objectClass=nsManagedDept)(objectClass=nsManagedDept
 AdminGroup))(cn=Family Group Administrators*))) (version 3.0;acl "Family
 Group Adm admin write access - product=ims5.0,class=nda,num=16,
 version=1"; allow (write) userdn="ldap:///<DCRoot>??sub?(memberOf=cn=
 Family Group Administrators*)" and
 groupdnattr="ldap:///<DCRoot>?uniquemember";)
#
# access to add,remove memberof attribute
#
aci: (targetattr="memberOf") (targetfilter=(objectClass=nsManagedPerson))
 (version 3.0;acl "Family Adm user access - product=ims5.0,class=nda,
 num=17,version=1"; allow (write)
 userdn="ldap:///<DCRoot>??sub?(memberOf=cn=Family Group
 Administrators*)" and groupdnattr="ldap:///<DCRoot>?nsDAModifiableBy";)
#
# Family Admin needs to read domain to get the dn
#
aci: (targetattr="objectclass||preferredmailhost||
```

Code Example A-2 DC Tree Root Node ACIs

```
 preferredmailmessagestore") (targetfilter=(objectClass=domain)) (version
 3.0;acl "Family Adm domain access - product=ims5.0,class=nda,num=18,
 version=1"; allow (read,search)
 userdn="ldap:///<OrgRoot>??sub?(memberOf=cn=Family Group
 Administrators*)" or userdn="ldap:///<DCRoot>??sub?(memberOf=cn=Family
 Group Administrators*)";)
#
#-----------------------------------
# Domain Organization Administrator
#
# Allow domain organization administrators to read the
# attributes from the dc tree.
#
aci: (targetattr="objectclass||preferredmailhost||
 preferredmailmessagestore||dc") (targetfilter=(objectClass=domain))
 (version 3.0;acl "Domain Organization Admin domain access -
 product=ims5.0,class=nda,num=20,version=1"; allow (read,search)
 userdn="ldap:///<DCRoot>??sub?(memberOf=cn=Domain Organization
 Administrators*)" or userdn="ldap:///<OrgRoot>??sub?(memberOf=cn=Domain
 Organization Administrators*)";)
#
# access to the Domain Organization nodes.
#
aci: (targetattr="*") (targetfilter=(objectClass=inetdomainorg))(version
 3.0; acl "Domain Organization Administrator - Dom Org node read & search
 access - product=ims5.0,class=nda,num=21,version=1"; allow (read,search)
 groupdnattr="ldap:///<DCRoot>?nsDAModifiableBy";)
#
# write access for selected attribute
#
aci: (targetattr="description||domOrgMaxUsers")
 (targetfilter=(objectClass=inetdomainorg))(version 3.0; acl "Domain
 Organization Administrator - Dom Org node write access -
 product=ims5.0,class=nda,num=22,version=1"; allow (write)
 groupdnattr="ldap:///<DCRoot>?nsDAModifiableBy";)
```

Hosted Domain ACIs

The ACIs below grant required access to Domain Administrators, Mail List Owners, and End Users. The six ACIs below are for the standard two-tree namespace. Five rules on the Organization Tree and one on the DC Tree. If you are using a namespace with just a single DC Tree, all six rules are set on the hosted domain node. These ACIs must be set for every domain you provision.

Code Example A-3 Hosted Domain ACIs

```
dn: <OrgNodeDN>
changetype: modify
add: aci
#
#-----------------------------------
# Domain Administrator access control
#
# allow full access to the domains user/group subtree
#
aci: (targetattr="*") (version 3.0; acl "Domain Admin Domain access -
 product=ims5.0,class=nda,num=18,version=1"; allow (all)
 groupdn="ldap:///cn=Domain Administrators,ou=Groups,<OrgNodeDN>";)
#
#-----------------------------------
# End user access control
# allow users to read and search all users in the domain
#
aci: (targetattr!="userPassword")
 (targetfilter=(|(objectClass=inetOrgPerson)(objectclass=nsManagedDomain
 ))) (version 3.0; acl "User access to all users in domain -
 product=ims5.0,class=nda,num=19,version=1"; allow (read,search)
 userdn="ldap:///<OrgNodeDN>??sub?(objectclass=inetOrgPerson)";)
#
# allow users to add themselves to self subscribe mail lists
#
aci: (targetattr="uniqueMember")
 (targetfilter=(&(objectClass=nsManagedMailList)
 (|(mgmanJoinability=anyone)(mgmanJoinability=all))))
 (version 3.0; acl "User mail list self subscribe access -
 product=ims5.0,class=nda,num=20,version=1"; allow (selfwrite)
 userdn="ldap:///<OrgNodeDN>??sub?(objectclass=inetOrgPerson)";)
#
# hide group members when they are marked hidden
#
aci: (targetattr!="uniqueMember||mgrpRfc822MailMember")
 (targetfilter=(&(objectClass=inetMailGroupManagement)
 (mgmanHidden=false))) (version 3.0; acl "User mail list access when
 visible - product=ims5.0,class=nda,num=21,version=1"; allow
 (read,search)
 userdn="ldap:///<OrgNodeDN>??sub?(objectclass=inetOrgPerson)";)
#
# hide group members when they are marked hidden
#
aci: (targetattr="uniqueMember||mgrpRfc822MailMember")
 (targetfilter=(&(objectClass=inetMailGroupManagement)
 (|(mgmanMemberVisibility=anyone)(mgmanMemberVisibility=all)))) (version
 3.0; acl "User mail list member access -
 product=ims5.0,class=nda,num=22,version=1"; allow (read,search)
 userdn="ldap:///<OrgNodeDN>??sub?(objectclass=inetOrgPerson)";)

dn: <DCNodeDN>
changetype: modify
add: aci
```

Code Example A-3 Hosted Domain ACIs

```
#
#-----------------------------------
# Domain Administrator access to iCS attributes
#
aci: (targetattr="icsTimeZone||icsMandatorySubscribed||
 icsMandatoryView||icsDefaultAccess||icsRecurrenceBound||
 icsRecurrenceDate||icsAnonymousLogin||icsAnonymousAllowWrite||
 icsAnonymousCalendar||icsAnonymousSet||icsAnonymousDefaultSet||
 icsSessionTimeout||icsAllowRights||icsExtended||
 icsExtendedDomainPrefs")(targetfilter=(objectClass=icsCalendarDomain))
 (version 3.0; acl "Domain Adm calendar access - product=ims5.0,
 class=nda,num=16,version=1"; allow (all) groupdn="ldap:///cn=Domain
 Administrators,ou=Groups,<OrgNodeDN>";)
```

Domain Organization ACIs

These need to be added to every domain organization provisioned.

Code Example A-4 Domain Organization ACIs

```
dn: <DomainOrgNodeDN>
changetype: modify
add: aci
#
# Rights to modify, add, delete users
#
aci: (target="ldap:///uid=*,ou=people,<DomainOrgNodeDN>")
 (targetattr ="*")
 (targetfilter=(objectclass=organizationalPerson))
 (version 3.0; acl "Domain Organization Admin User add,delete,write -
 product=ims5.0,class=nda,num=201,version=1";
 allow (add,write,delete)
 groupdn="ldap:///cn=Domain Organization
 Administrators,<DomainOrgNodeDN>";)
#
# Rights to modify, add, delete mailing lists.
#
aci: (target="ldap:///cn=*,ou=groups,<DomainOrgNodeDN>")
 (targetattr ="*")
 (targetfilter=(objectclass=inetMailGroup))
 (version 3.0; acl "Domain Organization Admin User add,delete,write -
 product=ims5.0,class=nda,num=202,version=1";
 allow (add,write,delete)
 groupdn="ldap:///cn=Domain Organization
 Administrators,<DomainOrgNodeDN>";)
```

Glossary

A record A type of DNS record containing a host name and its associated IP address. A records are used by messaging servers on the Internet to route email. *See also* **Domain Name System (DNS)** and **MX record**.

access control A method for controlling access to a server or to folders and files on a server.

access 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.

access domain Limits access to certain Messaging Server operations from within a specified domain. For example, an access domain can be used to limit where mail for an account can be collected.

account Information that defines a specific user or user group. This information includes the user or group name, valid email address or addresses, and how and where email is delivered.

address Information in an email message that determines where and how the message must be sent. Addresses are found both in message headers and in message envelopes. Envelope addresses determine how the message gets routed and delivered; header addresses are present merely for display purposes.

address handling The actions performed by the MTA to detect errors in addressing, to rewrite addresses if necessary, and to match addresses to recipients.

addressing protocol The addressing rules that make email possible. RFC 822 is the most widely used protocol on the Internet and the protocol supported by iPlanet Messaging Server. Other protocols include X.400 and UUCP (UNIX to UNIX Copy Protocol).

address token The address element of a rewrite rule pattern.

admin Administrator or administrative.

administration privileges The set of privileges that define a users administrative role.

administration console See **Console**.

administration server administrator User who has administrative privileges to start or stop a server even when there is no Directory Server connection. The administration server administrator has restricted server tasks (typically only Restart Server and Stop Server) for all servers in a local server group. When an administration server is installed, this administrator's entry is automatically created locally (this administrator is not a user in the user directory).

administrator A user with a defined set of administrative privileges. See also **configuration administrator**, **Directory Manager**, **administration server administrator**, **server administrator**, **message store administrator**, **top-level administrator**, **domain administrator**, **organization administrator**, **family group administrator**, **mailing list owner**.

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.

Allow filter A Messaging Server access-control rule that identifies clients that are to be allowed access to one or more of the following services: POP, IMAP, or HTTP. Compare **Deny filter**.

alternate address A secondary address for an account, generally a variation on the primary address. In some cases it is convenient to have more than one address for a single account.

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.

AUTH An SMTP command enabling an SMTP client to specify an authentication method to the server, perform an authentication protocol exchange, and, if necessary, negotiate a security layer for subsequent protocol interactions.

authentication (1) The process of proving the identity of a client user to iPlanet Messaging Server. (2) The process of proving the identity of iPlanet Messaging Server to a client or another server.

authentication certificate A digital file sent from server to client or client to server to verify and authenticate the other party. The certificate ensures the authenticity of its holder (the client or server). Certificates are not transferable.

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

AutoReply utility A utility that automatically responds to messages sent to accounts with the AutoReply feature activated. Every account in iPlanet Messaging Server can be configured to automatically reply to incoming messages.

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.

backend server An email server whose only function is to store and retrieve email messages. Also called a message store server.

backup The process of backing up the contents of folders from the message store to a backup device. See also **restore**.

banner A text string displayed by a service such as IMAP when a client first connects to it.

base DN A distinguished name entry in the directory from which searches will occur. Also known as a search base. For example, ou=people, o=siroe.com.

Berkeley DB A transactional database store intended for high-concurrency read-write workloads, and for applications that require transactions and recoverability. iPlanet Messaging Server uses Berkeley databases for numerous purposes.

bind DN A distinguished name used to authenticate to the Directory Server when performing an operation.

body One part of an email message. Although headers and envelopes must follow a standard format, the body of the message has a content determined by the sender—the body can contain text, graphics, or even multimedia. Structured bodies follow the MIME standard.

capability A string, provided to clients, that defines the functionality available in a given IMAP service.

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

Certificate Authority See **CA**.

certificate-based authentication Identification of a user from a digital certificate submitted by the client. Compare **password authentication**.

certificate database A file that contains a server's digital certificate(s). Also called a certificate file.

certificate name The name that identifies a certificate and its owner.

channel The fundamental MTA component that processes a message. A channel represents a connection with another computer system or group of systems. Each channel consists of one or more channel programs and an outgoing message queue for storing messages that are destined to be sent to one or more of the systems associated with the channel. See also **channel block**, **channel host table, channel program**.

channel block A single channel definition. See also channel host table.

channel host table The collective set of channel definitions.

channel program Part of a channel that performs the following functions: (1) transmits messages to remote systems and deletes messages from the queue after they are sent and (2) accepts messages from remote systems placing them in the appropriate channel queues. See also **master channel program**, **slave channel program**.

ciphertext Text that has been encrypted. Opposite of **cleartext**.

cipher An algorithm used in encryption.

CLI Command Line Interface.

client A software entity that requests services or information from a server.

CNAME record A type of DNS record that maps a domain name alias to a domain name.

cleartext Unencrypted text.

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

cn LDAP alias for common name.

comment character A character that, when placed at the beginning of a line, turns the line into a nonexecutable comment.

config Configuration.

configuration administrator Person who has administrative privileges to manage servers and configuration directory data in the entire iPlanet topology. The configuration administrator has unrestricted access to all resources in the iPlanet topology. This is the only administrator who can assign server access to other administrators. The configuration administrator initially manages administrative configuration until the administrators group and its members are in place.

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

Configuration Directory Server A Directory Server that maintains configuration information for a server or set of servers.

configutil A command-line utility for making changes to various configuration parameters stored in the directory server or in the local configuration file, configdb.

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.

Console A GUI (graphical user interface) that enables you to configure, monitor, maintain, and troubleshoot many iPlanet components.

cookie 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.

counterutil A command-line utility for displaying all counters in a counter object.

cronjob UNIX only. A task that is executed automatically by the cron daemon at a configured time. See **crontab file**.

crontab file UNIX only. A list of commands, one per line, that executes automatically at a given time.

daemon A UNIX program that runs in the background, independent of a terminal, and performs a function whenever necessary. Common examples of daemon programs are mail handlers, license servers, and print daemons. On Windows NT machines, this type of program is called a service. See also **service**.

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 a DC Tree would be cn=billbob,dc=bridge,dc=net,o=internet.

defragmentation The Multipurpose Internet Mail 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 for Messaging. A set of interfaces (GUI and CLI) that allow domain administrators to add and modify users and groups to a hosted domain.

Delegated Administrator Console A web browser-based software console that allows domain 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 administrator server A daemon program that handles access control to the directory by hosted domains.

delete message The act of marking a message for deletion. The deleted message is not removed from the message store until it is expunged or purged in a separate action by the user. See also **purge message**, **expunge message**.

deliver A command-line utility that delivers mail directly to the message store accessible by POP, IMAP, or HTTP mail clients.

delivery See **message delivery**.

delivery status notification A message giving status information about a message in route to a recipient. For example, a message indicating that delivery has been delayed because of network outages.

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.

Deny filter A Messaging Server access-control rule that identifies clients that are to be denied access to one or more of the following services: POP, IMAP, or HTTP. Compare **Allow filter**.

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.

directory context The point in the directory tree information at which a search begins for entries used to authenticate a user and password for message store access. See also **base DN**.

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.

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 lookup The process of searching the directory for information on a given user or resource, based on that user or resource's name or other characteristic.

Directory Manager User who has administrative privileges to the directory server database. Access control does not apply this user (think of the directory manager as the directory's superuser).

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

Directory Server The iPlanet directory service based on LDAP. See also **directory service**, **Lightweight Directory Access Protocol**, **Configuration Directory Server**, **User/Groups Directory Server**.

directory service A logically centralized repository of information about people and resources within an organization. See also **Lightweight Directory Access Protocol**.

directory synchronization The process of updating—that is, synchronizing—the MTA directory cache with the current directory information stored in the directory service. See also **MTA directory cache**.

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

Dispatcher The MTA component that handles connection requests for defined TCP ports. The Dispatcher is a multi-threaded connection dispatching agent that permits multiple multi-threaded servers to share responsibility for a given service. When using the Dispatcher, it is possible to have several multi-threaded SMTP server processes running concurrently.

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 mailing list or group. See also **expansion**, **member**, **moderator**, 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 See distinguished name.

dn LDAP alias for distinguished name. See also **distinguished name**.

DNS See **Domain Name System**.

DNS alias A host name that the DNS server recognizes as pointing to a different host—specifically a DNS CNAME record. Machines always have one real name, but they can have one or more aliases. For example, www.siroe.domain might be an alias that points to a real machine called realthing.siroe.domain where the server currently exists.

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

DNS spoofing A form of network attack in which a DNS server has been subverted to provide false information.

domain 1) A group of computers whose host names 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, corp.mktng.siroe.com. 2) A region of administrative control.

domain administrator User who has administrative privileges to create, modify, and delete mail users, mailing lists, and family accounts in a hosted domain by using the Delegated Administrator for Messaging GUI or CLIs. By default, this user can act as a message store administrator for all messaging servers in the topology.

domain alias A domain entry that points to another domain. By using aliases, hosted domains can have several domain names.

domain hosting The ability to host one or more domains on a shared messaging server. For example, the domains siroe.com and sesta.org might both be hosted on the siroe.net mail server. Users send mail to and receive mail from the hosted domain—the name of the mail server does not appear in the email address.

domain name (1) A host name used in an email address. (2) A unique name that defines an administrative organization. Domains can contain other domains. Domain names are interpreted from right to left. For example, siroe.com is both the domain name of the Siroe Company and a subdomain of the top-level com domain. The siroe.com domain can be further divided into subdomains such as corp.siroe.com, and so on. See also **host name** and **fully-qualified domain name**.

Domain Name System (DNS) A distributed name resolution software that allows computers to locate other computers on a network or the Internet by domain name. The system associates standard IP addresses with host names (such as www.siroe.com). Machines normally get this information from a DNS server. DNS servers provide a distributed, replicated, data query service for translating hostnames into Internet addresses. See also **A record**, **MX record**, **CNAME record**.

domain organization A sub-domain below a hosted domain in the Organization Tree. Domain organizations are useful for companies that wish to organize their user and group entries along departmental lines.

domain part The part of an email address to the right of the @ sign. For example, siroe.com is the domain part of the email address dan@siroe.com.

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

domain rewrite rules See **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.

DSN. See **Delivery Status Notification**.

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

dssetup A Directory Server preparation tool that makes an existing Directory Server ready for use by an iPlanet Messaging Server.

dynamic group A mail group defined by an LDAP search URL. Users usually join the group by setting an LDAP attribute in their directory entry.

EHLO command An SMTP command that queries a server to find out if the server supports extended SMTP commands. Defined in RFC 1869.

encryption The process of disguising information so that it cannot be deciphered (decrypted) by anyone but the intended recipient who has the code key.

enterprise network A network that consists of collections of networks connected to each other over a geographically dispersed area. The enterprise network serves the needs of a widely distributed company and is used by the company's mission-critical applications.

envelope A container for transport information about the sender and the recipient of an email message. This information is not part of the message header. Envelopes are used by various email programs as messages are moved from place to place. Users see only the header and body of a message.

envelope field A named item of information, such as RCPT TO, in a message envelope.

error handler A program that handles errors. In Messaging Server, issues error messages and processes error action forms after the postmaster fills them out.

Error-Handler Action form A form sent to the postmaster account that accompanies a received message that Messaging Server cannot handle. The postmaster fills out the form to instruct the server how to process the message.

error message A message reporting an error or other situation. iPlanet Messaging Server generates messages in a number of situations, notably when it gets an email message that it can't handle. Others messages, called notification errors, are for informational purposes only.

ESP Enterprise Service Provider.

ESMTP See **Extended Simple Mail Transfer Protocol**.

ETRN An SMTP command enabling a client to request that the server start the processing of its mail queues for messages that are waiting at the server for the client machine. Defined in RFC 1985.

expander Part of an electronic mail delivery system that 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. See also **EXPN**.

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

EXPN An SMTP command for expanding a mailing list. Defined in RFC 821.

expunge message The act of marking a message for deletion and then permanently removing it from the INBOX. See also **delete message**, **purge message**.

Extended Simple Mail Transfer Protocol (ESMTP) An Internet message transport protocol. ESMTP adds optional commands to the SMTP command set for enhanced functionality, including the ability for ESMTP servers to discover which commands are implemented by the remote site.

extranet The part of a company intranet that customers and suppliers can access. See also **intranet**.

facility In a Messaging Server log-file entry, a designation of the software subsystem (such as Network or Account) that generated the log entry.

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

family group administrator User who has administrative privileges to add and remove family members in a family group. This user can grant family group administrative access to other members of group.

firewall A network configuration, usually both hardware and software, that forms a barrier between networked computers within an organization and those outside the organization. A firewall is commonly used to protect information such as a network's email, discussion groups, and data files within a physical building or organization site.

folder A named collection of messages. Folders can contain other folders. Also called a mailbox. See also **personal folder**, **shared folder, INBOX**.

forwarding See **message forwarding**.

FQDN See **fully-qualified domain name**.

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

fully-qualified domain name (FQDN) The unique name that identifies a specific Internet host. See also **domain name**.

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.

greeting form A message usually sent to users when an account is created for them. This form acts as confirmation of the new account and verification of its contents.

group A group of LDAP mail entries that are organized under a distinguished name. Usually used as a distribution list, but may also be used to grant certain administrative privileges to members of the group. See also **dynamic group**, **static group**.

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

GUI Graphical User Interface

HA See **High Availability**.

hashdir A command-line utility for determining which directory contains the message store for a particular user.

header The portion of an email message that precedes the body of the message. The header is composed of field names followed by a colon and then values. Headers contain information useful to email programs and to users trying to make sense of the message. For example, headers include delivery information, summaries of contents, tracing, and MIME information; they tell whom the message is for, who sent it, when it was sent, and what it is about. Headers must be written according to RFC 822 so that email programs can read them.

header field A named item of information, such as From: or To:, in a message header. Often referred to as a “header line”.

High Availability Enables the detection of a service interruption and provides recovery mechanisms in the event of a system failure or process fault. In addition, it allows a backup system to takes over the services in the event of a primary system failure.

hop A transmission between two computers.

host The machine on which one or more servers reside.

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 Messaging Server host with other hosted domains. In earlier LDAP-based email systems, a domain was supported by one or more email server hosts. With Messaging Server, many domains can be hosted on a single server. For each hosted domain, there is an LDAP entry that points to the user and group container for the domain. Hosted domains are also called virtual hosted domains or virtual domains.

host name The name of a particular machine within a domain. The host name is the IP host name, which might be either a "short-form" host name (for example, mail) or a fully qualified host name. The fully qualified host name consists of two parts: the host name and the domain name. For example, mail.siroe.com is the machine mail in the domain siroe.com. Host names must be unique within their domains. Your organization can have multiple machines named mail, as long as the machines reside in different subdomains; for example, mail.corp.siroe.com and mail.field.siroe.com. Host names always map to a specific IP address. See also **domain name**, **fully-qualified domain name**, and **IP address**.

host name hiding The practice of having domain-based email addresses that don't contain the name of a particular internal host.

HTTP See **HyperText Transfer Protocol**.

hub A host that acts as the single point of contact for the system. When two networks are separated by a firewall, for example, the firewall computer often acts as a mail hub.

HyperText Transfer Protocol A standard protocol that allows the transfer of hypertext documents over the Web. iPlanet Messaging Server provides an HTTP service to support web-based email. See **Messenger Express**.

iDA iPlanet Delegated Administrator for Messaging.

IDENT See **Identification Protocol**.

Identification Protocol A protocol that provides a means to determine the identity of a remote process responsible for the remote end of a particular TCP connection. Defined in RFC 1413.

IMAP4 See **Internet Message Access Protocol Version 4**.

imsadmin A set of command line utilities for managing domain administrators, users, and groups.

imsasm A utility that handles the saving and recovering of user mailboxes. The imsasm utility invokes the imsbackup and imsrestore utilities to create and interpret a data stream.

imsbackup A command-line utility for backing up the message store.

imsimta commands A set of command line utilities for performing various maintenance, testing, and management tasks for the Message Transfer Agent (MTA).

imsrestore A command-line utility for restoring the message store.

imscripter A command-line utility that talks to an IMAP server. You can use this utility to execute a command or batch of commands on IMAP folders.

INBOX The name reserved for a user's default mailbox for mail delivery. INBOX is the only folder name that is case-insensitive. For example: INBOX, Inbox, and inbox are all valid names for a users default mailbox.

installation directory The directory into which the binary (executable) files of a server are installed. For the Messaging Server, it is a subdirectory of the server root: *ServerRoot* / bin / msg / . Compare **instance directory**, **server root**.

instance A separately executable configuration of a server or other software entity on a given host. With a single installed set of binary files, it is possible to create multiple instances of iPlanet servers that can be run and accessed independently of each other.

instance directory The directory that contains the files that define a specific instance of a server. For the Messaging Server, it is a subdirectory of the server root: *ServerRoot* / msg-*InstanceName* / , where *InstanceName* is the name of the server as specified at installation. Compare **installation directory**, **server root**.

Internet The name given to the worldwide network of networks that uses TCP/IP protocols.

Internet Message Access Protocol Version 4 (IMAP4) A standard protocol that allows users to be disconnected from the main messaging system and still be able to process their mail. The IMAP specification allows for administrative control for these disconnected users and for the synchronization of the users' message store once they reconnect to the messaging system.

Internet Protocol (IP) The basic network-layer protocol on which the Internet and intranets are based.

internet protocol address See **IP address**.

intranet A network of TCP/IP networks within a company or organization. Intranets enable companies to employ the same types of servers and client software used for the World Wide Web for internal applications distributed over the corporate LAN. Sensitive information on an intranet that communicates with the Internet is usually protected by a firewall. See also **firewall** and **extranet**.

invalid user An error condition that occurs during message handling. When this occurs, the message store sends a communication to the MTA, the message store deletes its copy of the message. The MTA bounces the message back to the sender and deletes its copy of the message.

IP See **Internet Protocol**.

IP address A set of numbers, separated by dots, such as 198.93.93.10, that specifies the actual location of a machine on an intranet or the Internet. A 32-bit address assigned to hosts using TCP/IP.

iPlanet Setup The installation program for all iPlanet servers and for iPlanet Console.

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 The MTA component responsible for scheduling and executing tasks upon request by various other MTA components.

key database A file that contains the key pair(s) for a server's certificate(s). Also called a key file.

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

LDAP See **Lightweight Directory Access Protocol**.

LDAP Data Interchange Format (LDIF) The format used to represent Directory Server entries in text form.

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 search string A string with replaceable parameters that defines the attributes used for directory searches. For example, an LDAP search string of "uid=%s" means that searches are based on the user ID attribute.

LDAP Server A software server that maintains an LDAP directory and services queries to the directory. The iPlanet 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 See **LDAP Data Interchange Format**.

Legato Networker A third-party backup utility distributed by Legato.

level A designation of logging verbosity, meaning the relative number of types of events that are recorded in log files. At a level of Emergency, for example, very few events are logged; at a level of Informational, on the other hand, very many events are logged.

Lightweight Directory Access Protocol (LDAP) Directory service protocol designed to run over TCP/IP and across multiple platforms. A simplification of the X.500 Directory Access Protocol (DAP) that allows a single point of management for storage, retrieval, and distribution of information, including user profiles, distribution lists, and configuration data across iPlanet servers. The iPlanet Directory Server uses the LDAP protocol.

listen port The port that a server uses to communicate with clients and other servers.

local part The part of an email address that identifies the recipient. See also **domain part**.

log directory The directory in which all of a service's log files are kept.

log expiration Deletion of a log file from the log directory after it has reached its maximum permitted age.

log rotation Creation of a new log file to be the current log file. All subsequent logged events are to be written to the new current file. The log file that was the previous current file is no longer written to, but remains in the log directory.

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

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

mail client The programs that help users send and receive email. This is the part of the various networks and mail programs that users have the most contact with. Mail clients create and submit messages for delivery, check for new incoming mail, and accept and organize incoming mail.

mail exchange record See **MX record**.

mailing list A list of email addresses to which a message can be sent by way of a mailing list address. Sometimes called a group.

mailing list owner A user who has administrative privileges to add members to and delete members from the mailing list.

mail relay A mail server that accepts mail from a MUA or MTA and relays it to the mail recipient's message store or another router.

mail router See **mail relay**.

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

master channel program A channel program that typically initiates a transfer to a remote system. See also **slave channel program**.

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

mboxutil A command-line utility for managing mail folders. This utility lists, creates, deletes, renames, or moves mailboxes (folders). It can also be used to report quota information.

MD5 A message digest algorithm by RSA Data Security. MD5 can be used to produce a short digest of data that is unique with high probability. It is mathematically extremely hard to produce a piece of data that produces the same message digest email.

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 The fundamental unit of email, a message consists of a header and a body and is often contained in an envelope while it is in transit from the sender to the recipient.

message access services The protocol servers, software drivers, and libraries that support client access to the Messaging Server message store.

message delivery The act that occurs when an MTA delivers a message to a local recipient (a mail folder or a program).

message forwarding The act that occurs when an MTA sends a message delivered to a particular account to one or more new destinations as specified by the account's attributes. Forwarding may be configurable by the user. See also **message delivery**, **message routing**.

message routing The act of transferring a message from one MTA to another when the first MTA determines that the recipient is not a local account, but might exist elsewhere. Routing is normally configurable only by a network administrator. See also **message forwarding**.

Message Handling System (MHS) A group of connected MTAs, their user agents, and message stores.

message queue The directory where messages accepted from clients and other mail servers are queued for delivery (immediate or deferred).

message quota A limit defining how much disk space a particular folder can consume.

message store The database of all locally delivered messages for a Messaging server instance. Messages can be stored on a single physical disk or stored across multiple physical disks.

message store administrator User who had administrative privileges to manage the message store for a Messaging Server installation. This user can view and monitor mailboxes, and specify access control to the store. Using proxy authorization rights, this user can run certain utilities for managing the store.

message store partition A message store or subset of a message store residing on a single physical file system partition.

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

Message Transfer Agent (MTA) A specialized program for routing and delivering messages. MTAs work together to transfer messages and deliver them to the intended recipient. The MTA determines whether a message is delivered to the local message store or routed to another MTA for remote delivery.

Messaging Multiplexor A specialized iPlanet Messaging Server that acts as a single point of connection to multiple mail servers, facilitating the distribution of a large user base across multiple mailbox hosts.

Messaging Server administrator The administrator whose privileges include installation and administration of an iPlanet Messaging Server instance.

Messenger Express A mail client that enables users to access their mailboxes through a browser-based (HTTP) interface. Messages, folders, and other mailbox information are displayed in HTML in a browser window. See also **webmail**.

mkbackupdir A utility that creates and synchronizes the backup directory with the information in the message store. It is used in conjunction with Legato Networker.

MHS See **Message Handling System**.

MIME See **Multipurpose Internet Mail Extension**.

MMP See Messaging Multiplexor.

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**, and **member**.

MoveUser A command-line utility for moving messages in a user's mail folder from one Messaging Server to another.

MTA See **Message Transfer Agent**.

MTA configuration file The file (imta.cnf) that contains all channel definitions for the Messaging Server as well as the rewrite rules that determine how addresses are rewritten for routing. See also **channel** and **rewrite rule**.

MTA directory cache a snapshot of the directory service information about users and groups required by the MTA to process messages. See also **directory synchronization**.

MTA hop The act of routing a message from one MTA to another.

MUA See **user agent**.

Multiplexor See **Messaging Multiplexor**.

Multipurpose Internet Mail Extension (MIME) A protocol you can use to include multimedia in email messages by appending the multimedia file in the message.

MX record Mail Exchange Record. A type of DNS record that maps one host name to another.

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

namespace The tree structure of an LDAP directory. See **directory information tree**.

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 iPlanet Directory Server, specific types of directory information are stored in naming contexts. For example, a naming context which stores all entries for marketing employees in the Siroe Corporation at the Boston office might be called ou=mktg, ou=Boston, o=Siroe, c=US.

NDN See **nondelivery notification**.

next-hop list A list of adjacent systems a mail route uses to determine where to transfer a message. The order of the systems in the next-hop list determines the order in which the mail route transfers messages to those systems.

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.

NMS Netscape Messaging Server.

node A domain entry in the DIT.

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

notary messages Nondelivery notifications (NDNs) and delivery status notifications (DSNs) that conform to the NOTARY specifications RFC 1892.

notification message A type of message, sent to the postmaster account by the Messaging Server, that is for informational purposes and requires no action from the postmaster. Compare **error message**.

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

off-line state A state in which the mail client downloads messages from a server system to a client system where they can be viewed and answered. The messages might or might not be deleted from the server.

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

organization administrator User who had administrative privileges to create, modify, and delete mail users and mailing lists in an organization or sub-organization by using the Delegated Administrator for Messaging GUI or CLIs.

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.

partition See **message store partition**.

password authentication Identification of a user through user name and password. Compare certificate-based authentication.

pattern A string expression used for matching purposes, such as in Allow and Deny filters.

permanent failure An error condition that occurs during message handling. When this occurs, the message store deletes its copy of an email message. The MTA bounces the message back to the sender and deletes its copy of the message.

personal folder A folder that can be read only by the owner. See also **shared folder**.

plaintext Refers to a method for transmitting data. The definition depends on the context. For example, with SSL plaintext passwords are encrypted and are therefore not sent as cleartext. With SASL, plaintext passwords are hashed, and only a hash of the password is sent as text. See also **SSL** and **SASL**.

plaintext authentication See **password authentication**.

POP3 See **Post Office Protocol Version 3**.

port number A number that specifies an individual TCP/IP application on a host machine, providing a destination for transmitted data.

postmaster account An alias for the email group and email addresses who receive system-generated messages from the Messaging Server. The postmaster account must point to a valid mailbox or mailboxes.

Post Office Protocol Version 3 (POP3) A protocol that provides a standard delivery method and that does not require the message transfer agent to have access to the user's mail folders. Not requiring access is an advantage in a networked environment, where often the mail client and the message transfer agent are on different computers.

process A self-contained, fully functional execution environment set up by an operating system. Each instance of an application typically runs in a separate process. Compare **thread**.

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 iPlanet Directory Server. These entries include users and groups and domain information.

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 message 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 message store file system. See also **delete message**, **expunge message**.

queue See **message queue**.

RC2 A variable key-size block cipher by RSA Data Security.

RC4 A stream cipher by RSA Data Security. Faster than RC2.

readership A command-line utility for collecting readership information on shared mail folders.

reconstruct A command-line utility for reconstructing mail folders.

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**.

regular expression A text string that uses special characters to represent ranges or classes of characters for the purpose of pattern matching.

relaying The process of passing a message from one messaging server to another messaging 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.

restore The process of restoring the contents of folders from a backup device to the message store. See also **backup**.

reverse DNS lookup The process of querying the DNS to resolve a numeric IP address into the equivalent fully qualified domain name.

rewrite rules Also known as domain rewrite rules. A tool that the MTA 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 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 matrix.” In OSI terminology, a router is a Network Layer intermediate system. See also **gateway**.

routing See **message routing**.

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.

SASL See **Simple Authentication and Security Layer**.

schema Definitions—including structure and syntax—of the types of information that can be stored as entries in iPlanet Directory Server. When information that does not match the schema is stored in the directory, clients attempting to access the directory might be unable to display the proper results.

SCM See **Service Control Manager**.

search base See **base DN**.

Secure Sockets Layer (SSL) A software library establishing a secure connection between two parties (client and server).

security-module database A file that contains information describing hardware accelerators for SSL ciphers. Also called secmod.

sendmail A common MTA used on UNIX machines. In most applications, iPlanet Messaging Server can be used as a drop-in replacement for sendmail.

server administrator Person who performs server management tasks. The server administrator provides restricted access to tasks for a particular server, depending upon task ACIs. The configuration administrator must assign user access to a server. Once a user has server access permissions, that user is a server administrator who can provide server access permissions to users.

server instance The directories, programs, and utilities representing a specific server installation.

server root The directory into which all iPlanet servers associated with a given Administration Server on a given host are installed. Typically designated *ServerRoot*. Compare **installation directory**, **instance directory**.

server side rules (SSR) A set of rules for enabling server-side filtering of mail. Based on the Sieve mail filtering language.

service (1) A function provided by a server. For example, iPlanet Messaging Server provides SMTP, POP, IMAP, and HTTP services. (2) A background process on Windows NT that does not have a user interface. iPlanet servers on Windows NT platforms run as services. Equivalent to **daemon**.

Service Control Manager Windows NT administrative program for managing services.

session An instance of a client-server connection.

shared folder A folder that can be read by more than one person. Shared folders have an owner who can specify read access to the folder and who can delete messages from the shared folder. The shared folder can also have a moderator who can edit, block, or forward incoming messages. Only IMAP folders can be shared. Compare **personal folder**.

Sieve A proposed language for filtering mail.

Simple Authentication and Security Layer (SASL) A means for controlling the mechanisms by which POP, IMAP or SMTP clients identify themselves to the server. iPlanet Messaging Server support for SMTP SASL use complies with RFC 2554 (ESMTP AUTH). SASL is defined in RFC 2222.

Simple Mail Transfer Protocol (SMTP) The email protocol most commonly used by the Internet and the protocol supported by the iPlanet Messaging Server. Defined in RFC 821, with associated message format descriptions in RFC 822.

SIMS Sun Internet Mail Server

single field substitution string In a rewrite rule, part of the domain template that dynamically rewrites the specified address token of the host / domain address. See also **domain template**.

single sign-on. The ability for a user to authenticate once and gain access to multiple services (mail, directory, file services, and so on).

SIZE An SMTP extension enabling a client to declare the size of a particular message to a server. The server may indicate to the client that it is or is not willing to accept the message based on the declared message size; the server can declare the maximum message size it is willing to accept to a client. Defined in RFC 1870.

slave channel program A channel program that accepts transfers initiated by a remote system. See also **master channel program**.

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

SMTP See **Simple Mail Transfer Protocol**.

SMTP AUTH See **AUTH**.

sn Aliased directory attribute for surname.

spoofing A form of network attack in which a client attempting to access or send a message to a server misrepresents its host name.

SSL See **Secure Sockets Layer**.

SSR See **Server Side Rules**.

static group A mail group defined statically by enumerating each group member. See also **dynamic group**.

stored A command-line utility that performs daily maintenance tasks on the message store. This utility expunges and erases messages stored on disk.

subdomain A portion of a domain. For example, in the domain name corp.siroe.com, corp is a subdomain of the domain siroe.com. See also **host name** and **fully-qualified domain name**.

subnet The portion of an IP address that identifies a block of host IDs.

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

synchronization (1) The update of data by a master directory server to a replica directory server. (2) The update of the MTA directory cache.

TCP See **Transmission Control Protocol**.

TCP/IP See **Transmission Control Protocol/Internet Protocol**.

thread A lightweight execution instance within a process.

TLS See **Transport Layer Security**.

top-level administrator User who has administrative privileges to create, modify, and delete mail users, mailing lists, family accounts, and domains in an entire Messaging Server namespace by using the Delegated Administrator for Messaging GUI or CLIs. By default, this user can act as a message store administrator for all messaging servers in the topology.

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

Transmission Control Protocol (TCP) The basic transport protocol in the Internet protocol suite that provides reliable, connection-oriented stream service between two hosts.

Transmission Control Protocol/Internet Protocol (TCP/IP) The name given to the collection of network protocols used by the Internet protocol suite. The name refers to the two primary network protocols of the suite: TCP (Transmission Control Protocol), the transport layer protocol, and IP (Internet Protocol), the network layer protocol.

Transport Layer Security (TLS). The standardized form of SSL. See also **Secure Sockets Layer**.

transport protocols Provides the means to transfer messages between MTAs, for example SMTP and X.400.

UA See **user agent**.

UBE See **Unsolicited Bulk Email**.

UID (1) User identification. A unique string identifying a user to a system. Also referred to as a userID. (2) Aliased directory attribute for userID (login name).

unified messaging The concept of using a single message store for email, voicemail, fax, and other forms of communication. iPlanet Messaging Server provides the basis for a complete unified messaging solution.

Unsolicited Bulk Email (UBE) Unrequested and unwanted email, sent from bulk distributors, usually for commercial purposes.

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 account An account for accessing a server, maintained as an entry on a directory server.

user agent (UA) The client component, such as Netscape Communicator, that allows users to create, send, and receive mail messages.

User/Groups Directory Server A Directory Server that maintains information about users and groups in an organization.

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, and so on.

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.

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

vanity domain A domain name associated with an individual user—not with a specific server or hosted domain. A vanity domain is specified by using the MailAlternateAddress attribute. The vanity domain does not have an LDAP entry for the domain name. Vanity domains are useful for individuals or small organizations desiring a customized domain name, without the administration overhead of supporting their own hosted domain. Also called custom domain.

/var/mail A name often used to refer to Berkeley-style inboxes in which new mail messages are stored sequentially in a single, flat text file.

Veritas Cluster Server High availability clustering software from Veritas Software with which iPlanet Messaging Server can integrate.

virtual domain (1) An ISP hosted domain. See also **hosted domain**. (2) A domain name added by the Messaging Multiplexor to a client's user ID for LDAP searching and for logging into a mailbox server.

VRFY An SMTP command for verifying a user name. Defined in RFC 821.

webmail A generic term for browser-based email services. A browser-based client—known as a “thin” client because more processing is done on the server—accesses mail that is always stored on a server. See also **Messenger Express**.

wildcard A special character in a search string that can represent one or more other characters or ranges of characters.

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

SYMBOLS

A

B

C

L

M

N

O

P

R

S

T

U

V

W