Zimbra Collaboration Suite
Administrator’s Guide

Beta 2005
August 2005

Building Better Products within the Open Source Community

Zimbra Collaboration Suite leverages many great technologies from the open source community: MySQL, OpenLDAP, Postfix, SpamAssassin, and Apache. Zimbra believes that great products come from contributing to and leveraging open source technologies. We are thankful for the great contributions that led to the creation of MySQL, OpenLDAP, Postfix, SpamAssassin, and Apache.

Trademark and Licensing

Zimbra, Inc.
1500 Fashion Island Boulevard, Suite 100
San Mateo, California 94004
650. 212.0505
www.zimbra.com

Table of Content

Chapter 1 Introduction

Zimbra is a full-featured mail system offering reliable, high-performance service and advanced mail features including advanced search capability, mail sorted by conversations, calendar, and tags, as well as standard mail features such as contacts, user-defined folders, and user-defined filters.

Intended Audience

This guide is intended for systems administrators responsible for installing, maintaining, and supporting the server deployment of Zimbra.

Readers of this guide should already possess the following recommended knowledge and skill sets:

- Familiarity with the associated technologies and standards, including Red Hat® Enterprise Linux® operating system and open source concepts
- Industry practices for mail system management

Available Documentation

Zimbra includes the following documentation titles:

- *Installation Guide*. This guide describes how to prepare for installing the Zimbra server application, including system requirements, server configurations, and testing your installation. Also included is the instructions for running the Migration Wizard to migrate accounts from the Microsoft Exchange server. This guide is provided in PDF format.
- *Administrator Guide*. This guide provides a comprehensive product overview, including architecture, server functionality, administration tasks, configuration options, and troubleshooting. This guide is provided in PDF format and is available from the administration console.
- *Zimbra administration console Help*. Describes how to use the system administrator console.
- *Zimbra Web Client Help*. Describes how to use the end-user interface.
- *Release Notes*. Late-breaking news for product releases are contained in Release Notes, provided as a **readme** file in the installation package.

Support for Recommended Third-Party Components

Where possible, Zimbra adheres to existing industry standards and open source implementations for backup management, user authentications, operating platform, and database management. However, Zimbra only supports the specific implementations described in the Zimbra Collaboration Suite architecture overview in the Product Overview chapter as officially tested and certified for the Zimbra Collaboration Suite. This document may occasionally note when other tools are available in the marketplace, but such mention does not constitute an endorsement or certification.

Support and Contact Information

The Zimbra Collaboration Suite is currently in Beta and we appreciate your feedback and suggestions. Visit **www.zimbra.com** to join the community and to be a part of building the best open source messaging solution.

- Click **Feedback** to send us an email. Let us know what at you like about the product, and what you would like to see in the product.
- Join the Zimbra Community Forum, to participate and learn more about the Zimbra Collaboration Suite.

If you encounter problems with this beta software, visit Zimbra.com and submit a bug and make sure to provide enough detail so that it can be easily duplicated.

Chapter 2 Product Overview

The Zimbra Collaboration Suite is a full-featured collaboration solution offering reliable, high-performance services and advanced mail features. This chapter describes the Zimbra application architecture, integration points, and information flow.

Overview

The Zimbra Collaboration Suite is designed to provide an end-to-end mail solution that is scalable and highly reliable. The messaging architecture is built with well-known open-system technology and standards and is composed of a mail server application and a client interface.

The architecture includes the following core advantages:

- **Open source integrations**. Linux®, Apache Tomcat, Postfix, MySQL®, OpenLDAP®.
- **Uses industry standard open protocols**. SMTP, LMTP, SOAP, XML, IMAP, POP.
- **Modern technology design**. Java, JavaScript thin client, DHTML.
- **Horizontal scalability**. Because each mailbox server includes its own data store, message store, and set mailbox accounts, you don't change anything on existing servers in order to scale the system. To scale for additional mail accounts, add more servers.
- Browser based client interface.
- Administration console to manage accounts and servers.

Core Functionality

The Zimbra Collaboration Suite offers a robust set of features. The core functionality within the Zimbra Collaboration Suite is as follows:

- Mail delivery and storage
- Indexing of mail messages upon delivery
- Backup services
- Mailbox server logging

- IMAP and POP support
- Mail delivery and routing
- Directory services
- Anti-spam protection
- Anti-virus protection

Administrators can easily manage domains, servers, and accounts from the browser based administration console and can manage backup and bulk provision accounts from the Command Line Utility.

- Move mailboxes from one server to another
- Import Microsoft Exchange user accounts
- Add accounts and domains
- Set account restrictions either for an individual account or by COS
- Manage servers
- Backup and restore
- Monitor usage

The Zimbra Web client mail features include the ability to:

- Compose, read, reply, forward, and other standard mail features
- View mail by conversation threads
- Tag mail to easily group messages for quick reference
- Use Search Builder to perform advanced searches
- Save searches
- Use the Calendar to schedule appointments
- Create a personal contact list
- Set mailbox usage preferences, including defining mail filtering options

Zimbra Components

Zimbra architecture includes open-source integrations using industry standard protocols. The third-party software listed below is bundled with Zimbra software and installed as part of the installation process. These components have been tested and configured to work with the software.

- Apache Tomcat, the web application server that Zimbra software runs in.
- Postfix, an open source message transfer agent (MTA) that routes mail messages to the appropriate Zimbra server.

- OpenLDAP software, an open source implementation of the Lightweight Directory Access Protocol (LDAP) that provides user authentication.
- MySQL database software.
- Lucene, an open-source full featured text index and search engine.
- Verity®, a third-party source that converts certain attachment file types to HTML.
- Anti-virus and anti-spam open source components including:
    - ClamAV, an anti-virus scanner that protects against malicious files.
    - SpamAssassin, a mail filter that attempts to identify spam.
    - Amavisd-new, which interfaces between the MTA and one or more content checkers.
- James/Sieve filtering, used to create filters for email.

System Architecture

Figure 1 on page 15 shows the Zimbra messaging system architectural design, including the open-source software bundled with the Suite and other recommended third-party applications.

The Zimbra Collaboration Suite includes the following application packages.

Zimbra Core

The Zimbra Core package includes the libraries, utilities, monitoring tools, and basic configuration files.

Zimbra LDAP

Zimbra messaging program uses the OpenLDAP software, an open source LDAP directory server. User authentication is provided through OpenLDAP. Each account on the Zimbra server has an unique mailbox ID that is the primary point of reference to identify the account.

The OpenLDAP schema has been customized for the Zimbra messaging program.

Zimbra MTA (mail routing server)

Postfix is the open source mail transfer agent (MTA) that receives email via SMTP and routes each message to the appropriate Zimbra mailbox server using Local Mail Transfer Protocol (LMTP). The Zimbra MTA also includes the anti-virus and anti-spam components.

Zimbra Store (Zimbra server)

The Zimbra store package installs the components for the mailbox server, including Apache Tomcat, which is the servlet container the Zimbra software runs within. Each account is configured on one mailbox server, and this account is associated with a mailbox that contains all the mail messages and file attachments for that mail account.

The mailbox server includes the following components:

- Data store
- Message store
- Index store
- HTML attachment conversion utility

Each Zimbra server has its own standalone data store, message store and index store for the mailboxes on that server.

As each mail arrives, the Zimbra server schedules a thread to have the message indexed (index store). Any attachments to the mail message are scheduled to be converted to HTML, and then the HTML version is scheduled to be indexed.

Data store. The **data store** is a MySQL database where internal mailbox IDs are linked with user accounts. The data store maps the mailbox IDs to users’ OpenLDAP accounts. This database contains each user’s set of tag definitions, folders, and contacts, as well as the status of each mail message - read, unread, tags associated to message, and folder the message resides in.

Message store. The **message store** is where all email messages and file attachments reside. Messages are stored in MIME format. A message that is sent to multiple recipients who have accounts on one mailbox server are stored only once in the file system.

Index store. Index and search technology is provided through Lucene. Index files are maintained for each mailbox.

HTML conversion. Verity converts specific attachment types to HTML at the time that each mail message is delivered to the mailbox server.

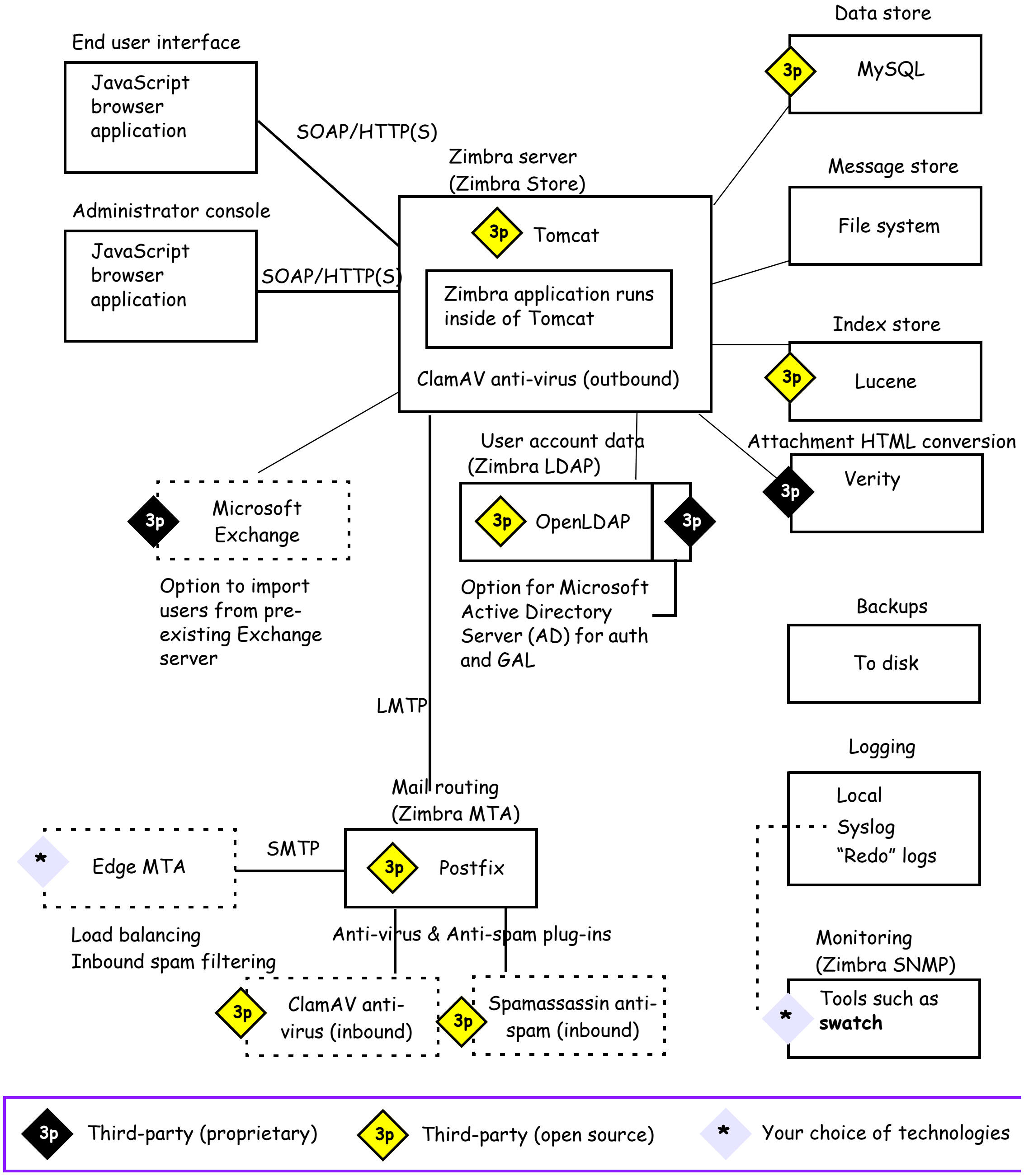
Zimbra-SNMP

Installing the Zimbra-SNMP package is optional. If you choose to install Zimbra-SNMP for monitoring, the package should be run on every server (Zimbra server, Zimbra LDAP, Zimbra MTA) that is part of the Zimbra configuration. Zimbra uses swatch to watch the syslog output to generate SNMP traps.

Backup Process Overview

Zimbra includes a configurable backup manager that resides on every Zimbra server and performs both backup and restore functions. You do not have to stop the server in order to run the backup process. You can use the backup manager to restore a single user in the event that one user's mailbox becomes corrupted. See Chapter 10, Backup and Restore.

Figure 1: Zimbra Collaboration Suite System Architecture


End user interface
JavaScript browser application
SOAP/HTTP(S)
Administrator console
JavaScript browser application
SOAP/HTTP(S)
Zimbra server (Zimbra Store)
3p
Tomcat
Zimbra application runs inside of Tomcat
ClamAV anti-virus (outbound)
Data store
3p
MySQL
Message store
File system
Index store
3p
Lucene
Attachment HTML conversion
3p
Verity
User account data (Zimbra LDAP)
3p
OpenLDAP
3p
Option for Microsoft Active Directory Server (AD) for auth and GAL
3p
Microsoft Exchange
Option to import users from pre-existing Exchange server
Backups
To disk
LMTP
Logging
Local
Syslog
"Redo" logs
Mail routing (Zimbra MTA)
*
Edge MTA
SMTP
3p
Postfix
Load balancing
Inbound spam filtering
Anti-virus & Anti-spam plug-ins
3p
ClamAV anti-virus (inbound)
3p
Spamassassin anti-spam (inbound)
Monitoring (Zimbra SNMP)
*
Tools such as swatch
3p
Third-party (proprietary)
3p
Third-party (open source)
*
Your choice of technologies

Zimbra System Directory Tree

Table 1 lists the main directories created by the Zimbra installation packages.

Note: *The directory organization is the same for any server in the Zimbra Collaboration Suite, installing under **/opt/liquid**.*

Table 1 Directory Structure for Zimbra Components

Parent	Directory	Description
/opt/ liquid/		Created by all Zimbra installation packages
	backup/	Backup target contains full and incremental backup data
	bin/	Zimbra application files, including the command-line utilities described in Appendix B: Command-Line Utilities
	cnvfiles/	Files that are being converted by Verity
	conf/	Configuration information
	db/	Data Store
	index/	Index Store
	java/	Contains Java application files
	lib/	Libraries
	libexec/	Internally used executables
	liquidmon/	Contains the control scripts and Perl modules
	log/	Local logs for Zimbra server application
	mysql/	MySQL database files
	redolog/	Contains current transaction logs for the Zimbra server
	openldap/	OpenLDAP server installation, pre-configured to work with Zimbra
	postfix/	Postfix server installation, pre-configured to work with Zimbra

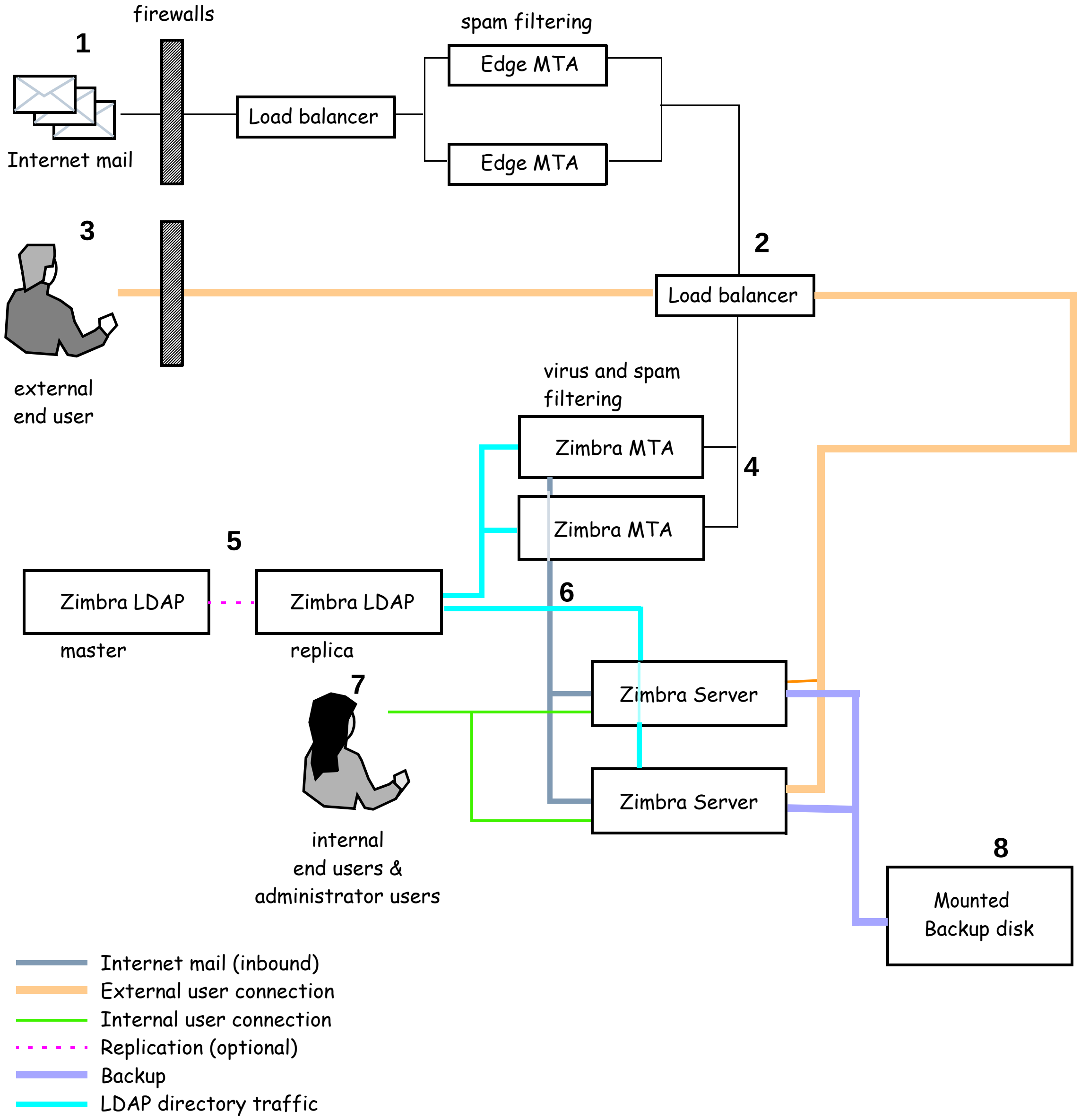
Parent	Directory	Description
	sleepycat/	Berkeley DB
	snmp/	SNMP monitoring files
	ssl/	Certificates
	store/	Message Store
	tomcat/	Tomcat application server instance

Example of a Typical Configuration

The exact configuration for each deployment is highly dependent on variables including the number of mailboxes, mailbox quotas, performance requirements, existing network infrastructure, IT policies, security methodologies, spam filtering requirements, and so forth.

Figure 2 shows a typical configuration with incoming traffic and user connection. Alternate ways of configuring at many points within the network are possible.

Figure 2: Typical Configuration with Incoming Traffic and User Connections



Explanation of Figure 2 follows.

1 Inbound Internet mail goes through a firewall and load balancing to the edge MTA for spam filtering.

2 The filtered mail then goes through a second load balancer.

3 An external end user connecting to the messaging server also goes through a firewall to the second load balancer.

4 The inbound Internet mail goes to any of the Zimbra MTA servers and goes through spam and virus filtering.

5 The designated Zimbra MTA server looks up the addressee’s directory information from the Zimbra LDAP replica server.

6 After obtaining the user’s information from the Zimbra LDAP server, the MTA server sends the mail to the appropriate Zimbra server.

7 Internal end-user connections are made directly to any Zimbra server which then obtains the user’s directory information from Zimbra LDAP and redirects the user as needed.

8 Zimbra servers’ backups are processed to this mounted disk.

Chapter 3 Using the Administration Console

The Zimbra administration console is the browser-based user interface used to centrally manage all Zimbra servers and mailbox accounts.

The administration console is installed when you install the Zimbra Collaboration Suite and the administrator's user name and password are configured during installation as an account on the system, allowing you to log on to the console.

You can log on to the administration console from either the Mozilla Firefox browser, 1.0 or later or the MIcrosoft Internet Explorer browser, 6.0 or later.

Administrator Accounts

Only administrator accounts can log into the administration console, to add accounts and manage server configuration. One administrator account is initially created when the software is installed. Additional administrator accounts can be created.

To give administrator privileges to an account, when you configure an account, check the Administrator Account box on the **Advanced** page, or for existing accounts, check the Administrator Account box on the **Advanced** tab.

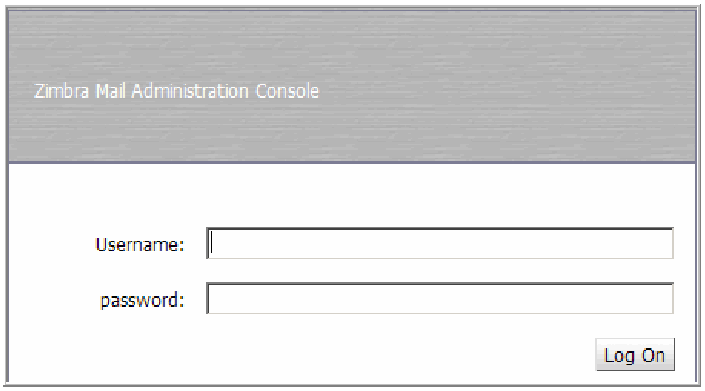
Logging on

To start the console in a typical installation, use the following URL pattern.

https://server.domain.com:7071/

Where **server.domain.com** is the current running Zimbra server name or IP address and **default** HTTP listen port is 7071.

The first time you log on, enter the administrator name and password configured during the Zimbra installation and click **Log On**.

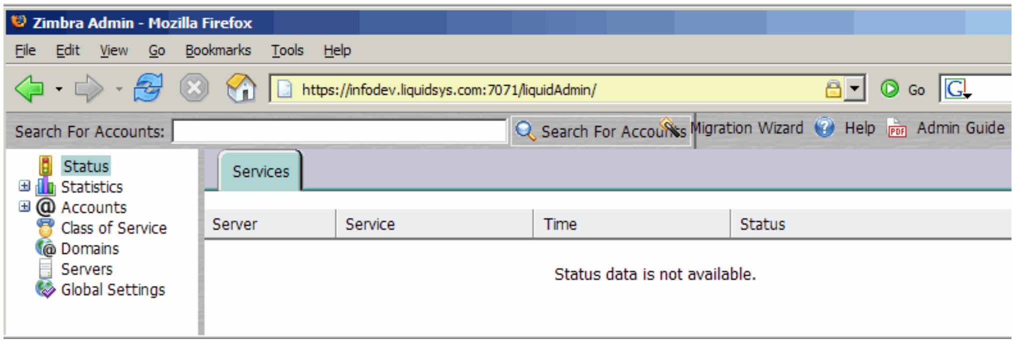


Changing Administrator Passwords

The administrator password is changed from the **Accounts** toolbar. Select the administrator account and click change password.

About the Administration Console

When you open the admin console, the servers' statuses are displayed on the right pane and the navigation bar is on the left. The navigation pane allows you to access the functions exposed through the console.



Status data is not available.

The left navigation pane includes the following folders:

- **Status**. Shows the current status, either **On** or **Off,** for all servers that are running Liquid MTA, Liquid LDAP, Liquid Store, SNMP, and the anti-virus service.
- **Statistics**. Shows both system-wide and server specific data about the inbound message volume, inbound message count, and disk usage for messages processed in the last 24 hours, the last three months, and the last year.

- **Accounts**. Lists all accounts. If you select a specific domain, only accounts on that domain are displayed. In the **Accounts** folder, you create and manage end-user accounts, setting options, class of service, passwords and aliases for an account.
- **Class of Service**. Lists classes of services (COS) that have been created. As a minimum, the default COS is displayed. You can create, edit, or delete COS definitions.
- **Domains**. Lists the domain in the Zimbra environment. You can create and manage domains, configure GAL, and configure the authentication mechanism to use for that domain.
- **Servers**. Lists the servers, the host name and description. You can configure services, MTA, SMTP, IMAP, and POP features for servers.
- **Global Settings**. From the Global Settings folder, you set the global defaults rules for GAL search results, acceptance of incoming attachments, for MTA, POP, IMAP, anti-spam and anti-virus configuration. These default settings are used when personal options and COS settings are not set.

In addition to these functions, the **Search For Accounts:** field allows you to quickly find accounts for editing.

See the Chapter 7, Managing the Zimbra System, for information about how to configure these functions.

Management Tasks from the Administration Console

From the Administration Console, you can do the following:

- Manage end-user accounts
- Monitor server status and performance statistics
- Add or remove domains
- Create Classes of Service (COS), which are used to define settings for user accounts
- Enable or disable optional user-interface features such as conversations and contacts in the email client
- Configure various global settings for security, address book, and MTAs.
- Use the Migration Wizard to migrate Microsoft Exchange server email accounts to the Zimbra server and to import the email and contact information

Management Tasks Not Available from Administration UI

The Zimbra command-line interface (CLI) is another method of configuring and maintaining the Zimbra system. The CLI tool set contains utilities that are not available through the administration console. The CLI options are executed on each server individually.

Use CLI command utilities for the following

- Backup and restore. See Chapter 10, Backup and Restore
- Start and stop services. See CLI Command “lqcontrol (Start/Stop Service)” on page 94
- Create self-signed certificates. See CLI Command, “lqcreatecert (Generate Self-Signed Certificate)Syntax” on page 95
- Manage local server configuration, See CLI command “lqlocalconfig (Local Configuration)” on page 95
- Provision accounts in bulk. See CLI command “lqprov (Provisioning)” on page 84
- Create distribution lists. See CLI command lqprov, “CreateDistributionList” on page 86

See Appendix B: Command-Line Utilities for a description of the command line utilities.

Chapter 4 Zimbra Server

The Zimbra server is a dedicated server that manages all of the mailbox contents, including messages, contacts, calendar, and attachments. Messages are received from the Zimbra MTA server and then passed through any filters that have been created. Messages are then indexed and deposited into the correct mailbox.

In addition to content management, the Zimbra mailbox server has dedicated volumes for backup and log files.

Each Zimbra mailbox server in the system can see only its own storage volumes. Zimbra mailbox servers cannot see, read, or write to another Zimbra server.

In a Zimbra single server environment, all services are on one server, and during installation the computer is configured to partition the disk to accommodate each of the services.

In a Zimbra multi-server environment, the Zimbra LDAP and Zimbra MTA services would be installed on separate servers.

Incoming Mail Routing

The Zimbra MTA, receives mail via SMTP and routes each mail message to the appropriate Zimbra mailbox server using LMTP. As each mail arrives, the Zimbra server schedules a thread to have Lucene index it.

Disk Layout

The mailbox server includes the following volumes:

- **System Metadata.** Any files directly in **opt/liquid**
- **Message Store.** Mail message files are in **opt/liquid/store**
- **Data Store.** The MySQL Database files are in **opt/liquid/db**
- **Index Store.** Index files are in **opt/liquid/index**
- **Backup Area.** Full and incremental backups are in **opt/liquid/backup**
- **Log files.** Each component in the Zimbra installation package has log files.

Note: *The system logs, the redo logs, and the backup disk should be on separate volumes to minimize the possibility of unrecoverable data loss in the event that one of those volumes fails.*

Message Store

The Zimbra Message Store is where all email messages reside, including message body and any file attachments. Messages are stored in MIME format.

The Message Store is located on each Zimbra server under **/opt/liquid/store**. Each mailbox has a dedicated directory named after the internal Zimbra mailbox ID.

Note: *Mailbox IDs are unique per server, not system-wide.*

Single-Copy Message Storage

“Single copy storage” allows messages with multiple recipients to be stored only once on the file system. On UNIX systems, the mailbox directory for each user contains a hard link to the actual file. In multi-server configurations, where recipients may be in different Data Stores, one copy exists per server.

Data Store

The Zimbra Data Store is a MySQL database where internal mailbox IDs are linked with user accounts. The data store contains all the metadata regarding the messages including tags, conversations, and pointers to where the messages are stored in the file system.

Each account (mailbox) resides only on one server. Each Zimbra server has its own standalone data store containing data for the mailboxes on that server.

The Data Store contains:

- Mailbox-account mapping. The primary identifier within Zimbra is the mailbox ID, rather than a user name or account name. The mailbox ID is only unique within a single mailbox server. The Data Store maps the Zimbra mailbox IDs to the users’ OpenLDAP accounts.
- Each user’s set of tag definitions, folders, and contacts, calendar appointments, filter rules.
- Information about each mail message, including whether it is read or unread, and which tags are associated.

Index Store

The index and search technology is provided through Apache Lucene. Each mail is automatically indexed as it enters the system. Each mailbox has an index file associated with it.

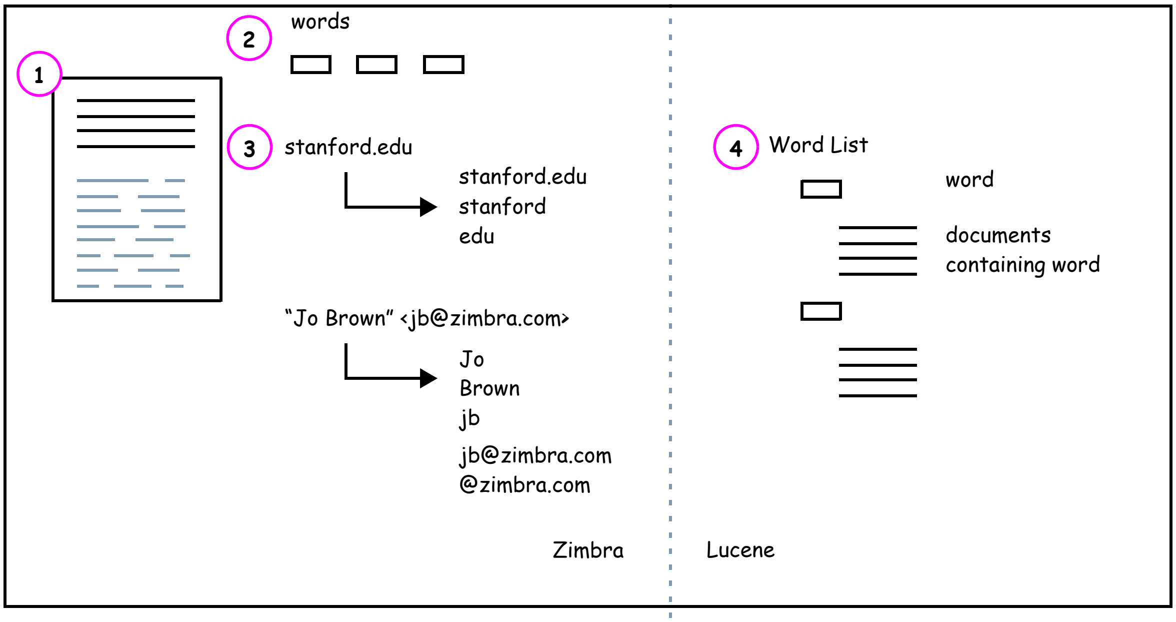
The tokenizing and indexing process is not configurable by administrators or end users.

The process is as follows:

1. The Zimbra MTA routes the incoming email to the Zimbra mailbox server that contains the account's mailbox.
2. The mailbox server parses the message, including the header, the body, and all readable file attachments such as PDF files or Microsoft Word documents, in order to tokenize the words.
3. The mailbox server passes the tokenized information to Lucene to create the index files.

   Note: *Tokenization: The method for indexing is by each word. Certain common patterns, such as phone numbers, email addresses, and domain names are tokenized as shown in Figure 3.*

Figure 3: Message tokenization



Other Dedicated Directories

The Zimbra server also maintains the following functions:

- Backup files
- Redo log files
- Logs files

Backup Files

Zimbra includes a configurable backup manager that resides on every Zimbra server and performs both backup and restore functions. You do not have to stop the Zimbra server in order to run the backup process. The backup manager can be used to restore a single user, rather than having to restore the entire system in the event that one user's mailbox becomes corrupted. See Chapter 10, Backup and Restore.

Redo Log Files

Each Zimbra server generates redo logs that contain every transaction processed by that server. If an unexpected shutdown occurs to the server, the redo logs are used for the following:

- To ensure that no uncommitted transactions remain, the server rereads the redo logs upon startup.
- During restore, to recover data written since the last full backup in the event of a server failure.

When the redo log file size reaches 100MB, the redo log rolls over to an archive directory. At that point, the server starts a new redo log. All uncommitted transactions from the previous redo log are preserved. In the case of a crash, when the server restarts, the current redo log and the archived logs are read to re-apply any uncommitted transactions. When an incremental backup is run, the redo logs are moved from the archive to the backup directory.

Log Files

A Zimbra deployment consists of various third-party components with one or more Zimbra mailbox servers. Each of the components may generate its own logging output.

Selected Zimbra log messages generate SNMP traps, which you can capture using any SNMP monitoring software. See Chapter 9, Monitoring Zimbra Servers

Chapter 5 Zimbra Directory Service

The Zimbra LDAP service is a directory service running a version of the OpenLDAP software that has the Zimbra schema already installed. This chapter describes how the directory service is used for user authentication and account configuration and management.

Note: *Zimbra also supports integration with Microsoft's Active Directory Server. Contact Zimbra support for more detailed information on specific directory implementation scenarios.*

The LDAP server is identified when the Zimbra software is installed. Each server has its own LDAP entry that includes attributes specifying operating parameters. In addition, there is a global configuration object that sets defaults for any server whose entry does not specify every attribute.

A selected subset of these attributes can be modified through the Zimbra administration console; others can be changed through the CLI utility.

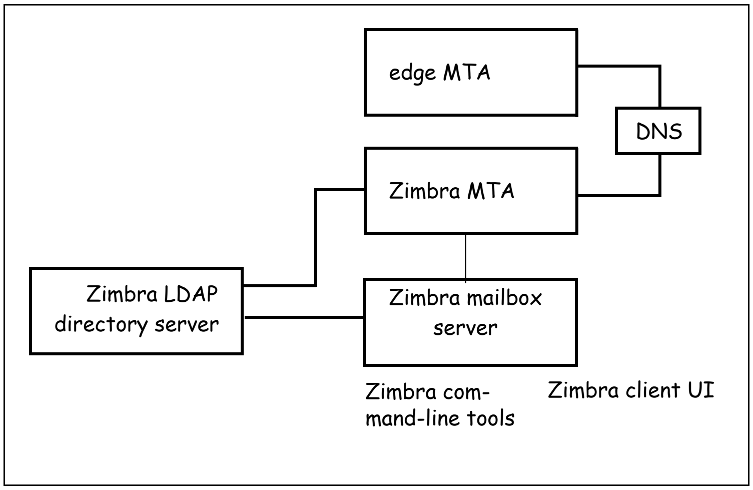
Directory Services Overview

LDAP directory services provide a centralized repository for information about users and devices that are authorized to use your network. The central repository used for Zimbra's LDAP data is the OpenLDAP directory server.

Figure 4 shows traffic between the Zimbra-LDAP directory server and the other servers in the Zimbra system. The Zimbra MTA and the Zimbra mailbox server read from, or write to, the LDAP database on the directory server. The edge MTA does not connect to the LDAP database; instead, it uses the DNS server's MX entry to determine where to direct mail.

The Zimbra clients connect through the Zimbra server, which in turn connects to LDAP.

Figure 4: LDAP Directory Traffic



At the core of every LDAP implementation is a database organized using a *schema*. The schema specifies the types of objects that are stored in the database, and what types of attributes they have.

An LDAP directory entry consists of a collection of attributes and has a globally unique *distinguished name* (DN). The attributes allowed for an entry are determined by the *object classes* associated with that entry. The values of the object class attributes determine the schema rules the entry must follow.

The object classes determine what type of object the entry refers to, and what type of data can be stored for that entry. An entry's object classes that determines what kind of entry it is, is called a structural object class and cannot be changed. Other object classes are called auxiliary and may be added to or deleted from the entry.

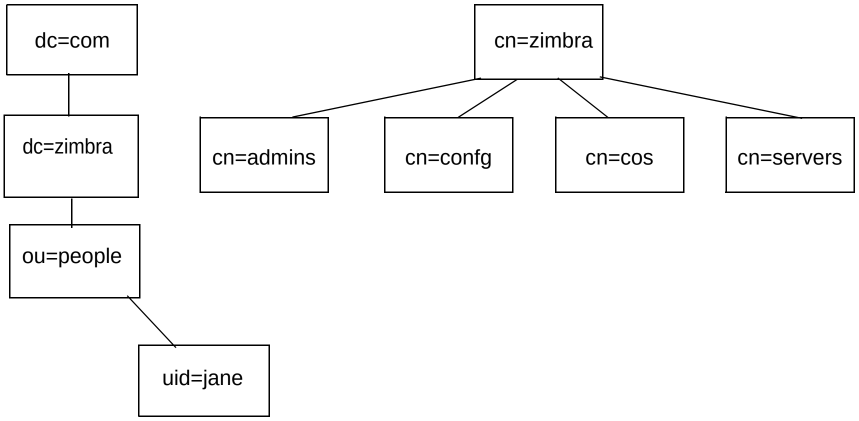
Use of auxiliary object classes in LDAP allows for an object class to be combined with an existing object class. For example, an entry with structural object class **inetOrgPerson,** and auxiliary object class **liquidAccount,** would be an account, either administrator or end-user. An entry with the object class **liquidServer** would be a server in the Zimbra system that has one or more Zimbra software packages installed.

LDAP Hierarchy

LDAP directories are arranged in an hierarchal tree-like structure. In the Zimbra system, the structure is arranged based on Internet domain names. LDAP entries typically include items such as user accounts, organizations, or servers.

Figure 5 shows the Zimbra LDAP hierarchy. Each type of entry (object) has certain associated object classes.

Figure 5: Zimbra LDAP Hierarchy



For a complete listing of the Zimbra auxiliary object classes, see the Zimbra LDAP Schema.

Zimbra Schema

Every LDAP implementation has a schema that defines its domain structure, account attributes, and other data structures in use by the organization. Zimbra includes a custom LDAP schema that extends the generic schema included with OpenLDAP software and is designed to potentially co-exist with existing directory installations. The Zimbra server, the Zimbra administration console, the command-line account provisioning, and the management utilities require the Zimbra schema.

All attributes and object classes specifically created for Zimbra are prefaced by “liquid”, as in **liquidMailRecipient** object class or the **liquidAttachmentsBlocked** attribute.

The Zimbra schema assumes a baseline schema. In the OpenLDAP installer package included with the Zimbra system, the following schema files are included in the OpenLDAP implementation:

- **core.schema**
- **cosine.schema**
- **inetorgperson.schema**
- **liquid.schema**

Note: *You cannot modify the Zimbra schema.*

Account Authentication

This section describes the account authentication mechanisms and formatting directives supported:

- **Internal**
- **External LDAP**
- **External Active Directory**

The **Internal** authentication method assumes the Zimbra schema, running on the OpenLDAP directory server.

The **External LDAP** and **External Active Directory** authentication methods attempt to bind to the specified LDAP server, using the supplied user name and password. These method can be used if the email environment uses Microsoft Active Directory directory services for authorization and the Zimbra-LDAP directory services for all other Zimbra-related transactions. This requires that users exist in both OpenLDAP and in the Active Directory servers.

The method type is set on a per-domain basis, using the **liquidAuthMech** attribute, with other information also coming from the domain. If this attribute is not set, the default is to use the internal method as the authentication.

The Internal Authentication Mechanism

For accounts stored in the OpenLDAP server, the **userPassword** attribute stores a salted-SHA1 (SSHA) digest of the user’s password. This information is not used to connect to the directory server; it is only used to compare with the information on the OpenLDAP server, using a pool of re-usable administrator LDAP connections.

External LDAP and External Active Directory Authentication Mechanism

Unlike the internal authentication mechanism, the external authentication mechanism attempts to bind to the directory server using the supplied user name and password. If this bind succeeds, the connection is closed and the password is considered valid.

Two additional domain attributes are required for the external mechanism: **liquidAuthLdapURL** and **liquidAuthLdapBindDn**.

liquidAuthLdapURL Attribute and SSL

The **liquidAuthLdapURL** attribute contains the URL of the Active Directory server to bind to. This should be in the form:

ldap://*ldapserver***:***port*/

where *ldapserver* is the IP address or host name of the Active Directory server, and *port* is the port number. You can also use the fully qualified host name instead of the port number.

Examples include:

ldap://server1:389
ldap://exch1.acme.com

For SSL connection, use **ldaps:** instead of **ldap:**. If the SSL version is used, the SSL certificate used by the server must be configured as a trusted certificate.

liquidAuthLdapBindDn Attribute

The **liquidAuthLdapBindDn** attribute is a format string used to determine which user name to use when binding to the Active Directory server.

During the authentication process, the user name starts out in the format:

user@domain.com

The user name may need to be transformed into a valid LDAP bind dn (distinguished name). In the case of Active Directory, that bind dn might be in a different domain.

Zimbra Objects

Zimbra uses auxiliary object classes to add Zimbra-specific attributes to existing objects such as an account. The LDAP objects used in Zimbra include the following:

- Accounts
- Class of Service (COS)
- Domains
- Distribution Lists
- Recipients
- Servers
- Global Configurations
- Aliases

Accounts Object

An account represents an account on the Zimbra mailbox server that can be logged into. Account entrees are either administrators or end user accounts that can be logged into. The object class name is **liquidAccount**. This object class extends the **liquidMailRecipient** object class.

The object class, **liquidMailRecipient**, is a directory entry that represents an entity that can receives mail. This is a visible external mail address that is expanded through aliases or forwarding into one or more internal/external addresses.

All accounts have the following properties:

- A name in the format of user@some.domain.

- A unique ID that never changes and is never reused.
- A set of attributes, some of which are user-modifiable (options) and others that are only configurable by the system administrator.

All user accounts are associated with a domain, so a domain must be created before creating any accounts.

For more about account provisioning, see the Managing User Accounts section, Chapter 7.

Class of Service (COS) Object

Class of Service is a Zimbra-specific object that defines the default attributes an email account has and what features are added or denied. The COS controls mailbox quotas, message lifetime, password restrictions, attachment blocking and server pools for creation of new accounts. The object class name is **liquidCOS.**

Each account is assigned a class of service. COS is used to group accounts and define the feature levels for those accounts. For example, executives can be assigned to a COS that allows the Calendar application. By grouping accounts into specific type of COS, account features can be updated in block.

If the COS is not explicitly set, or if the COS assigned to the user no longer exists, values come from a pre-defined COS called "default".

A COS is not restricted to a particular domain or set of domains.

Domains Object

A Domains object represents an email domain such as *ace.**com*** or *zink.**org**.* A domain must exist before email addressed to users in that domain can be delivered. The object class name is **liquidDomain**.

Distribution Lists Object

Distribution Lists, also known as mailing lists, are used to send mail to all members of a list by sending a single email to the list address. The object class name is **liquidDistributionList**.

Recipient Object

Recipient object represents an entity that can receive mail. An external email address exists, and the recipient can be expanded through aliases or forwarding into one or more internal/external addresses. The object class name is **liquidMailRecipient**. This object class name is only used in conjunction with **liquidAccount** and **liquidDistributionlist** classes.

Servers Object

The servers object represents a particular server in the Zimbra system that has one or more of the Zimbra software packages installed. During the installation, the software is automatically registered on the OpenLDAP server. The object class name is **liquidServer**. Attributes describe server configuration information, such as which services are running on the server.

The server name is used by the Zimbra system to make a request for the server object in the directory. The server requested gets its configuration information and picks up any changes that might have been made by the administrator through the Zimbra Administrator Console.

Global Configuration Object

The Global Configuration object specifies default values for the following objects: server, account, COS, and domain. If the attributes are not set for other objects, the values are inherited from the global settings. The object class name is **liquidGlobalConfig**.

Global configuration values are required and are set during installation as part of the Zimbra core package. These become the default values for the system.

Alias Object

Alias object is a placeholders in the directory to reserve a name. The object class name is **liquidAlias**. The attribute points to another entry.

Company Directory/GAL

A company directory is a company-wide listing of users, usually within the organization itself, that is available to all users of the email system. Sometimes called “white pages” or global address list (GAL), Zimbra uses the company directory to look up user addresses from within the company.

For each domain used in Zimbra, you can choose from the following GAL search options:

- Use an external LDAP server for the GAL
- Use the Zimbra implementation in OpenLDAP
- Include both external LDAP server and OpenLDAP in GAL searches

GAL Searches in Zimbra Client

The Zimbra client can search the GAL. The GAL search returns a list of directory entries that match the user’s search.

When the user supplies a name to search for, that name is turned into an LDAP search filter similar to the following example:

```
(|(cn = %s*)(sn=%s*)(gn=%s*)(mail=%s*)
   (liquidMailDeliveryAddress = %s*)
```

```
        (liquidMailAlias=%s*)
        (liquidMailAddress = %s*)
```

The string “%s” is replaced with the name the user is searching for.

GAL Attributes in Zimbra

Two possible sources for GAL information are the Zimbra server and the Active Directory server. The relevant LDAP/Active Directory fields are referenced in the Zimbra schema under the same names as listed in the Active Directory schema.

Table 2 maps generic GAL search attributes to their Zimbra contact fields.

Table 2 Attributes Mapped to Zimbra contact

Standard LDAP Attribute	Zimbra Contact Field
co	workCountry
company	Company
givenName/gn	firstName
sn	lastName
cn	fullName
initials	initials
l	workCity
physicalDeliveryOfficeName	office
ou	department
postalAddress	workStreet
postalCode	workPostalCode
telephoneNumber	workPhone
st	workState
title	jobTitle
mail	email
objectClass	Not currently mapped

Zimbra GAL Search Parameters

Like authentication, GAL is configured on a per-domain basis. From the administration console, you can run the GAL Configuration Wizard to configure the domain’s attributes.

Modifying Attributes

The OpenLDAP directory should not be modified directly. Any additions, changes and deletions are made through the Zimbra administration console or from the CLI utility for provisioning, **lqprov**.

End users modify attributes for their entry (accounts) in the OpenLDAP directory when they change their options from the Zimbra Web Client.

Administrators can also modify LDAP attributes using the command-line tools described in Appendix B: Command-Line Utilities.

Important: *Do not use any LDAP browsers to change the Zimbra LDAP content.*

Chapter 6 Zimbra MTA

The Zimbra MTA (Mail Transfer Agent) receives mail via SMTP and routes each message, using Local Mail Transfer Protocol (LMTP), to the appropriate Zimbra server.

The Zimbra MTA server includes the following programs:

- Postfix MTA, for mail routing, mail relay, and message blocking.
- Clam AntiVirus, an antivirus engine integrated with the MTA, designed for email scanning.
- SpamAssassin, a mail filter that attempts to identify unsolicited commercial email (spam), using a variety of mechanisms.
- Amavisd-New, a Postfix content filter used as an interface between Postfix and ClamAV / SpamAssassin.

In the Zimbra Collaboration Suite configuration, mail transfer and delivery are distinct functions. Postfix primarily acts as a Mail Transfer Agent (MTA) and the Zimbra mail server acts as a Mail Delivery agent (MDA).

A configuration script automatically polls the LDAP directory every two minutes for modifications, and updates the Postfix configuration files with the changes.

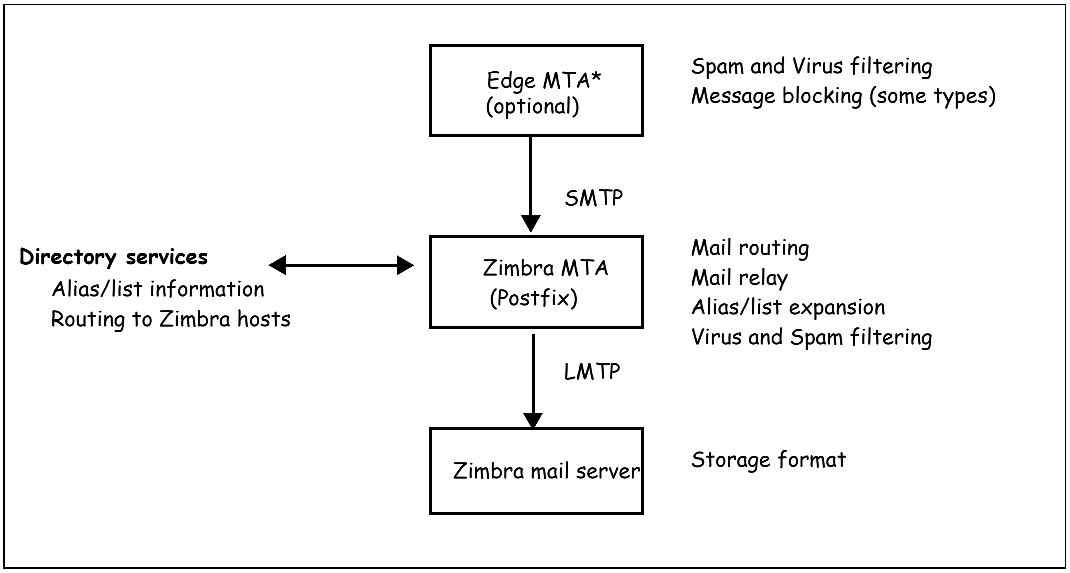
Zimbra MTA Deployment

The Zimbra Collaboration Suite includes a precompiled version of Postfix. This version does not have any changes to the source code, but it does include configuration file modifications, additional scripts, and other tools.

Postfix performs the Zimbra mail transfer and relay. It receives inbound messages via SMTP, and hands off the mail messages to the Zimbra server via LMTP, as shown in Figure 6. The Zimbra MTA can also perform anti-virus and anti-spam filtering.

Postfix also plays a role in transfer of outbound messages. Messages composed from the Zimbra web client are sent by the Zimbra server through Postfix, including messages sent to other users on the same Zimbra server.

Figure 6: Postfix in a Zimbra Environment



***Edge MTA** The term "edge MTA" is a generic term referring to any sort of edge security solution for mail. You may already deploy such solutions for functions such as filtering. The edge MTA is optional. Some filtering may be duplicated between an edge MTA and the Zimbra MTA.

Postfix Configuration Files

Zimbra modified the following Postfix files specifically to work with the Zimbra Collaboration Suite:

- **main.cf** - Modified to include the LDAP tables. The configuration script in the Zimbra MTA pulls data from the Zimbra LDAP and modifies the Postfix configuration files.
- **master.cf** - Modified to use Amavisd-New.

Important: *Do not modify the Postfix configuration files directly! Some of the Postfix files are rewritten when changes are made in the administration console. Any changes you make will be overridden.*

MTA Functionality

Zimbra MTA Postfix functionality includes:

- SMTP authentication
- Attachment blocking
- Relay host configuration
- Postfix-LDAP integration

- Integration with Amavisd-New, ClamAV, and Spam Assassin

SMTP Authentication

SMTP authentication allows authorized mail clients from external networks to relay messages through the Zimbra MTA. The user ID and password is sent to the MTA when the SMTP client sends mail so the MTA can verify if the user is allowed to relay mail.

Note: *User authentication is provided through the Zimbra LDAP directory server, or if implemented, through the Microsoft's Active Directory Sever.*

SMTP Restrictions

In the administration console, you can enable restrictions that cause messages to not be accepted by Postfix when non-standard or other disapproved behavior is exhibited by an incoming SMTP client. These restrictions provide some protection against ill-behaved spam senders. By default, SMTP protocol violators (e.g., clients that do not greet with a fully qualified domain name) are restricted. DNS based restrictions are also available.

Important: *Understand the implications of these restrictions before you implement them. You may want to receive mail from people outside of your mail system, but those mail systems may be poorly implemented. You may have to compromise on these checks to accommodate them.*

Relay Host Settings

Postfix can be instructed to send all non-local mail to a different SMTP server. Such a destination SMTP server is commonly referred to as a “relay” or “smart” host. You can set this relay host on the administration console. Common use case for a relay host is when an ISP requires that all your email be relayed through designated host, or if you have some filtering SMTP proxy server,

In the administration console, the relay host setting must not be confused with web mail MTA setting. Relay host is the MTA to which Postfix relays non-local email. Webmail MTA is used by the Zimbra server for composed messages and must be the location of the Postfix server in the Zimbra MTA package.

Important: *Use caution when setting relay host to prevent mail loops*

MTA-LDAP Integration

The Zimbra LDAP directory service is used to look up email delivery addresses. The version of Postfix included with Zimbra is configured during the installation of the Suite to use the Zimbra LDAP directory.

Account Quota and the MTA

Account quota is the storage limit allowed for an account. Account quotas can be set by COS or per account. The MTA attempts to deliver a message, and if a Zimbra user's mailbox exceeds the set quota, the Zimbra mailbox server rejects the message as mailbox is full and the sender gets a bounce message.

MTA and Amavisd-New Integration

The Amavisd-New utility is the interface between the Zimbra MTA and Clam AV and SpamAssassin scanners.

Anti-Virus Protection

Clam AntiVirus software is bundled with the Zimbra Collaboration Suite as the virus protection engine. Questions during the installation process asks if you want to enable anti-virus protection. You can also enable or disable virus checking from Global Settings on the administration console.

The Zimbra MTA checks every two hours for any new anti-virus updates from ClamAV. You can modify the frequency. The global settings for the anti-virus protection is configured with these options enabled:

- Block encrypted archives, such as password protected zipped files.
- Send notification to administrator to alert administrators that a virus has been found.
- Send notification to recipient to alert that a mail message to them had a virus and was not delivered.

Anti-Spam Protection

SpamAssassin software is bundled with the Suite as the spam filtering engine. Question during the installation process asks if you want to enable anti-spam protection. The global settings for the anti-spam protection is configured with these options enabled.

- Kill percent at 75%. Mail that is scored at 75% is considered spam and is not delivered.
- Tag percent at 33%. Mail that is scored at 33% is considered spam and is delivered to the Junk folder.
- Subject prefix field is blank. The prefix entered in this field is added to the subject line for messages tagged as spam.

When a message is tagged as spam, the message is delivered to the recipient's Junk folder. Users can view the number of unread messages that are in their Junk folder and can open the junk folder to review the messages marked as spam.

Users can use the Junk button on their toolbar to report a message as spam.

Chapter 7 Managing the Zimbra System

This chapter describes the following functions used to manage the Zimbra Collaboration Suite. Features can be managed from either the administration console or from the CLI utility.

- Global configuration
- Domains
- Servers
- User Accounts
- Backing up the system

Help is available from the administration console about how to perform tasks from the administration console. If the task is only available from the CLI, see Appendix B: Command-Line Utilities for a description of how to use the CLI utility.

Managing Global Configurations

Global Settings control default global rules that apply to accounts in the Zimbra servers. These are set during installation. The settings can be modified from the administration console.

Global settings include the following tabs:

- General
- Attachments
- MTA
- Pop
- IMAP
- Anti-Spam
- Anti-Virus

Note: *Configurations set in Global Settings define inherited default values for the following objects: server, account, COS, and domain if these attributes are set in the COS or Account set up, they override the global settings.*

General Tab

In the General tab configure the **Most results returned by GAL search** field, which sets a global ceiling for the number of GAL results returned from a user search. The default is 100 returns per search.

Attachments Tab

Zimbra supports the following types of attachment blocking:

- **Global settings**, to disable attachment viewing and to reject messages that include attachments with specific extensions
- **Class of Service**, to disable attachment viewing for members of that COS
- **Accounts**, to disable attachment viewing for individual accounts

In Global Settings, the **Attachments** tab can be configured with global rules to reject mail with files attached and to disable viewing files attached to mail messages in users' mailboxes. When attachment settings are configured in Global Settings, the global rule takes precedence over COS and Account settings.

If **Disable attachment viewing from web mail UI** is enabled, users cannot view any attachments in their mailbox. You can set this global setting to prevent a virus outbreak if you think that mail has already been sent.

Reject messages with attachment extension lets you select which file types are unauthorized for all accounts. The most common extensions are listed. You can also add different extension types to the list. Messages with those type of files attached are rejected and the sender gets a bounce notice. The recipient does not get the mail message and is not notified.

MTA Tab

From the MTA tab, you can enable or disable authentication and you can configure a relay hostname, the maximum message size, whether to enable DNS lookup, protocol checks, and DNS checks. For a description of Zimbra MTA, see Chapter 6, Zimbra MTA

- Authentication should be enabled, to support mobile SMTP authentication users so that their email client can talk to the Zimbra MTA
- **TLS authentication only** should be checked
- The **Relay hostname** is the Zimbra MTA to Gateway host
- The Protocol fields are checked to reject unsolicited commercial email (UCE), for SPAM control.
- The DNS fields are checked to reject mail, if the client's IP address is unknown, the hostname in the greeting is unknown and/or if the sender's domain is unknown.

POP Tab

POP3 (Post Office Protocol) can be enabled to allow users with a POP client to access their mail stored on the Zimbra server and download new mail to their computer. The POP configuration determines if messages are deleted from the Zimbra server when downloaded.

IMAP Tab

The Internet Message Access Protocol (IMAP) can be enabled to allow users with an IMAP client to access their mail stored on the Zimbra mailbox server from more than one computer. The messages are stored on the mailbox server.

Anti-Spam Tab

Anti-spam protection can be enabled for each server when the Zimbra software is installed. The following options are configured:

- Kill percent at 75%. Mail that is scored at 75% is considered spam and is not delivered.
- Tag percent at 33%. Mail that is scored at 33% is considered spam and is delivered to the Junk folder.
- Subject prefix field is blank. The prefix entered in this field is added to the subject line for messages tagged as spam.

When a message is tagged as spam, the message is delivered to the recipient’s Junk folder. Users can view the number of unread messages that are in their Junk folder and can open the junk folder to review the messages marked as spam.

Turning On or Off RBLs

RBL can be turned on or off in SpamAssassin from the Zimbra CLI.

The three RBL’s that are enabled during installation are the following:

- reject_invalid_hostname
- reject_non_fqdn_hostname
- reject_non_fqdn_sender

You can set the following, in addition to the three above:

- reject_rbl_client dnsbl.njabl.org
- reject_rbl_client opm.blitzed.org
- reject_rbl_client relays.ordb.org
- reject_rbl_client cbl.abuseat.org
- reject_rbl_client bl.spamcop.net
- reject_rbl_client dnsbl.sorbs.net

- reject_rbl_client sbl.spamhaus.org
- reject_rbl_client relays.mail-abuse.org

To turn RBL on

1. Log on to the server and go to the Zimbra directory (su - liquid)
2. Enter **lqprov gacf | grep liquidMtaRestriction,** to see what RBLs are set.
3. To add any new RBL types, you must list the existing RBLs and the new RBLs all in one command as:

   lqprov mcf liquidMtaRestriction [RBL type]

   To add all the possible restrictions, the command would be

   lqprov mcf liquidMtaRestriction reject_invalid_hostname liquidMtaRestriction reject_non-fqdn_hostname liquidMtaRestriction reject_non_fqdn_sender liquidMtaRestriction “reject_rbl_client dnsbl.njabl.org” liquidMtaRestriction “reject_rbl_client opm.blitzed.org” liquidMtaRestriction “reject_rbl_client relays.ordb.org” liquidMtaRestriction “reject_rbl_client cbl.abuseat.org” liquidMtaRestriction “reject_rbl_client bl.spamcop.net” liquidMtaRestriction “reject_rbl_client dnsbl.sorbs.net”

Note: *Quotes must be added to RBL types that are two words.*

Anti-Virus

Anti-virus protection can be enabled for each server when the Zimbra software is installed.

The Zimbra MTA checks every two hours for any new anti-virus updates from ClamAV. You can modify the frequency. The global settings for the anti-virus protection is configured with these options enabled:

- Block encrypted archives, such as password protected zipped files.
- Send notification to administrator to alert administrators that a virus has been found.
- Send notification to recipient to alert that a mail message to them had a virus and was not delivered.

Important: *When the Zimbra Collaboration Suite was installed, if you defined a anti-virus notification address at a zimbra domain name, don’t forget to provision the address as an account on the Zimbra server.*

Managing Domains

One domain is identified during the installation process and additional domains can be easily added to the Zimbra system from the administration console. For domains, you configure the Global Address List mode and the authentication mode.

The administration console can also be used to edit domain information or to remove a domain.

Global Address List (GAL) Mode

The Global Address List (GAL) is your company directory. The GAL mode to use for lookup can be set as **Internal** to use the Zimbra LDAP, as **External** to use an external LDAP server, or as **Both**.

A GAL configuration wizard steps you through configuring the GAL mode and to set the maximum number of results returned for a search in GAL.

- The **Most results returned by GAL search** can be configured for performance reasons.
- Select which GAL mode to use for lookup, **Internal, External**, or **Both**. If External or Both are selected, subsequent settings are required.

Authentication Modes

Authentication is the process of granting login access to legitimate users based on user name and password information provided at login time.

Authentication mechanism options are **Internal**, **External**, or **External Active Directory.** See “Account Authentication” on page 31.

An Authenticating configuration wizard steps you through configuring the authenticating mode.

Managing Servers

During the installation, the software is automatically registered on the Zimbra LDAP server. You can view the current status of all the servers that are configured with Zimbra software, and you can edit or delete existing server records. You cannot add servers. The Zimbra Installation program must be used to add new servers because the installer packages are designed to register the new host at the time of installation.

Servers inherit global settings if those values are not set in the server configuration. Settings that can be inherited from the Global configuration include MTA, SMTP, IMAP, POP, anti-virus, and anti-spam configurations.

Reviewing Server Status

View the**Status** page to see if all servers and services are running, The servers listed include the MTA, LDAP and mailbox server. The possible services are SNMP, MTA, LDAP, mailbox, antivirus. The status is either On of Off.

To start a server if it is not running, use the **lqcontrol** CLI command.

Viewing Performance Statistics

View the **Statistics** page to see Inbound message volume and message count and to monitor disk usage. The information is available for the last 24 hours, last 3 months, and last 12 months.

Managing User Accounts

Managing accounts in the Zimbra system allows you to create accounts and change features easily from the administration console or by using the **lqprov** command-line tool described in Appendix B.

From the administration console the following account functions help you manage user accounts:

- Quickly create new accounts with the **New Account Wizard**
- **F**ind a specific account using the Search For Accounts feature
- Edit account information
- Change password for a selected account
- View an account’s mailbox
- Change account status
- Restore a mailbox
- Delete an account

See the Chapter 8, Managing End-User Mailbox Features, for descriptions of the mailbox features that can be configured.

Search for Accounts

Search is used to quickly locate individual accounts in the LDAP server. Search by display name, first name, last name, the first part of the email address, alias, delivery address. If you don’t know the complete name, you can enter a partial name. Partial names can result in a list of accounts that have the partial name string anywhere in the information.

You can also use the Zimbra mailbox ID number to search for an account. To return a search from a mailbox ID, the complete ID string must be entered in the search. Only that account will be returned.

Adding user accounts

If you are using the administration console, the New Account Wizard steps you through the account information to be completed. Before you add an user account, you should determine what features and access privileges should be assigned. You configure the following type of information:

- General information, including account name, class of service to be assigned, password
- Contact information, including phone number, company name and address

- Alias names to be used.
- Forwarding directions
- Features and preferences available for this specific account. Changes made at the account level override the rules in the COS assigned to the account.

Creating an account sets up the appropriate entries on the Zimbra LDAP directory server. When the end-user logs in for the first time or when an email is delivered to the user's account, the mailbox is created on the mailbox server.

Batch Provisioning from the CLI Utility

For provisioning many accounts at once, you create a formatted text file with the user names. This file runs through a script, using the CLI command, **lqprov.** The **lqprov** utility provisions one account at a time.

Create a text file with the list of the accounts you want to add. Each account should be typed in the format of ca (Create Account), email address, empty password. For example, **ca name@company.com ''**

The empty quote is required, as it indicates that there is no local password.

When the text file includes all the names to provision, log on to the Zimbra server and type the CLI command

lqprov provisionaccounts.txt

Each of the names listed in the text file will be provisioned.

See Appendix Appendix B, CLI Commands, for additional syntax definitions.

Class of Service

Class of Service (COS) is a Zimbra-specific object that determines what default attributes a Zimbra Web Client email account has and what features are added or denied. The COS controls mailbox quotas, message lifetime, password restrictions, attachment blocking, and server pools for creation of new accounts.

A default COS is automatically created during the installation of Zimbra software. You can modify the default COS to set the attributes to your email restrictions and you can create new COSs to assign to accounts.

Each account is assigned one class of service. When an account is created, if the COS is not explicitly set, the default COS is assigned. Also, if the COS assigned to the user no longer exists, the account is automatically assigned the default COS.

Note: *COS settings assigned to an account are not enforced for IMAP clients.*

A COS is global and is not restricted to a particular domain or set of domains.

Assigning the COS to an account is a way to quickly configure account features and restrictions. Some of the COS settings can be overridden either by global settings or by user settings. For example:

- Whether outgoing messages are saved to **Sent** can be changed in the end user **Options.**
- Attachment blocking set as a global setting can override the COS setting.

See the Administration Console Help for a complete description of the fields in a class of service object.

Account Distribution by COS

In an environment with multiple mailbox servers, the class of service is used to assign the next account to a mailbox server. The COS server pool tab lists the mailbox servers in your Zimbra environment. When you configure the COS you select which servers to add to the server pool. Within each pool of servers, a random algorithm assigns new mailboxes to any available server.

Changing Password

Password restrictions can be set either at the COS level or at the account level. You can configure the following password rules:

- Password length. The default is minimum 6, maximum 64. The password is case sensitive.
- When passwords expire. The Zimbra default is to never expire the password.
- How frequently a password can be reused. The default password history allows the password to be reused.
- Password locked. Password cannot be changed.

View an Account's Mailbox

View Mail, within Accounts, lets you logon without the account's password and view the selected account's mailbox content, including all folders, calendar entries, and tags. This feature can be used to assist users who are having trouble with their mail account as you and the account user can be logged on to the account.

Any View Mail action to access an account is logged to the *audit.log* file.

Distribution Lists

A distribution list, also known as a mailing list, is a group of email addresses contained in a list with a common email address. When you send to a distribution list, you are sending to everyone whose address is included in the list. Examples of the type of distribution lists to create include department lists, common interest groups lists, and location lists.

Distribution lists can be added, changed and deleted with the CLI utility, **lqprov**. This tools creates the appropriate directory entries on the Zimbra LDAP server. For the Zimbra system, the Zimbra MTA is configured to look up and expand alias lists using LDAP rather than its own built-in mail list functionality.

To create a distribution list and add members

1. To create the list, from the /opt/liquid/bin directory, type **lqprov cdl [***list-name@domain].*
   where *list-name@domain* is the distribution list name.

2. Add an email address to the list, type **lqprov adlm** [*list-name@domain member@domain*].

   For example, marketing@jane@abc.com

3. Continue to type the **lqprov adlm** command to add more names to the list.

To remove a member from the list

From the /opt/liquid/bin directory, type **lqprov rdlm [***list-name@domain member@domain].*

To see the list of all distribution lists in the Zimbra system

From the /opt/liquid/bin directory, type **lqprov gadl**

To see a specific distribution list in the Zimbra system

From the /opt/liquid/bin directory, type **lqprov gdl [***list-name@domain|id***]**

To delete a distribution list in the Zimbra system

From the /opt/liquid/bin directory, type **lqprov ddl [***list name@domain|id***]**

Changing an Account's Status

Account status determines whether an user can log in and receive mail. The account status is displayed when account names are listed on the Accounts page.

The following account statuses can be set:

- **Active**. Active is the normal status for a mailbox account. Mail is delivered and users can log into the client interface.
- **Maintenance**. When a mailbox status is set to maintenance, login is disabled, and mail addressed to the account is queued at the MTA. An account can be set to maintenance mode for backing up, importing or restoring the mailbox.
- **Locked**. When a mailbox status is locked, the user cannot log in, but mail is still delivered to the account. The locked status can be set, if you suspect that a mail account has been hacked or is being used in an unauthorized manner.
- **Closed**. When a mailbox status is closed, the login is disabled. Messages are bounced. This status is used to soft-delete an account before deleting it from the server.

Enforcing Mailbox and Contact Quotas

You can specify mailbox quotas and number of contacts allowed for each user, through the Zimbra administration console. These limits can be set in the following places:

- Global defaults
- Class of Service
- Per-account overrides

Moving a Mailbox

You can move a mailbox from one server to another without taking down the servers. A migration tool is provided through a command-line interface as described inAppendix Appendix B.

The migration tool does the following:

- Puts the mailbox into maintenance mode. In this mode, incoming and outgoing messages are queued but not delivered or sent, and the user will be temporarily unable to access the mailbox
- Packs up the mailbox’s Message Store directory and Index directory on the source server
- Marks all rows associated with the mailbox in the Data Store on the source server
- Creates the new entries and directories on the target server
- Updates the routing information for mail delivery
- Puts the mailbox back into the active mode

Chapter 8 Managing End-User Mailbox Features

When an account is provisioned, you create the email mailbox, assign the email address and configure how users access and use their mailboxes. This chapter describes the features, advanced controls, and user preferences that can be configured for an account either by assigning a COS or by specifying the feature when you create the account.

When accounts are created from the administration console, the New Account Wizard enables most of the features applicable to that account. The account creation utility creates the appropriate entries on the Liquid LDAP directory server. The mailbox is created on the Zimbra server upon the user's first log in to the system.

User Mailbox Features

The COS assigned to an account sets the default feature for the account. These defaults can be changed for individual accounts. The following table lists the features that can be configured either by COS or by Account.

Table 3 Configurable Mailbox Features

Feature Name	COS	Account	Description
Contacts	X	X	Lets users create their own personal address book. The maximum number of contacts an account can have can be set in the advanced options.
Calendar	X	X	A calendar and scheduling tool to let users maintain their calendar and schedule meetings.
Tagging	X	X	Tags allows users to create labels and assign them to messages, to organize their mailbox.

Table 3 Configurable Mailbox Features

Feature Name	COS	Account	Description
Advanced search	X	X	Advanced search allows users to build a search request that can include looking for email by date, domain, flag, object, size, attachment, and location.
Saved searches	X	X	Saved searches allows users to save a search criteria that they “build”.
Conversations	X	X	Messages can be displayed grouped as conversations or as a message list. Conversations group emails by subject. When email subject lines match, the emails are grouped for easy viewing. If this feature is turned on, Conversations is the default.
Change password	X	X	Change password allows users to change their password at any time.
Initial search preference	X	X	Users can select which of their folders to display when they log on. Also, this is the folder that will be searched before other folders.
User-defined mail filters	X	X	Allows users to create rules for managing their email. Rules can include routing mail to different folders.
GAL access	X	X	GAL access allows users to access the company LDAP directory.
HTML Compose	X	X	Allows the user to compose rich emails that can contain different fonts, colors, and style.
IMAP Access	X	X	Enables users to use third party mail applications, such as Thunderbird or Outlook, to access their mailbox using the IMAP protocol.
POP3 Access	X	X	Enables users to use third party mail applications, such as Thunderbird or Outlook, to access their mailbox using the POP protocol.

Advanced Options

Advanced options that can be set at both the COS and Account level are described in the table that follows.

Table 4 Configurable Advanced Options

Advanced Options	COS	Account	Description
Account quota	X	X	Mailbox size limit in MB. The default does not set a mailbox quota.
Address book size limit	X	X	Maximum number of contacts a user can have in their personal Contacts list.
Minimum/Maximum password length	X	X	Specifies the required length of a password.
Disable attachment viewing from web mail UI	X	X	This can be used to block email messages with attachments from being read on the web client UI.
Minimum / Maximum password age	X	X	Number of days that a password must remain unchanged before users can change their password, or the number of days that can elapse before users are forced to change their password.
Enforce password history	X	X	Number of times users can change their password before they can re-use an old password.
Password locked	X	X	Users cannot change their password.
Email message lifetime	X	X	Number of days an email can remain in any folder before it is automatically purged.
Trashed message lifetime	X	X	Number of days an email remains in users' Trash folder before the message is automatically purged.
Spam message lifetime	X	X	Number of days an email can remain in users' Junk folders, before the message is automatically purged.

Table 4 Configurable Advanced Options

Advanced Options	COS	Account	Description
Session token lifetime	X	X	Session token lifetime sets how long a user session remains active, if no activity occurs. Activity includes any clickable mouse action, such as viewing contents of a folder or clicking a button. The default is 2 days.
Must change password		X	When users log on for the first time, they are required to change their password.
Administrator Account		X	This enables the account to be an administrator account and allows the user to log on to the administration console.

Preferences

How user mailboxes display and behave when a message is composed are preferences that can be configured in the COS and in Account configuration. If the preference is set in an account, users can change them from their mailbox Options application.

The preferences are listed in the following table.

Table 5 Configurable Preferences

Preferences	COS	Account
Save to Sent	X	X
View mail as HTML	X	X
Always compose in new window	X	X
Reply/Forward using format of the original messages	X	X
Always compose mail using either text or HTML Default is text.	X	X
Signature style, use a separator between message or not. Default is to not separate the signature.	X	
Enable auto adding of contacts	X	X
Contacts per page. Contacts can be displayed in list or detailed view.	X	X
Number of items to display per page	X	X
Initial mail search	X	X

Table 5 Configurable Preferences

Preferences	COS	Account
Show search string	X	X
Group mail by conversations	X	X
Enable address for new mail notification and add address		X
Enable mail signature		X
Show time-zone list in appointment view	X	

Additional Account Options

Email Aliases

An alias is an email address that forwards all email it receives to another email account. It is not an email account. An unlimited number of email aliases can be created for an account. Email sent to an alias address is automatically forwarded to the user's email account.

Email Forwarding

When setting up an account, you can define different email addresses to forward mail. A copy of mail messages sent to the designated account are immediately forwarded to the designated forward-address.

Email forwarding only can be removed or changed from the administration console.

Creating Forward-Only Accounts

Commonly used accounts such as postmaster@domain.com, info @domain.com, and webmaster@domain.com can be set up as forward-only accounts. Create the accounts, but not a mailbox by using lqprov and explicitly leave the liquidMailHost attribute blank.

Type **lqprov ca account name@domain.com liquidMailHost““liquidMailForwardingAddress forwardname@domain.coom**

Displaying HTML in an Email

You can determine whether or not to render the HTML formatting of email messages that contain such formatting. The emails will look nicer for the end users than plain text, but you may not want to display inline images as part of the email.

The images can be a hypertext reference (HREF) link to a site, rather than file attachments. If the user opens the email message, a spammer can tell that an email address is valid by tracking image downloads from the referenced web site.

Users Preferences

End-users can further customize their mailboxes when they log on to the Zimbra client. The options they modify overrule the account and COS preference settings. The end user preferences include a General tab, Mail tab, Filter tab, and Contacts tab.

General

Users can:

- Change their passwords
- Select whether to include Junk and Trash folders in their search folders
- Select to always show the search string in the search field
- Set the initial calendar view

Note: *If Microsoft AD is used for user authentication instead of Zimbra LDAP, you must disable the user's Set Password in the Class of Service. The Change password option is not displayed.*

Mail

Users can define the following features for their mailbox's behavior:

- Default view to use (by conversation or by mail message).
- The number of messages to display on a page.
- Whether to save copies of outbound messages to the **Sent** folder.
- Reply-to address.
- Reply and forwarding preferences, whether to include original text, and if so, as inline text or separate attachment.
- Whether to automatically append a signature to their outgoing mail messages and what the text should be.
- Enable a vacation/out of office message and what the text should be.
- Whether to generate new mail notification and if so, to which email address notifications should be sent.
- Whether to view mail as HTML for messages that include HTML codes for formatting. The default is to display message as plain text.
- Enable another email address to receive new mail notification.

Filter Rules

Users can define a set of rules and corresponding actions to apply to incoming mail. When an incoming mail message matches the conditions of a filter rule, the corresponding actions associated with that rules are applied.

Contacts

Users can set preferences for how they want to view their contacts list and whether to automatically add new addresses to their contacts list when new addresses are typed in **To**, **cc**, or **bcc** address fields during message compose.

Chapter 9 Monitoring Zimbra Servers

Checking the overall health of the system as a whole is beyond the scope of this document, which only describes monitoring of the Zimbra server portion. Administrators may want the freedom to implement a variety of monitoring tools and techniques.

Selected error messages generate SNMP traps, which can be monitored using a SNMP tool such as the one included with Zimbra.

Log Files

Zimbra and Zimbra-related processes generate the following types of log files:

- **Local logs** created by each of the following processes
  - Tomcat server logs
  - MySQL binary logs
  - Lucene logs
  - Postfix logs
  - OpenLdap logs
- **Syslog** file. This is written by the operating system, and contains a subset of the messages written to the local logs. SNMP monitoring typically looks at the syslog file and generates traps for critical errors.

Using log4j to Configure Logging

The Zimbra server uses **log4j**, a Java logging package. By default, the Zimbra server has **log4j** configured to log to the local system log. You can configure **log4j** to direct output to another location.

Logging Levels

The levels for Zimbra logging messages are shown below, along with which ones generate SNMP traps and where, by default, each message type is logged.

Table 1 Zimbra Logging Levels

Level	Local?	Syslog ?	SNMP Trap?	When Used
Critical	Y	Y	Y	A component is down, such as disk full
Error	Y	Y	N	Single user error, unexpected; for example, can’t open index
Warnin g	Y	N	N	Non-fatal error for operation, such as user login failed
Info *	Y	N	N *	Transaction-level logging, such as “user X logged in”
Debug	Y	N	N	Parameters to transactions

* A few non-critical messages such as service startup messages, will generate traps.

Server Statistics

You can view server statistics through the Zimbra administration console showing, by message bytes or message count:

- Number of messages or bytes delivered
  - Per server
  - Total
- Disk storage, percentage free or used. The server will not function if a disk partition fills up past a specified threshold.

You can choose a time window as well, to view data for the last 24 hours, the last 3 months, or the last 12 months.

Using a Monitoring Server

In environments with more than one Zimbra server, one of these servers is designated as the “monitor host”. The Zimbra monitor host is responsible for checking the status of all the other Zimbra servers and presenting this information to the Zimbra administration console.

SNMP

SNMP Monitoring Tools

You will probably want to implement server monitoring software in order to monitor system logs, CPU and disk usage, and other runtime information.

Zimbra uses swatch to watch the syslog output to generate SNMP traps.

SNMP Configuration

Zimbra includes an installer package with SNMP monitoring. This package should be run on every server (Zimbra, OpenLDAP, and Postfix) that is part of the Zimbra configuration.

The only SNMP configuration is the destination host to which traps should be sent.

Errors Generating SNMP Traps

The Zimbra error message that generates SNMP traps is when a service is stopped or is started. You can capture these messages using SNMP monitoring software such as the one included with Zimbra, and direct selected messages to a pager or other alert system.

Chapter 10 Backup and Restore

Backing up the Zimbra server on a regular basis can help you quickly restore your mail service, if there is an unexpected crash. The backup process writes a consistent snapshot of mailboxes to a designated backup directory.

The Zimbra backup and restore procedures are run as CLI commands. The following utilities are provided to create backup schedule, perform full and incremental backups, restore the mail server, or restore the LDAP server.

- **lqschedulebackup**. This command is used to schedule full backups and incremental backups and adds the commands to your cron table.
- **lqbackup**. This command performs full or incremental backup of the mail server. This is run on a live server, i.e., while the Tomcat process and the mailbox server are running. In addition to full and incremental backup of the server, the lqbackup command can be used to backup specific accounts for archiving purposes.
- **lqbackupabort**. This command stops a full backup that is in process.
- **lqbackupquery**. This command lists the information about ongoing and completed backups, including labels and dates.
- **lqrestore**. This command preforms a full or incremental restore to the mail server. The lqrestore command is performed on a server that is running.
- **lqrestoreoffline.** This command restores the Zimbra mail server when the Tomcat process is stopped. This tool is run for disaster recovery. The reason is that normal server startup sequence will re-run the latest redo log, but in a disaster recovery scenario you would not want to do this until after performing a full restore followed by an incremental restore.
- **ldaprestoreldap.** This command restores the complete LDAP directory server, including accounts, domains, servers, COS and other data.

This chapter describes how to use these tools to backup or restore your Zimbra server. In addition, this chapter also provides information and general guidelines for disaster recovery. Refer to Appendix B, CLI Commands for usage and definitions about each of these commands.

Important: *Custom configurations, such as Tomcat's server.xml, are not backed up.*

Zimbra Backup and Restore

A full backup backs up all the information needed to restore mailboxes, including the LDAP directory server, data store database, index directory and message directory for each mailbox.

An incremental backup backs up the LDAP directory server and gathers all the redo log transactions written since the last incremental backup or the last full backup, if this is the first incremental since the last full backup was run. If the incremental backup process finds no previous full backup for a mailbox, a full backup is performed on that mailbox.

When backing up shared messages, the backup process looks to see whether a BLOB representing a message already exists on the backup. If it does, it simply flags this object as such and does not copy its content again.

Incremental backups move the redo logs to the backup directory. The redo logs are a journal of every activity that has been logged. They contain a full copy of all BLOBs delivered, as well as metadata such as tags, contacts, and conversations.

These backup files can be used to restore the complete Zimbra system or individual mailboxes so that account and message data is completely restored.

Note: *The Zimbra MTA is not backed up as the data is only on the server for a very short time.*

LDAP Directory Server Backup

The LDAP directory is backed up as part of either the full or incremental backup process. All accounts, domains, servers, COS, and other data are backed up. You can restore the LDAP directory without restoring the message server, and restore specific accounts to the LDAP server.

Redo Logs and Backup

Each Zimbra server generates redo logs that contain every transaction processed by that server. If an unexpected shutdown occurs to the server, the redo logs are used for the following:

- To ensure that no uncommitted transactions remain, the server rereads the redo logs upon startup.
- To recover data written since the last full backup in the event of a server failure.

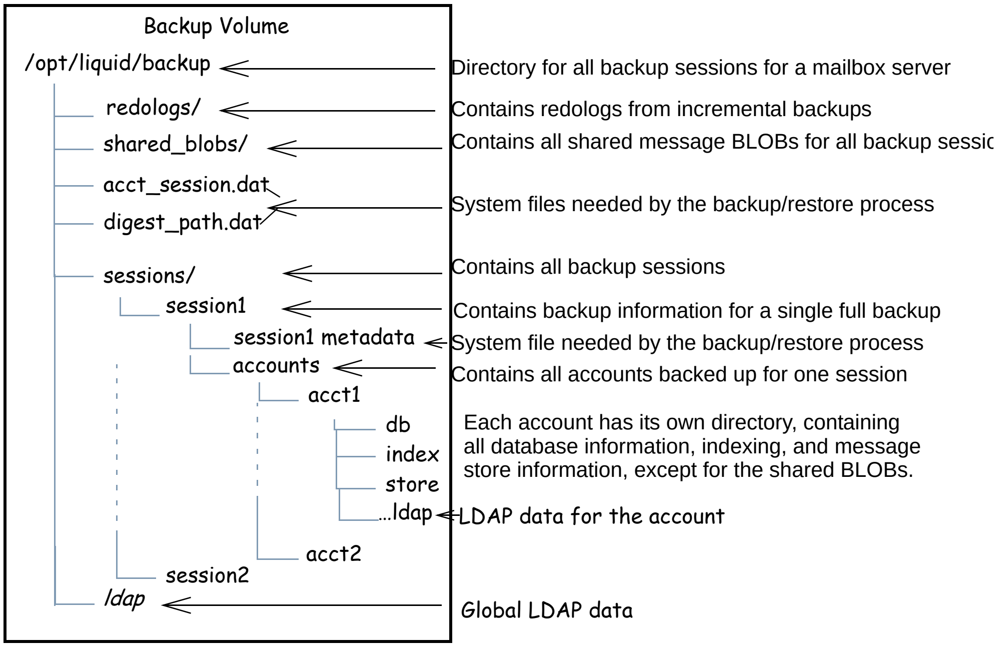
If the server is restored using the **lqrestore** command, after the backed up files are fully restored, any redo logs in the archive and redo logs in use are replayed to bring the system to the point before the failure.

Directory Structure for Backup Files

The Zimbra default backup directory is **opt/liquid/backup**. The backup directory structure created by the backup process is shown in Figure 7. You can run regularly scheduled backups to the same target area without overwriting previous backup sessions.

Note: *Keeping the same backup target saves disk space, because shared binary large object files (BLOB) and other files do not have to be duplicated every time the backup process runs.*

Figure 7: Backup directory structure


Backup Volume
/opt/liquid/backup
Directory for all backup sessions for a mailbox server
redologs/
Contains redologs from incremental backups
shared_blobs/
Contains all shared message BLOBs for all backup sessic
acct_session.dat
digest_path.dat
System files needed by the backup/restore process
sessions/
Contains all backup sessions
session1
Contains backup information for a single full backup
session1 metadata
System file needed by the backup/restore process
accounts
Contains all accounts backed up for one session
acct1
db
index
store
...ldap
Each account has its own directory, containing all database information, indexing, and message store information, except for the shared BLOBs.
LDAP data for the account
acct2
session2
ldap
Global LDAP data

The default directory for the Zimbra backups is on the same mail server as is being backed up. This backup directory would be able to be used to restore a single user account, but in the case of a major crash, the backup files may not be accessible. Good practice is to save backup files to a disk that is not on the mail server.

When you run the backup, specify the directory location. This information then can be transferred to tape.

Scheduling Backups

Zimbra CLI commands, **lqschedulebackup**, is used to write a command to schedule full or incremental backups and to add the command to your cron table.

A recommended schedule would have incremental backups run daily and a full backup run weekly. The Zimbra backup process writes a consistent snapshot of mailboxes to the designated backup directory. Files are not overwritten. You can optionally move the backup files to tape.

When Zimbra Collaboration Suite is installed, no backup schedule is configured. You can schedule backups for anytime or you can enter the command **lqschedulebackup -D** for the default schedule. The default full backup is scheduled for 1:00 a.m., every Sunday. The default incremental backups are scheduled for 1:00 a.m., Monday through Saturday.

The **lqscheudlebackup** command allows you to do the following:

- Enter specific times and days for a backup
- Add a new backup time to your current schedule
- Replace a current backup schedule with another schedule
- Review your backup schedule
- Save the schedule command to a text file. This would allow you to easily recreate the same instructions, in case you need to completely restore the Zimbra system.

See Appendix B: Command-Line Utilities, **lqschedulebackup** for details about how to use the command.

Backing up the Mailbox Server

Backups are run using the Zimbra CLI command, **lqbackup**. This command performs full backups and incremental backups for all the accounts on a designated server or for specified accounts on a designated mailbox server. By default, the backup files are saved to the server’s backup directory.

Full Backup Process

The full backup process goes through the following steps to backup the message store, the data store, the indexes, and the LDAP directory :

1. Backs up the LDAP directory.
2. Iterates through each mailbox to be backed up.
3. Places each mailbox into “maintenance mode” to temporarily block mail delivery and user access to that mailbox.
4. Backs up the mailbox.
   a. Creates MySQL dump for all entries related to that mailbox.
   b. Creates a backup of the index directory for that mailbox.
   c. Backs up the message directory for that mailbox.

4. Returns that mailbox to “active mode” and moves on to the next one.

Incremental Backup Process

The incremental backup process is run with the CLI **lqbackup** tool. The process for incremental backup is as follows:

1. Backs up the LDAP directory.
2. Determines which accounts to backup.
3. Looks in the backup target directory to find the latest full backup for that account.
4. Moves the redo logs since the last backup.

   If no full backup for this account is found, the backup process performs a full backup on this account, even if only an incremental backup was specified.

Examples Backup Commands

- Perform a full backup of all mailboxes on **server1** to target at /mnt/disk.

  lqbackup -f -s server1.domain.com -a all -t /mnt/disk

- Perform incremental backup of all mailboxes on **server1** since last full backup.

  lqbackup -i -s server1.domain.com -a all -t /mnt/disk

- Perform full backup of only **user1**'s mailbox on **server1**.

  lqbackup -f -a all -s server1 -a user1@domain.com

- Perform incremental backup of **user1**'s mailbox on **server1**.

  lqbackup -i -s server1 -a user1@domain.com

Aborting Full Backup In Progress

To stop a backup that is in progress, run the CLI command, **lqbackupaport**. The backup is immediately stopped and the files are marked as a failed session.

Before you can abort a backup you must know its label. If you do not know the full backup label, use **lqbackupquery** to find the label name.

Example

- Stop the backup, if you know the label name

  lqbackupabort -lb backup200507121559510 -s server1

- Stop the backup, if you do not know the label

  a. **lqbackupquery -s server1 -l 2005/07/12**

  b. **lqbackupabory -s server1 -lb backup200507121559510**

Finding Specific Backup Files

The **lqbackupquery** command is used to find full backup sets. The command can be used to find a specific full backup set, full backup sets since a specific date, or all backup sets in the backup directory.

Example

See a list of all the backup labels

```
lqbackupquery -s server1
```

Restoring the Files

Three types of restore procedures can be run.

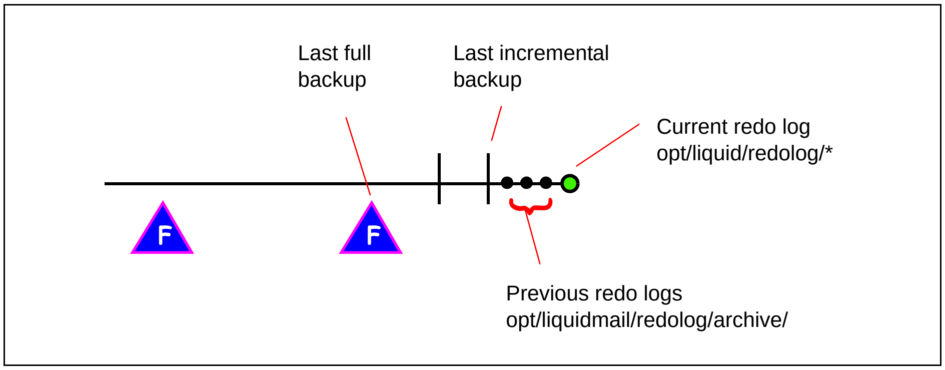
- The **lqrestore** command is used to restore the mail service while the application is running.
- The **lqrestoreoffline** is used to restore the mail server when the mail server is down. This command is run for disaster recovery.
- The **lqrestoreldap** is used to restore the content of the LDAP directory server.

The lqrestore processes mirror the lqbackup processes in that full backup and incremental backup of all accounts or of individual accounts can be specified.

The speed of the full versus incremental backup is inversely proportional to the ease of restoring. Full backups take longer to run, but the restore from a full backup is much faster. Incremental backups are quicker than full backups, but restoring from incremental backups is slower, because it re-applies the transactions in the redo logs.

Figure 8 shows, in order, each of the mail server items that would be required for a full point-in-time recovery. When a system is restored, the last full backup is restored and then each incremental backup since the last backup is restored.

Figure 8: Sample backup timeline


Last full backup
Last incremental backup
Current redo log opt/liquid/redolog/*
F
F
Previous redo logs opt/liquidmail/redolog/archive/

Restore Process

The **lqrestore** process goes through the following steps to restore the message store, the data store, the indexes, and the LDAP directory.

Before you run the restore process, from the administration console, put the accounts to be restored in maintenance mode. After the restore is complete, return the accounts to active.

The restore process goes through the following steps:

1. Retrieves specified accounts of mailboxes to be restored. If the command-line does not specify any mailbox address, the list of all mailboxes on the specified mail host are retrieved from Zimbra LDAP directory server.
2. Iterates through each mailbox:
   a. Deletes the mailbox on the server
   b. Restore the last full backup from the backup area
   c. Restores all incremental backups for that mailbox in order, since last full backup. This involves replaying the redo logs from the backup target area
   d. Restores all archived redo logs for that mailbox, from the redo log archive area on the mailbox server

Example

- Perform complete restore of all accounts on **server1**, including last full backup and any incremental backups since last full backup.

```
lqrestore -s server1.domain.com
```

- Perform restore only to last full backup, excluding incremental backups since then, for all accounts on **server1**.

```
lqbackup -rf -s server1.domain.com
```

The **lqrestoreoffline** command is run when the Tomcat process is stopped.

The restore offline process goes through the following steps to restore the message store, the data store, the indexes, and the LDAP directory:

1. Retrieves the list of all accounts on the specified mail host from Zimbra LDAP directory server.
2. Iterates through each mailbox:
   a. Deletes the mailbox on the server
   b. Restore the last full backup from the backup area
   c. Restores all incremental backups for that mailbox in order, since last full backup. This involves replaying the redo logs from the backup target area
   d. Restores all archived redo logs for that mailbox, from the redo log archive area on the mailbox server

Example

lqrestoreoffline -s server1

Restoring the LDAP Server

The restore process uses the LDAP directory server to look up mailbox account information. Therefore, in a disaster recovery where you need to restore not just one server, but the entire system including all devices, you should restore your directory server first.

The **lqrestoreldap** command restores the LDAP directory server. You can restore the complete LDAP serer, which recreates the entire schema or you can restore specific accounts. The restore command has to be run on the LDAP server being restored.

Examples

- Restore the complete LDAP directory server

  lqrestoreldap -f

- Restore specific accounts

  lqrestoreldap -f -a tac@abc.com jane@abc.com

Disaster Recovery for Specific Situations

This section provides general guidelines for disaster recovery. To cover every possible situation would not be possible, because the steps might vary depending on setup and configuration.

Crash Recovery: Server Startup Sequence

When you system is unexpectedly stopped and then restarted, on startup, the server automatically searches the redo log for any uncommitted transactions, and replays any that it finds. Replaying the redo logs brings the system to a consistent state in the event that the server was stopped due to a power loss or other event.

General Steps for Disaster Recovery

For a general disaster scenario involving multiple machines, do the following:

1. Bring your LDAP directory server to a known good state before doing anything with the Zimbra server. (**lqrestoreldap**)

2. Put all Zimbra mailboxes into maintenance mode, to prevent mail delivery and/or user login. The Zimbra server should not be running.

3. Restore mailbox server application files.

4. Restore mailboxes.

5. Start Zimbra server.
6. Put all Zimbra mailboxes back in active status.
7. Run a full backup of the server. (**lqbackup -f**)

Restore the Zimbra Collaboration Suite Servers

This would be in the case of complete computer failure.

1. Reinstall the Zimbra Collaboration Suite. See the Installation Guide.
2. Restore the LDAP server (**lqrestoreldap**)
3. Run the **lqrestoreoffline** to restore all account mailbox and messages
4. Start the Zimbra server (**lqcontrol startup**)
5. Go to the administration console to verify that the accounts are set to active. (**Accounts>General** tab)

Restoring Individual Accounts on a Live System

Use the **lqrestore** command to restore one or more selected accounts. In the event that a user's mailbox has become corrupted, you might want to restore that user from the last full and incremental backup sets.

Note: *You can restore one account at a time from Accounts on the administration console.*

To restore using the CLI command

1. From the administration console, **Accounts>General** tab, change the account status to Maintenance.

   The maintenance mode prevents delivery of new emails during the restore. Otherwise, the emails would be overwritten during the restore process.

2. Run the **lqrestore** command to restore the accounts. Use commas between accounts.

   ```
   lqrestore -s <server> -a (account@abc.com, account@abc.com)
   ```

3. Go to the administration console to put the account mailbox back to Active status.

Note: *If an user account is restored and the COS that the account was assigned no longer exists, the default COS is assigned to the account.*

LDAP Master corrupted with no replicas

1. Reinstall the LDAP server. See the Installation Guide.
2. Run the **lqrestoreldap** command, with no arguments to restore all accounts, domains, servers, COS, etc. for the LDAP server.

3. Run the **lqrestoreldap** command, with no arguments to restore all accounts, domains, servers, COS, etc. for the LDAP server.
4. Run the **lqrestore** command.

   lqrestore -s <server>
5. Go to the administration console to put the account mailboxes back to active status.

Appendix A Glossary

The Glossary lists terms and acronyms used in this document, and includes both industry terms and application-specific terms. If a general industry concept or practice has been implemented in a specific way within the product, that is noted as well.

Account Policy

Class of Service as exposed in Zimbra administration console.

AD

Microsoft Active Directory Server. Used in Zimbra as an optional choice for authentication and GAL, along with OpenLDAP for all other Zimbra functions.

Alias

An “also known as” email address, which should be routed to a user at a different email address.

Attribute

Contains object-related data for directory server entries. Attributes store information such as a server host name or email forwarding address.

Authentication

Process by which user-supplied login information is used to validate that user’s authority to enter a system.

Blacklist

Anti-spam term, indicates a known bad IP address. This could be one that has been hijacked by spammers, or also one from a poorly maintained but legitimate site that allows mail relaying from unauthorized parties.

BLOB

Binary Large Object file.

Class of Service (COS)

Describes an object in the Zimbra LDAP data schema, which contains settings for things like user mail quotas. Each Zimbra account includes a COS, and the account inherits all the settings from the selected COS.

CLI

Command-Line Interface. Used to refer to the collective set of Zimbra command-line tools, such as lqbackup and lqprov.

Cluster

A type of network configuration for high availability, using clusters of servers (nodes). If one server fails or drops off the network, a spare takes over.

Contacts

Within Zimbra, Contacts are a user-interface feature listing that user’s personal collection of address and contact information.

Conversation

Within Zimbra, Conversations are a user-interface feature that presents email threads (emails sharing the same subject line) as a single Conversation listing. Users can expand the Conversation to view all emails within it.

DHTML

Dynamic HTML. A technology employed in the Zimbra Web Client.

DNS

Domain Name Services.

Edge MTA

Generic term used to refer to any mail transfer agent that is the first line of defense in handling incoming email traffic. Functions that may occur on the Edge MTA include spam filtering.

Entry

An item in the directory server, such as an account or mail host.

Failover

Takeover process where a spare server machine detects that a main server is unavailable, and the spare takes over processing for that server.

GAL

Global Address List, the Outlook version of a company directory. Lists contact information, including email addresses, for all employees within an organization.

Global Configuration

A Zimbra object containing default settings for servers and Class of Service.

HTTP

HyperText Transfer Protocol, used along with SOAP for UI integration.

IMAP

Internet Message Access Protocol is a method of accessing mail from a remote message store as if the users were local.

Index Store

Within Zimbra, a directory area that stores all the indexing information for mail messages on a particular mailbox server.

Indexing

The process of parsing incoming email messages for search words.

Java

Java is an industry standard object-oriented programming language. Used for the core Zimbra application server.

JavaScript

Scripting largely developed by Netscape that can interact with HTML source code. Technology used in the Zimbra Web Client.

LDAP

Lightweight Directory Access Protocol, an industry standard protocol used for authentication.

Zimbra administration console

The Zimbra administrator interface.

Zimbra Web Client

The Zimbra end-user interface.

LMTP

Local Mail Transfer Protocol, used for transferring messages from Postfix MTA to the Zimbra server for final delivery.

Mailbox Server

Alternative term for Zimbra server.

MAPI

Messaging Application Programming Interface. A system built into Microsoft Windows to enable different email applications to work together.

Message Store

Within Zimbra, a directory area that stores the mail messages on a particular mailbox server.

MDA

Mail Delivery Agent, sometimes known as a mail host. The Zimbra server functions as an MDA.

Metadata

Data that describes other data, rather than actual content. Within Zimbra, metadata consists of user folders, threads, message titles and tags, and pointers.

MIME

Multipurpose Internet Mail Extensions, a specification for formatting non-ASCII Internet message content such as image files. Format used to store messages in Message Store.

MTA

Message Transfer Agent. MTA is a program that delivers mail and transports it between machines. A Zimbra deployment assumes both the Postfix MTA and an edge MTA.

MX (Record)

Mail eXchange. Entry on a DNS server to help lookup.

OOTO

Common shorthand for “out of the office”, used when sending vacation messages.

Open Source

Refers to software created by groups of users for non-commercial distribution, where source code is published rather than proprietary.

OS

Operating system, such as Linux, UNIX, or Microsoft Windows.

POP

Post Office Protocol is used to retrieve email from a remote server over TCP/IP and save it to the local computer.

Provisioning

The process of creating accounts or other data, usually in batch or automated fashion.

RBH

Real-time black hole. Usually refers to web sites that, as a public service, provide lists of known bad IP addresses from which mail should be blocked, because the servers are either known to be spammers, or are unsecured and exploited by spammers.

Redo Logs

Detailed transaction log for the Zimbra server, used for replay and replication.

SAN

Storage Array Network. A high-availability data storage area.

Schema

Describes the data structures in use for by directory services at a particular organizational site.

SMTP

Simple Mail Transfer Protocol. Used in Zimbra deployments between the Edge MTA and the Postfix MTA.

SNMP

Simple Network Monitoring Protocol. Used by monitoring software to pick up critical errors from system logs.

SOAP

Simple Object Access Protocol, an XML-based messaging protocol used for sending requests for Web services. The Zimbra servers use SOAP for receiving and processing requests, which can come from Zimbra command-line tools or Zimbra user interfaces.

Spam

Unsolicited commercial email. Spammers refer to their output as “bulk business email”.

SQL

Structured Query Language, used to look up messages in the Message Store.

SSL

Secure Sockets Layer.

Tags

A Zimbra Web Client feature. Users can define tags and apply them to mail messages for searching.

TCO

Total Cost of Ownership. Zimbra reduces total cost of ownership (TCO) by reducing requirements for server hardware, OS licensing fees, supporting application license fees, disk storage requirements, and personnel (IT, help desk, consulting).

TLS

Transport Layer Security.

UCE

Unsolicited commercial email, also known as spam.

Virtual Alias

A type of mail alias recognized in the Postfix MTA.

Whitelist

Anti-spam term for a known good mail or IP address. Mail coming from such an address may be “automatically trusted”.

XML

eXtended Markup Language.

Appendix B Command-Line Utilities

Command Line Interface (CLI) can be used to create, modify and delete certain features and functions of the Zimbra Collaboration Suite. The administration console is the main tool for maintaining the Zimbra Collaboration Suite, but some functions can only be changed from the CLI utility.

The CLI utility can be used for the following:

- Provisioning accounts*
- Back up and restore
- Start and stop a service
- Move mailboxes
- Install self-signed certificates
- Local configuration (?)

*This function should be performed from the administration console.

General Tool Information

The Zimbra command-line utilities follow standard UNIX command-line conventions.

Long Name	Option	Description and Example
-- help	**-h**	Displays the usage options for the tool. Example: **lqmailboxmove -h** lists all the options available for the **lqmailboxmove** utility.
--mailbox	**-m**	Specify a mailbox. Type the full email address.
--server	**-s**	Specify a server or host name.

Syntax Conventions

When demonstrating the syntax of each tool, the following conventions indicate required, optional, and alternate values:

- {attribute} in curly brackets is required information.
- [attribute] in square brackets are optional arguments or information.
- {a|b|c} or [a|b|c] options separated by the pipe character | means “a” OR “b” OR “c”
- For attribute names that may contain spaces, surround the name with double quotes.

Location of Command-Line Utilities

The command-line tools available for administrators are all located in the **/opt/liquid/bin** directory on the Zimbra server

lqprov (Provisioning)

The **lqprov** tool performs all provisioning tasks in Zimbra LDAP, including creating, aliases, domains, and distribution lists.Each operation is invoked through command-line options, each of which has a long name and a short name. For example, these two commands are equivalent:

```
lqprov createAccount joe@domain.com test123
lqprov ca joe@domain.com test123
```

Long Name	Short Name	Syntax, Example, and Notes
CreateAccount	**ca**	Syntax:{name@domain} {password} [attribute1 value1 etc] lqprov ca joe@domain.com test123 zimbraMailHost server1
DeleteAccount	**da**	Syntax:{name@domain|id|adminName} lqprov da joe@domain.com
GetAccount	**ga**	Syntax:{name@domain|id|adminName} lqprov GetAccount joe@domain.com
GetAllAccounts	**gaa**	[-v] [{domain}] lqprov gaa lqprov gaa -v domain.com
GetAllAdminAccounts	gaaa	

Long Name	Short Name	Syntax, Example, and Notes
ModifyAccount	**ma**	{name@domain|id|adminName} [attribute1 value1 etc] lqprov ma joe@domain.com zimbraAccountStatus maintenance
SetPassword	**sp**	{name@domain|id|adminName} {password} lqprov sp joe@domain.com test321
AddAccountAlias	**aaa**	{name@domain|id|adminName} {alias@domain} lqprov aaa joe@domain.com joe.smith@engr.domain.com
RemoveAccountAlias	**raa**	{name@domain|id|adminName} {alias@domain} lqprov raa joe@domain.com joe.smith@engr.domain.com
SetAccountCOS	**sac**	{name@domain|id|adminName} {cos-name|cos-id} lqprov sac joe@domain.com FieldTechnician
SearchAccounts	**sa**	[-v] {ldap-query}
SearchGAL	**sg**	{domain} {name} lqprov sg joe
RenameAccount	**ra**	{name@domain|id} {newname@domain} lqprov ra joe@domain.com joe23@domain.com
CreateDomain	**cd**	{domain} [attribute1 value1 etc] lqprov cd mktng.domain.com liquidAuthMech zimbra
DeleteDomain	**dd**	{domain} lqprov dd mktng.domain.com
GetDomain	**gd**	{domain|id} lqprov gd mktng.domain.com
GetAllDomains	**gad**	[-v]
ModifyDomain	**md**	{domain|id} [attribute1 value1 etc] lqprov md domain.com liquidGalMaxResults 50

Long Name	Short Name	Syntax, Example, and Notes
CreateCos	**cc**	{name} [attribute1 value1 etc] lqprov cc Executive liquidAttachmentsBlocked FALSE liquidAuthTokenLifetime 60m liquidMailQuota 100M liquidMailMessageLifetime 0
DeleteCos	**dc**	{name|id} lqprov dc Executive
GetCos	**gc**	{name|id} lqprov gc Executive
GetAllCos	**gac**	[-v] lqprov gac -v
ModifyCos	**mc**	{name|id} [attribute1 value1 etc] lqprov mc Executive liquidAttachmentsBlocked TRUE
RenameCos	**rc**	{name|id} {newName} lqprov rc Executive Business
CreateServer	**cs**	{name} [attribute1 value1 etc]
DeleteServer	**ds**	{name|id}
GetServer	**gs**	{name|id}
GetAllServers	**gas**	[-v]
ModifyServer	**ms**	{name|id} [attribute1 value1 etc]
GetAllConfig	**gacf**	
GetConfig	**gcf**	{name}
ModifyConfig	**mcf**	
CreateDistributionList	**cdl**	{list@domain} lqprov cdl needlepoint-list@domain.com

Long Name	Short Name	Syntax, Example, and Notes
AddDistributionListMember	**adlm**	{list@domain|id} {member@domain} lqprov adlm needlepoint-list@domain.com singer23@mail.free.net
RemoveDistributionListMember	**rdlm**	{list@domain|id} lqprov rdlm needlepoint-list@domain.com singer23@mail.free.net
GetAlldistributionLists	**gadl**	[-v]
GetDistributionList	**gdl**	{list@domain|id}
DeleteDistributionList	**ddl**	(list@domain|id}

lqbackup

This tool performs full backups, incremental backups for a designated mail host. You can either specify specific accounts, or, if no accounts are specified, all accounts are included.

This utility has short option names and full names. The short option is preceded by a single dash, the full option is proceeded by a double dash. For example, **-fb** is the same as **--fullBackup**.

Syntax

lqbackup {-f | -i} -acct (account) <options>

Description

Long Name	Short Name	Description
--fullBackup	**-f**	Starts a full backup
--incrementalBackup	**-i**	Starts an incremental backup
--account	**-a**	Specifies the account email addresses. Separate accounts with a blank space. Enter all to backup all accounts.

Common Options

Long Name	Short Name	Description
--server	**-s**	Mail server host name. For format, use either the plain host name or the server.domain.com name. The default is the localhost name.
--target	**-t**	Specifics the target backup location. The default is <liquid_home>/backup.
--help	**-h**	Displays the usage options for this command.

Examples

In these examples, the server (-s) is `server1.domain.com`. The (-t) is not required if the target is the default directory, **(liquid_home/backup**).

- Perform a full backup of all mailboxes on **server1**.

```
lqbackup -f -a all -s server1.domain.com
```

- Perform incremental backup of all mailboxes on **server1** since last full backup.

```
lqbackup -i -a all -s server1.domain.com
```

- Perform full backup of only **user1**'s mailbox on **server1**. Note that hostname does not need full domain if account is used.

```
lqbackup -f -s server1 -a user1@domain.com
```

- Perform incremental backup of **user1**'s mailbox on **server1**.

```
lqbackup -i -s server1 -a user1@domain.com
```

lqschedulebackup

This command is used to schedule backups and add the command to your cron table. Familiarity with the cron format is helpful.

The default schedule is as follows:

- Full backup, every Sunday at 1:00 a.m. (0 1 * * 0)
- Incremental backup, Monday through Saturday at 1:00 a.m. (0 1 * * 1-6)

Each crontab entry is a single line composed of five fields separated by a blank space. Specify the fields as follows:

minute	0 through 59
hour	0 through 23
day of month	1 through 31

month	1 through 12
day of week	0 through 7 (0 or 7 is Sunday)

Type an asterisk (*) in the fields you are not using.

This command automatically writes the schedule to the crontab.

Syntax

lqschedulebackup [-q]|-s|-A|-R|-F|-D] [schedule] [schedule...]

Description

Name	Command Name	Description
	i:	Incremental backup
	f:	full backup
Common Options		
Query	-q	Default command, displays the existing Zimbra backup schedule.
Save	-s	Save the schedule. Allows you to save the schedule command to a text file so that you can quickly regenerate the backup schedule when the system is restored.
Flush	-F	Removes the current schedule and cancels all scheduled backups
Append	-A	Adds an additional specified backup to the current schedule
Replace	-R	Replaces the current schedule with the specified schedule.
Default	-D	Replaces the current schedule with the default schedule
Help	**-h**	Displays the usage options for this command

Examples

- To schedule the default full and incremental backup

  lqschedulebackup [-D**]**

- To replace the existing schedule with a new schedule

  lqschedulebackup -R f [schedule] [schedule]

- To add an additional full backup to the existing schedule

  lqschedulebackup -A f [schedule]

- To add an additional incremental backup to the existing schedule

  ```
  lqschedulebackup -A i [schedule]
  ```

- To display the existing schedules a

  ```
  lqschedulebackup -q
  ```

- To display the schedules on one line as a command, so that they can be copied to a text file and saved to be used if the application needs to be restored.

  ```
  lqschedulebackup -s
  ```

lqbackupabort (Aborting a Full Backup)

The lqabort command can be used to stop a backup process. Before you can abort an account you must know backup label. This label is displayed after you start the backup procedure. If you do not know the label, use the **lqbackupquery** to find the label name.

Syntax

lqbackupabort -lb (label) -acct (email addresses) -s (hostname) -t (liquid-home/backup)

Description

Long Name	Short Name	Description
--label <label>	**-lb**	Label of the full backup to be aborted. Use the **lqbackupquery**, to find the file name.
--account	**-acct**	Specifies the account email addresses. Separate accounts with a blank space. If accounts are omitted, all accounts are backed up.
Common Options		
--server	**-s**	Mail server host name. For format, use either the plain host name or the server.domain.com name. The default is the localhost name.
--target	**-t**	Specifies the target backup location. The default is <liquid_home>/backup.
--help	**-h**	Displays the usage options for this command.

lqbackupquery

The **lqbackupquery** command is used to find full backup sets. The command can be used to find a specific full backup set, full backup sets since a specific date, or all backup sets in the backup directory.

Syntax

lqbackupquery <options>

Description

Long Name	Short Name	Description
--label <label>	-lb	The label of the full backup set to query. An example of a label is **backup200507121559510**
--list <date>	-l	If a date is used, the query returns a list of full backup sets since the specified date. Enter date as **YYYY/MM/DD hh:mm:ss**. The hours, minutes, and seconds are not required.
--verbose	-v	Returns detailed status information about what was backup up. Lists all the accounts in the backup.
Common Options		
--server	**-s**	Mail server host name. For format, use either the plain host name or the server.domain.com name. The default is the localhost name.
--target	**-t**	Specifies the backup target location. The default is <liquid_home>/backup.
--help	**-h**	Displays the usage options for this command.

lqrestore

This tool performs full restores and incremental restores, for a designated mail host. You can either specify specific accounts, or, if no accounts are specified, all accounts are in the backup are restored.

This utility has short option names and full names. The short option is preceded by a single dash, the full option is proceeded by a double dash. For example, **-rb** is the same as **--restorefullBackupOnly**.

Syntax

lqrestore <-f | -i> -a (account) <options>

Description

Long Name	Short Name	Description
--label	**-lb**	The label of the full backup to restore to ??????
--restorefullBackupOnly	**-rf**	Restores to the full backup only, not any incremental backups since that backup.
--restoreAccount	**-ra**	Restores the account in directory service..????
--prefix	-pre	<prefix> The prefix to prepend to the original account names. WHY>> what can a prefix consist of?
--createAccount	-ca	Restores accounts to new target accounts whose names are prepended with <prefix>
--account	-acct	Specifies the account email addresses. Separate accounts with a blank space. Type all to restore all accounts.
Common Options		
--server	**-s**	Mail server host name. For format, use either the plain host name or the server.domain.com name. The default is the localhost name.
--target	**-t**	Specifies the backup target location. The default is <liquid_home>/backup.
--help	**-h**	Displays the usage options for this command.

Examples

- Perform complete restore of all accounts on **server1**, including last full backup and any incremental backups since last full backup.

```
lqrestore -acct all -s server1.domain.com
```

- Perform restore only to last full backup, excluding incremental backups since then, for all accounts on **server1**.

```
lqbackup -rf -acct all -s server1.domain.com
```

lqrestoreoffline (Offline Restore)

This tool is run when the Zimbra server (i.e., the Tomcat process) is down. The MySQL database for the server and the OpenLDAP directory server must be running before you start the lqrestoreoffline command.

Syntax

lqrestoreoffline <options> [<accounts>]

Description

Long Name	Short Name	Description
--label	**-lb**	The label of the full backup to restore. Type this label to specify a backup file other then the latest.
--restorefullBackupOnly	**-rf**	Restores to the full backup only, not any incremental backups since that backup.
--restoreAccount	**-ra**	Restores the account in directory service.
--prefix	-pr	<prefix> The prefix to prepend to the original account names. WHY>> what can a prefix consist of?
--createAccount	-ca	Restores accounts to new target accounts whose names are prepended with <prefix>
--account	-acct	Specifies the account email addresses. Separate accounts with a blank space. If accounts are omitted, all accounts are restored.
Common Options		
--server	**-s**	Mail server host name. For format, use either the plain host name or the server.domain.com name. The default is the localhost name.
--target	**-t**	Specifies the backup target location. The default is <liquid_home>/backup.
--help	**-h**	Displays the usage options for this command.

Examples

Before you begin lqrestoreoffline, the LDAP directory server must be running

- Perform a complete restore of all accounts on **server1**, including last full backup and any incremental backups since last full backup.

```
lqrestoreoffline -s server1.domain.com
```

lqcontrol (Start/Stop Service)

This command is run to start or to stop services.

Syntax

lqcontrol [-v -h] command [args]

Description

Long Name	Short Name	Description
	-v	display Liquid software version.
	-h	Displays the usage options for this command.
Command in...		
reload		Restarts the manager without affecting other services.
shutdown		Shutdown all services and manager on this host. When the manager is shutdown, you cannot query that status
start		Startup manager and all services on this host
startup		Startup manger and all services on this host
status		Returns services information for the named host
stop		Stop all services but leave the manager running.

moves the mailbox and data files, and updates the metadata on the new server.

lqmailboxmove (Move Mailbox)

This command is used to move selected mailboxes from one Zimbra server to another. This tool the metadata on the new server.

Syntax

maiboxMoveUtil <options> <account email address> <src host>[<port>] <target host>[:<port>]

Description

Long Name	Short Name	Description
--password	-pwd	<**arg**> type the administrator's password
--switchover	-switch	Switch to the new mail host
--username	-user	<**arg**> Administrator user name.

If **--switchover** is specified, the target host will take over as the mail host. Otherwise only the mailbox is migrated to the target host, the original host remains the mail host.

If port is omitted, the default, 80, is assumed.

lqcreatecert (Generate Self-Signed Certificate)

Syntax

lqcreatecert

lqcertinstall (Install Certificate)

Syntax

lqcertinstall mailbox <cert_filename>

lqlocalconfig (Local Configuration)

This tool is set or get the local configuration for a Zimbra server.

Syntax

lqlocalconfig [options] [args]

Description

Long Name	Short Name	Description
--config <arg>	-c	File in which the configuration is stored.
--default	-d	The default values for the keays listedin [args} is listed.
--edit	-e	Edit the configuration file, change keys and values specified. the [args] is in the key=value form.
--force	-f	Edit the keys whose change is known to be potentially dangerous
--help	**-h**	Shows the help for the usage options for this tool.
--info	**-i**	Shows the documentation for the keys listed in [args]
--format <arg>	**-m**	Shows the values in one of these formats: plain (default), xml, shell, nokey.
--changed	-n	Shows the values for only those keys listed in the [args] that have been changed from their defaults.
--path	-p	Shows which configuration file will be used.
--random	-r	This option is used with the edit option. Specified key is set to a random password string.
--show	-s	Forces the display of the password strings.
--expand	-x	Expand values

Index