MBOXADM-19.1

M-Box Administration Guide

Table of Contents

1 Software version

This guide is published in support of Isode M-Box R19.1. It may also be pertinent to later releases. Please consult the release notes for further details.

2 Readership

This guide is intended for administrators who plan to configure M-Box, a high performance Internet Message Store, supporting IMAP (Internet Message Access Protocol) and POP (Post Office Protocol).

3 Related publications

Related topics are discussed in the volumes of the Isode documentation set listed below.

Volume	Title
SWADM-19.1	*M-Switch Administration Guide*
VAUADM-19.1	*M-Vault Administration Guide*

4 Typographical conventions

The text of this manual uses different typefaces to identify different types of objects, such as file names and input to the system. The typeface conventions are shown in the table below.

Object	Example
File and directory names	*isoentities*
Program and macro names	mkpasswd
Input to the system	`cd newdir`
Cross references	see Section 5, “File system place holders”
Additional information to note, or a warning that the system could be damaged by certain actions.	Notes are additional information; cautions are warnings.

5 File system place holders

Where directory names are given in the text, they are often place holders for the names of actual directories where particular files are stored. The actual directory names used depend on how the software is built and installed. All of these directories can be changed by configuration.

Certain configuration files are searched for first in *(ETCDIR)* and then *(SHAREDIR)*, so local copies can override shared information.

The actual directories vary, depending on whether the platform is Windows or UNIX.

Name	Place holder for the directory used to store...	Windows (default)	UNIX
(ETCDIR)	System-specific configuration files.	*C:\Isode\etc*	*/etc/isode*
(SHAREDIR)	Configuration files that may be shared between systems.	*C:\Program Files\Isode\share*	*/opt/isode/share*
(BINDIR)	Programs run by users.	*C:\Program Files\Isode\bin*	*/opt/isode/bin*
(SBINDIR)	Programs run by the system administrators.	*C:\Program Files\Isode\bin*	*/opt/isode/sbin*
(EXECDIR)	Programs run by other programs; for example, M-Switch channel programs.	*C:\Program Files\Isode\bin*	*/opt/isode/libexec*
(LIBDIR)	Libraries.	*C:\Program Files\Isode\bin*	*/opt/isode/lib*
(DATADIR)	Storing local data.	*C:\Isode*	*/var/isode*
(LOGDIR)	Log files.	*C:\Isode\log*	*/var/isode/log*
(CONFPDUSPOOLDIR)	Large PDUs on disk.	*C:\Isode\tmp*	*/var/isode/tmp*
(QUEDIR)	The M-Switch queue.	*C:\Isode\switch*	*/var/isode/switch*
(DSADIR)	The Directory Server's configuration.	*C:\Isode\d3-db*	*/var/isode/d3-db*

6 Support queries and bug reporting

A number of email addresses are available for contacting Isode. Please use the address relevant to the content of your message.

- For all account-related inquiries and issues: customer-service@isode.com. If customers are unsure of which list to use then they should send to this list. The list is monitored daily, and all messages will be responded to.
- For all licensing related issues: license@isode.com.
- For all technical inquiries and problem reports, including documentation issues from customers with support contracts: support@isode.com. Customers should include relevant contact details in initial calls to speed processing. Messages which are continuations of an existing call should include the call ID in the subject line. Customers without support contracts should not use this address.

- For all sales inquiries and similar communication: sales@isode.com.

Bug reports on software releases are welcomed. These may be sent by any means, but electronic mail to the support address listed above is preferred. Please send proposed fixes with the reports if possible. Any reports will be acknowledged, but further action is not guaranteed. Any changes resulting from bug reports may be included in future releases.

Isode sends release announcements and other information to the Isode News email list, which can be subscribed to from the address: http://www.isode.com/company/subscribe.html

7 Conformance

7.1 LMTP

RFC 2033: Local Mail Transfer Protocol.

RFC 3848: ESMTP and LMTP Transmission Types Registration.

RFC 2920: SMTP Service Extension for Command Pipelining.

RFC 1652: SMTP Service Extension for 8-bit MIME transport.

7.2 Sieve

RFC 5228: Sieve: A Mail Filtering Language.

RFC 5231: Sieve Email Filtering: Relational Extension.

RFC 5233: Sieve Email Filtering : Subaddress Extension.

RFC 3894: Sieve Extension: Copying Without Side Effects.

RFC 5230: Sieve Email Filtering: Vacation Extension.

RFC 5232: Sieve Email Filtering: Imap4flags Extension.

RFC 5260: Sieve Email Filtering: Date and Index Extensions.

RFC 5804: A Protocol for Remotely Managing Sieve Scripts.

RFC 5490: The SIEVE mail filtering language - extension for accessing mailbox metadata.

7.3 SASL

RFC 4422: Simple Authentication and Security Layer (SASL).

RFC 2195: IMAP/POP AUTHorize Extension for Simple Challenge/Response.

RFC 4616: The PLAIN Simple Authentication and Security Layer (SASL) Mechanism.

RFC 2831: Using Digest Authentication as a SASL Mechanism.

RFC 4752: The Kerberos V5 ("GSSAPI") Simple Authentication and Security Layer (SASL) Mechanism.

RFC 5802: Salted Challenge Response Authentication Mechanism (SCRAM) SASL and GSS-API Mechanisms.

7.4 IMAP

RFC 3501: INTERNET MESSAGE ACCESS PROTOCOL - Version 4rev1.

RFC 2088: IMAP4 non-synchronizing literals.

RFC 2342: IMAP4 Namespace.

RFC 4315: INTERNET MESSAGE ACCESS PROTOCOL - UIDPLUS extension.

RFC 3691: Internet Message Access Protocol (IMAP) UNSELECT command.

RFC 2177: IMAP 4 IDLE command.

RFC 3503: Message Disposition Notification (MDN) profile for Internet Message Access Protocol (IMAP).

RFC 5092: IMAP URL Scheme.

RFC 4467: Internet Message Access Protocol (IMAP) - URLAUTH extension.

RFC 4469: Internet Message Access Protocol (IMAP) CATENATE extension.

RFC 4551: IMAP Extension for Conditional STORE Operation or Quick Flag Changes Resynchronization.

RFC 4314: IMAP4 ACL extension.

RFC 2087: IMAP4 QUOTA extension.

RFC 4731: IMAP4 extension to SEARCH command for controlling what kind of information is returned.

RFC 3516: IMAP4 Binary Content Extension.

RFC 5256: INTERNET MESSAGE ACCESS PROTOCOL - SORT AND THREAD EXTENSIONS.

RFC 5258: IMAP 4 LIST Command extensions.

RFC 4959: IMAP Extension for SASL Initial Client Response.

RFC 4978: The IMAP COMPRESS Extension.

RFC 5032: WITHIN Search extension to the IMAP Protocol.

RFC 5267: Contexts for IMAP4.

RFC 5162: IMAP4 Extensions for Quick Mailbox Resynchronization.

RFC 5161: The IMAP ENABLE Extension.

RFC 6154: IMAP LIST Extension for Special-Use Mailboxes.

RFC 6851: Internet Message Access Protocol (IMAP) - MOVE Extension.

draft-ietf-morg-inthread-01: The IMAP SEARCH=INTHREAD and THREAD=REFS Extensions.

7.5 POP

RFC 1939: Post Office Protocol - Version 3.

RFC 2449: POP3 Extension Mechanism.

RFC 1734: POP3 AUTHentication command.

RFC 2595: Using TLS with IMAP, POP3 and ACAP.

RFC 5034: The Post Office Protocol (POP3) Simple Authentication and Security Layer (SASL) Authentication Mechanism.

8 Export controls

Many Isode products use TLS (Transport Layer Security) to encrypt data in transit. This means that these products are subject to UK Export Controls.

For some countries (at the time of shipping this release, these comprise all EU countries, United States of America, Canada, Australia, New Zealand, Switzerland, Norway, Japan), these Export Controls can be handled by administrative process as part of evaluation or purchase. For other countries, a special Export License is required. This can be applied for only in context of a purchase order for those Isode products.

You must ensure that you comply with these Export Controls where applicable, i.e. if you are licensing or re-selling Isode products.

The TLS feature of Isode products is enabled by a TLS Product Activation feature. This feature may be turned off, and Isode products without this TLS feature are not export controlled. This can be helpful to support evaluation of Isode products in countries that need a special export license.

Isode products are used to administer sensitive data and so Isode strongly recommends that all operational deployments of Isode products use the export-controlled TLS feature.

All Isode Software is subject to a license agreement and your attention is also called to the export terms of your Isode license.

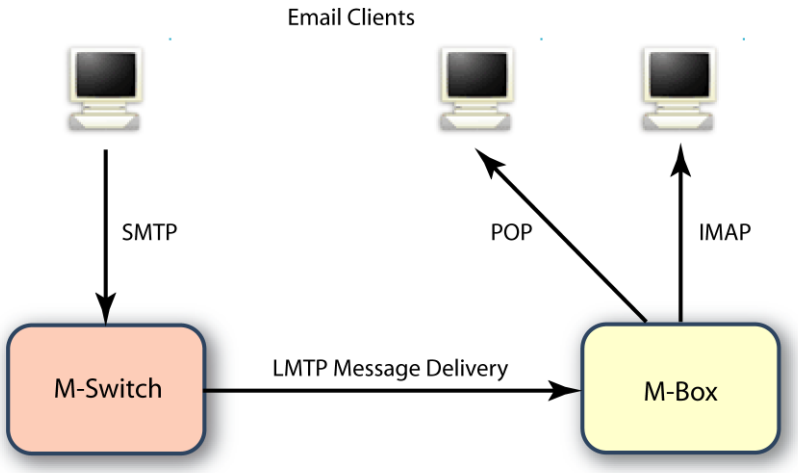
Chapter 1 Introduction to M-Box

This section introduces M-Box and its components, and talks about how its configuration is stored.

1.1 Overview

The Isode IMAP (Internet Message Access Protocol) and POP (Post Office Protocol) server, M-Box, is a scalable enterprise mail store designed for use in small to large enterprise environments using standards-based technologies. Mail User Agents (UA) can access mail from M-Box using the IMAP and POP3 protocols.

Figure 1.1. Email clients accessing messages via M-Box



Mail is delivered from Isode M-Switch or from another Message Transfer Agent to M-Box using LMTP (Local Mail Transfer Protocol).

Mail Filtering rules can be specified using Sieve Mail Filtering Language. Sieve scripts can be managed remotely over ManageSieve Protocol.

1.2 Folders

M-Box stores messages in folders according to the IMAP model. Every user has an **Inbox** folder into which new messages are delivered by default. This **Inbox** can be accessed by either POP or IMAP. Users may create additional folders, which may either appear adjacent to the **Inbox** folder, or as a hierarchical sub-folder of any folder. This gives flexible personal filing of messages.

M-Box also supports shared folders. A hierarchy of mailboxes can be shared by defining a shared root and specifying what kind of access different users have to the hierarchy. This will appear to the user as a special folder, typically called **Shared Folders**. Folders immediately subordinate to **Shared Folders** are created by the administrator, who may permit other users to create sub-folders.

1.3 Configuration overview

M-Box holds server configuration information, and configuration information on M-Box users and shared folders in an LDAP directory, such as Isode M-Vault. Server configuration may be managed using MConsole, which accesses information stored in the directory. This is described in detail in this manual. Server and shared folder configuration may also be managed in the directory by use of other tools.

User information may be directly managed using MConsole and msadm. It may also be loaded into M-Vault from a provisioning system.

1.4 Mail delivery

Messages are delivered from a Message Transfer Agent (MTA) using LMTP, the Local Mail Transfer Protocol, which is a protocol similar to SMTP, and is designed for transferring mail to the final message store. LMTP allows MTAs to deliver "local" mail over a network.

M-Box implements LMTP via the isode.lmtpd daemon. LMTP can either be used over a network via TCP or, on UNIX platforms, local via a UNIX domain socket. The LMTP server supports Sieve filtering (see Chapter 6, *Filtering Messages Using Sieve*).

Chapter 2 M-Box Server Configuration

This section provides detailed description of M-Box server configuration options.

M-Box is normally configured by use of information stored in an LDAP directory. This includes both server configuration information (described in this section) and information on users and shared folders (described in Chapter 3, *M-Box User Management*).

The management of server and user data takes place using the Internet Messaging Administration (IMA) Web application Once *(ETCDIR)/ms.conf* is set up to use LDAP configuration as described in Section 2.1.6.2, “Setting up M-Box to use LDAP configuration”, you can manage M-Box server configuration using IMA. The M-Box server configuration entry specified in the `ldap_basedn` option doesn’t have to exist, it will be created by IMA if its parent entry exists in the Directory.

Information may also be managed directly with LDAP. Relevant LDAP attributes are described in Section 2.2, “Server configuration options”.

A local XML file holds information on the location of the LDAP server used to hold configuration information. This file may also be used to hold all M-Box server configuration information removing the need for an LDAP directory. Details are given in Section 2.2, “Server configuration options”.

Interworking with M-Switch is achieved using LASER based routing. See the *M-Switch Administration Guide* for details on how this is configured.

2.1 Initial configuration

This section describes initial configuration of M-Box. Please read this section to find out about all configuration steps required before starting M-Box for the first time.

This section also provides basic information about services included in M-Box and describes how to start/stop M-Box services once the initial configuration is complete.

2.1.1 M-Box configuration file

The default M-Box configuration file is *(ETCDIR)/ms.conf*. Copy the appropriate config file from one of sample configuration files provided: *(ETCDIR)/ms.conf.local* and *(ETCDIR)/ms.conf.ldap*. The former example file configures M-Box not to use LDAP.

2.1.2 M-Box runtime user

Create the M-Box UNIX runtime user account "mbox" (for example using **useradd** on Linux), or set the value of the `ms_user` option in *(ETCDIR)/ms.conf* to be an existing Unix account.

2.1.3 M-Box filestore

(UNIX only) Create the working directory used by M-Box for storing statistic information, trace logs and other information. This directory must be owned by the M-Box runtime user.

```
# mkdir -p /var/isode/ms
# chown mbox /var/isode/ms
```

Create a mail partition owned by the M-Box runtime user (the default is */var/isode/ms/user*, unless the `userdir` option is set in *`(ETCDIR)`/ms.conf*). On Linux you can run:

```
# mkdir -p /var/isode/ms/user
# chown mbox /var/isode/ms/user
```

If you have configured an alternative runtime user name to the default of "mbox", use that instead in the command lines shown above.

(Windows) On Windows the default working directory is created automatically by the installation. If you want to use a non default location, you need to create the directory.

Runtime user is not used on Windows.

2.1.4 Installing M-Box Product Activation Key (PAK)

Isode Product Activation system supersedes the legacy licensing system used prior to R19.0.

2.1.4.1 Isode Product Activation Overview

Each Isode product has one main feature and a number of possible sub-features which can be activated. For example the **M-Box** Product can have the **tls** subfeature activated.

M-Box must be activated as a feature in order to start. Similarly, some M-Box programs may need to be activated in order to start.

Activation of product and subfeatures works by requesting a Product Activation Key (PAK) from Isode. Such a request must be accompanied by a Product Activation Request (PAR). Isode messaging products can all be activated using the Messaging Activation Server which is available separately from M-Box at http://www.isode.com/

The Messaging Activation Server manages PAKs, and will put the PAK into *`(ETCDIR)`/activate.dat* .

2.1.4.2 Effects of Product Activation

When M-Box services start up they will check that the M-Box product (aka feature) is activated.

Any use of TLS will always be preceded by a check on TLS activation.

2.1.5 Use of ServPass for password obfuscation

The ServPass facility allows passwords in *`(ETCDIR)`/ms.conf* to be obfuscated to prevent them being read, and then decrypted at the point of use. It is currently necessary to use command line tools when setting up password obfuscation for the *`(ETCDIR)`/ms.conf* file. The steps to follow are described below.

- Create the Service Key for the `isode.mbox` service. You will need to perform this operation as `root` on Unix platforms.

```
[root@diabolo /]# spassmgt set isode.mbox root
Passphrase: 1234567890abcABC
Re-enter: 1234567890abcABC
```

The final parameter in the command line above specifies the userid under which the isode.mbox daemon will run on Unix. The last parameter is omitted on Windows. Choose

a memorable passphrase, which must include a mixture of upper and lower-case letters and digits.

- You then need to obtain a verifier for the service:

```
[root@diabolo /]# spassmgt get isode.mbox
AlsuJCfnoNUo
```

- Edit this verifier into your *(ETCDIR)/ms.conf* file:

```
<servpass:info service="isode.mbox" verifier="AlsuJCfnoNUo"/>
```

- Configure relevant passwords for TLS keys, LDAP connections and Sieve SMTP submission in the *(ETCDIR)/ms.conf* file in the clear. For example:

```
 <tls_key_password servpass:encrypt='true'>My#Secret7_pwd</tls_key_passwor
```

- Encrypt the password(s) in the *(ETCDIR)/ms.conf* file:

```
[root@diabolo /]# spasscrypt -e -s isode.mbox -x /etc/isode/ms.conf
```

where *(ETCDIR)* has the value /etc/isode

2.1.6 Configuring LDAP server and configuring M-Box to use LDAP

This section assumes that you are using the Isode M-Vault directory to hold M-Box configuration.

2.1.6.1 Configuring the LDAP server (M-Vault)

M-Box uses LDAP for user authentication, and passes POP and IMAP credentials to the Directory. Basic LDAP uses Directory Names for authentication, and so there is a need to map from the POP/IMAP login name to a directory name. Isode's approach is to use SASL (Simple Authentication and Security Layer) authentication with LDAP. With SASL, the POP/IMAP login name can be used directly as the authentication ID. This approach is efficient and elegant. It also has an advantage of providing unified SASL configuration for accessing LDAP, IMAP and POP.

In order to make this work, the directory server needs to be configured to map from a SASL authentication ID to a directory name. This section describes one of the possible M-Vault configurations. Full details are given in section 6.3.2 of the *M-Vault Administration Guide*. M-Vault provides a number of options for SASL authentication configuration, and the exact choice will depend on the details of directory and authentication ID approaches.

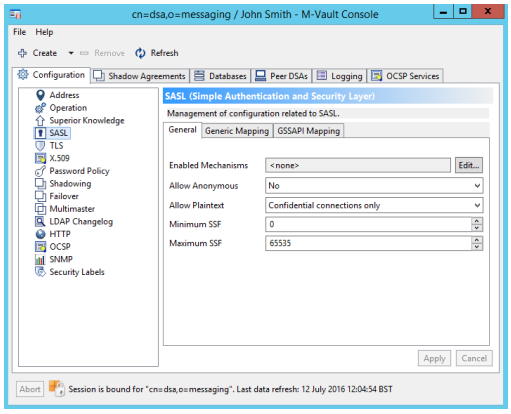
This section explains a simple setup that will be appropriate for many deployments. It assumes that the directory is structured using a domain hierarchy, so that different domains will use a different part of the directory information tree. This is the approach used by Microsoft Active Directory (AD).

Information about mapping from the SASL authentication ID to a directory name is stored in the DSA's own entry. In order for M-Box to share SASL configuration with M-Vault, the M-Box `dsa_config_dn` option will have to point to this entry, as described in Section 2.1.6.2, “Setting up M-Box to use LDAP configuration”.

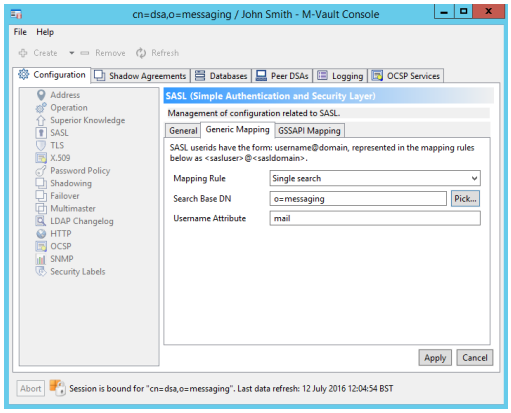
Note: POP/IMAP login names used by M-Box look like email addresses, i.e. they have the form *<user>@<domain>*.

Once you have initialized the M-Vault server using MVC, you must configure SASL in M-Vault.

1. Select the SASL properties in MVC and then select the **General** tab:



2. Make sure that some SASL mechanisms are enabled.
3. Select the **Generic mappings** tab:



   - Keep the mapping rule as **Single search**.
   - Select a suitable **Search Base DN**. All user accounts are located below the selected entry.
   - Optionally, change the **Username attribute**.

2.1.6.2 Setting up M-Box to use LDAP configuration

In order to use M-Box with Isode M-Vault, M-Vault must be configured to use SASL authentication using MVC (see Section 2.1.6.1, “Configuring the LDAP server (M-Vault)”).

The following LDAP options are required in the M-Box configuration file:

- `config_location` must point to the LDAP server that contains M-Box configuration.
- One of `ldap_bind_id` or `ldap_bind_dn` must be specified. `ldap_bind_id` is a SASL username for the Directory account that has permissions to read userPassword attribute and other user attributes from the Directory (M-Box Manager), `ldap_bind_dn` is its

DN. (Note that if you want to use IMA then you must specify `ldap_bind_dn`, as currently IMA doesn't support SASL binds to Directory)

- `ldap_bind_pwd` must contain the M-Box Manager's password.
- If you want to bind using simple bind the `ldap_bind_method` option must contain the value `SIMPLE`. (The value SIMPLE is required if you want to use IMA).
- `dsa_config_dn` option contains DN of the entry containing SASL configuration for M-Box. Typically this configuration is shared between M-Box and M-Vault (see also Section 2.1.6.1, “Configuring the LDAP server (M-Vault)”), so this option would contain the DN of the DSA's own entry. The default value matches the M-Vault's default.
- `ldap_basedn` option contains DN of the M-Box configuration entry. This entry contains the entire M-Box configuration, except for information on Shared Folders and users, it can be located anywhere in Directory Information Tree. It is recommended that the least significant RDN value of the M-Box server entry contains the hostname of the M-Box server.
- `auxprop_plugin` option located below the `sasl` XML element must contain value `ldapdb`.

Example: If M-Box is responsible for domain `myisp.net` (running on `imap.myisp.net`) and M-Vault is running on `ldap.myisp.net`, default M-Vault port 19389, and DSA Manager DN is **cn=DSA Manager, cn=dsa, o=myisp** and its password is `"secret"`, *`(ETCDIR)`/ms.conf* may look like this:

```
<ms_options>
<config_location>ldap://ldap.myisp.net:19389</config_location>
<ldap_bind_method>SIMPLE</ldap_bind_method>
<ldap_bind_dn>cn=DSA Manager, cn=dsa, o=myisp</ldap_bind_dn>

<ldap_bind_pwd>secret</ldap_bind_pwd>
<dsa_config_dn>cn=core,cn=config</dsa_config_dn>
<ldap_basedn>dc=imap, dc=myisp, dc=net, cn=Servers,
 cn=Internet Mailstore, cn=Messaging Configuration,
 ou=MHS, o=MyISP</ldap_basedn>
<sasl>
<auxprop_plugin>ldapdb</auxprop_plugin>
</sasl>
</ms_options>
```

2.1.7 Configuring mail delivery

M-Box includes LMTP server, which by default listens on port 2003 (on Windows) or on */var/run/lmtp* Unix domain socket (on Unix). Messages are delivered to M-Box mailstore over LMTP, so you need to configure the Message Transfer Agent to deliver messages using LMTP. This can be via TCP or a Unix Domain Socket. See the *M-Switch Administration Guide* (section 7.9) for details.

2.1.8 Starting and stopping M-Box

2.1.8.1 M-Box processes

M-Box includes a number of processes. Starting an M-Box installation therefore involves starting the various processes associated with that M-Box. This section summarizes what you must start, and what you may need to start depending on your configuration.

Command line parameters are described in Appendix A, *Command Line Parameters for M-Box Service Applications*.

Each process is introduced using its full name, but subsequent references will use a shortened form. For example, `sieved` will refer to the process `isode.sieved`.

`isode.mseventd`
: This service is used by other M-Box services, so it should be always started.

`isode.lmtpd`
: This service implements LMTP [RFC 2033] server. Sieve processing (see Chapter 6, *Filtering Messages Using Sieve*) happens as a part of message delivery.

`isode.imapd`
: This service implements IMAP4rev1 [RFC 3501] server.

`isode.pop3d`
: This service implements POP3 [RFC 1939] server.

`isode.sieved`
: This service implements ManageSieve [RFC 5804] server.

`isode.msadmind`
: This service implements some administrative functions, such as deletion of users' state on disk.

`isode.ms_syncd`
: This service implements functionality of IMAP/POP3 gateway or IMAP/POP3 migration. It is not enabled by default and thus will not be started.

2.1.8.2 Starting/stopping M-Box on Linux (without systemd)

This section is specific to Linux.

An example startup/shutdown script, */opt/isode/sbin/mbox.sh*, is included in the M-Box package.

The script can start, stop and query all of the M-Box services: `isode.lmtpd`, `isode.imapd`, `isode.pop3d`, `isode.sieved`, `isode.ms_syncd`, `isode.msadmind` and `isode.mseventd`. A symbolic link *mbox* to the script is created in the *rc* directory specific to the platform.

For example, on Red Hat Enterprise Linux 7.0 this will be */etc/init.d/mbox*

- To start M-Box run **/etc/init.d/mbox start**
- To stop it run **/etc/init.d/mbox stop**
- To check which M-Box services are running: **/etc/init.d/mbox status**

2.1.8.3 Starting/stopping M-Box on Linux with systemd

This section is specific to Linux with systemd.

An example startup/shutdown script, */opt/isode/sbin/mbox-systemd.sh*, is included in the M-Box package.

The script can start, stop and query all of the M-Box services: `isode.lmtpd`, `isode.imapd`, `isode.pop3d`, `isode.sieved`, `isode.ms_syncd`, `isode.msadmind` and `isode.mseventd`.

Alternatively you can use systemctl directly:

- To start M-Box run **systemctl start mbox**
- To stop it run **systemctl stop mbox**
- You can also start or stop one or more individual M-Box services by using their corresponding service name without the "isode." prefix, for example **systemctl start imapd mseventd**
- To check which M-Box services are running: **/etc/init.d/mbox-systemd.sh status**
- To check status of a particular M-Box service: **systemctl status imapd**

2.1.8.4 Installing M-Box on Windows

On Windows each M-Box process is installed as a Windows service. These processes run under the Local System account.

Once you have installed an M-Box PAK and created a proper *(ETCDIR)/ms.conf* file, you can use the **Install M-Box services** shortcuts in the **Isode** program group to install M-Box services. Alternatively you can run

```
(SBINDIR)\mbox install
```

from a command line with Administrator's privileges.

2.1.8.5 Starting/stopping M-Box on Windows

By default M-Box services are set to **Automatic Start**, i.e. they will be started automatically at system startup.

It is also possible to start installed M-Box services from the command line using *(SBINDIR)/mbox.exe* utility.

In order to start M-Box run

```
(SBINDIR)\mbox start
```

In order to stop it run

```
(SBINDIR)\mbox stop
```

To check which M-Box services are are installed and running:

```
(SBINDIR)\mbox status
```

In order to enable a particulaer service, use:

```
(SBINDIR)\mbox enable <list-of-services>
```

For example:

```
(SBINDIR)\mbox enable ms_syncd
```

will change `ms_syncd` to **Automatic Start**. (`ms_syncd` is disabled by default)

2.1.9 M-Box users

Users may be added in several ways. See Chapter 3, *M-Box User Management* for more details.

2.2 Server configuration options

The M-Box configuration file *(ETCDIR)/ms.conf* contains an XML fragment. The top level XML element is `ms_options`. Each configuration option is represented as an XML element.

2.2.1 How M-Box applications read configuration

2.2.1.1 Configuration Location

Option location: Directory Configuration Browser page, **General** section.

Description: Specifies where M-Box configuration is stored. The value `ldap:` means that the LDAP server specified in the `ldap_server` and `ldap_port` options will be consulted after reading the configuration file. Alternatively this option can specify one or more LDAP/LDAPS URL. The default value `"local"` means that the whole configuration is stored in the configuration file. The next section describes how M-Box loads its configuration from LDAP.

Default value: `local`

Example: `ldap://publicdir.example.com:389`

XML option name: `config_location`

Parent XML element: top level `<ms_options>`

2.2.1.2 How M-Box applications read configuration from an LDAP Directory

When configuration is stored in LDAP the entry with the DN specified in the `ldap_basedn` option is read first. This is the M-Box configuration entry, which contains the entire M-Box configuration, except for information on SASL Configuration, Shared Folders and Users. Most of the attributes map one-to-one to M-Box options (see the subsequent section for the complete list), with the following exceptions:

The **mboxListeners** attribute is a multivalued attribute, each value is a URL containing

- hostname/IP address and an optional port number;
- path to a Unix domain socket (UNIX only).

The **mboxBackends** and **mboxConnectors** attributes are single-valued attributes, each containing an URL with hostname/IP address and an optional port number.

Once the M-Box configuration entry (as specified in the `ldap_basedn` option) is read, M-Box will try to read the DSA's own entry (as specified in the `dsa_config_dn` option), which contains information necessary to read and manage user information in LDAP. Failure to read this entry (e.g. it is not configured) is not considered fatal.

And finally, M-Box will perform a search for all shared folder roots, located below the DN specified in the **mboxSharedRootsDN** attribute from the M-Box configuration entry. The **mboxSharedRootsDN** attribute contains a DN of an entry, which is the parent of all entries describing shared roots. If this attribute is not specified, the default value `"cn=Shared Folders, <ldap_basedn>"` is assumed, where `<ldap_basedn>` is the DN specified in the `ldap_basedn` option. A warning is logged to standard error if the **mboxSharedRootsDN** entry doesn't exist.

Entries that don't contain the **mboxSharedRootsDN** object class are ignored.

See Chapter 5, *M-Box Shared Folders* for more details on shared folders.

2.2.2 LDAP configuration options

This section describes configuration options that are needed to connect to an LDAP server that contains M-Box server configuration and information about users and shared folders.

These options can be viewed in the Internet Messaging Administration Web Application, but they can't be modified by it.

The options listed in subsections of this section are only used if M-Box is configured to retrieve its configuration from an LDAP server by setting the `config_location` option to an LDAP URL or a special value `ldap:`.

Note: If an LDAP configuration is in force all options specified in Section 2.2, "Server configuration options" are first read from the configuration file. The configured LDAP server is consulted next and any configuration option value found there will override any value specified in the configuration file. Otherwise the value specified in the configuration file is kept. If the configured LDAP server can't be contacted when an M-Box service or application is starting up, the error message is printed to `stderr` and the service/application will refuse to run.

2.2.2.1 Server

Option location: Directory Configuration Browser page, **LDAP options** section.

Description: Specifies the LDAP server to use for retrieving configuration and user related information. This option is ignored if the `config_location` option contains an LDAP URL.

Default value: -None-

Example: `publicdir.example.com`

XML option name: `ldap_server`

Parent XML element: top level `<ms_options>`

2.2.2.2 Port

Option location: Directory Configuration Browser page, **LDAP options** section.

Description: Specifies the LDAP port to use retrieving configuration and user related information. This option is ignored if the `config_location` option contains an LDAP URL.

Default value: 19389

Example: 389

XML option name: `ldap_port`

Parent XML element: top level `<ms_options>`

2.2.2.3 Bind DN

Option location: Directory Configuration Browser page, **LDAP options** section.

Description: Specifies the LDAP Bind DN. This option is ignored if SASL bind is used to authenticate to the LDAP server. See also `ldap_bind_method`.

Default value: -None-

Example: cn=Manager, o=Corp, c=US

XML option name: `ldap_bind_dn`

Parent XML element: top level `<ms_options>`

2.2.2.4 Bind user ID

Option location: Directory Configuration Browser page, **LDAP options** section.

Description: Specifies the LDAP Bind userid. This option is ignored if simple bind is used. See also `ldap_bind_method`.

Default value: -None-

Example: `frank@example.com`

XML option name: `ldap_bind_id`

Parent XML element: top level `<ms_options>`

2.2.2.5 Bind password

Description: Specifies the LDAP Bind password that is used to bind to the LDAP server together with `ldap_bind_dn` (LDAP simple bind) or `ldap_bind_id` (LDAP SASL bind). This value can be obfuscated using the Service Key facility (see Section 2.1.5, “Use of ServPass for password obfuscation”).

Default value: -None-

Example: `supersecret`

XML option name: `ldap_bind_pwd`

Parent XML element: top level `<ms_options>`

Note: This option is not accessible through the Internet Messaging Administrator. The value of this option is used to login to the Internet Messaging Administrator with username Administrator.

2.2.2.6 Bind method

Option location: Directory Configuration Browser page, **LDAP options** section.

Description: Specifies the LDAP authentication method. If this option has the value `SIMPLE`, then LDAP simple bind is used with DN defined by the `ldap_bind_dn` option. Otherwise it contains the name of a SASL mechanism to use in LDAP SASL Bind (and the userid defined by the `ldap_bind_id` option is used).

Default value: `DIGEST-MD5`

Example: `SIMPLE`

XML option name: `ldap_bind_method`

Parent XML element: top level `<ms_options>`

2.2.2.7 M-Box configuration entry DN

Option location: Directory Configuration Browser page, **LDAP options** section.

Description: Specifies the LDAP entry containing M-Box configuration. This option is required if M-Box configuration is stored in LDAP.

Default value: -None-

Example: `cn=server1.example.com,cn=M-Box servers,o=Corp,c=US`

XML option name: `ldap_basedn`

Parent XML element: top level `<ms_options>`

2.2.2.8 Shared roots DN

Option location: Edit Message Store Configuration page, **Users Root Directory** section.

Description: Specifies the parent LDAP entry for all entries describing Shared Folder Roots. If this option is not set, the value of this option is assumed to be **cn=Shared Folders, <*ldap_basedn*>**.

Default value: -None-

Example: cn=Shared Folders, o=Corp, c=US

LDAP attribute name: mboxSharedRootsDN

XML option name: `ldap_shared_folders_dn`

Parent XML element: top level `<ms_options>`

2.2.2.9 DSA own entry DN

Option location: Directory Configuration Browser page, **LDAP options** section.

Description: Specifies the LDAP entry containing DSA SASL configuration. Information in this entry is used to find information about M-Box users in the Directory. If M-Box server configuration is stored in LDAP, this entry would be read together with M-Box server configuration. Reading of SASL configuration from Directory can be disabled by setting this option to the empty string.

Default value: cn=core,cn=config

Example: cn=dsa, c=US

XML option name: `dsa_config_dn`

Parent XML element: top level `<ms_options>`

2.2.2.10 Number of connect retries

Description: Specifies the number of attempts to connect to the LDAP server(s) specified in the `config_location` option.

Default value: 3

Example: 7

XML option name: `ldap_connect_retries`

Parent XML element: top level `<ms_options>`

2.2.2.11 Delay between connect retries

Description: Specifies the delay (in seconds) between two attempts to connect to the LDAP server(s) specified in the `config_location` option.

Default value: 10

Example: 0

XML option name: `ldap_connect_retry_pause`

Parent XML element: top level `<ms_options>`

2.2.3 Virtual servers and user aliases

M-Box servers can listen on multiple interfaces. Each interface may be assigned a different default domain, which can be useful for hosting multiple "virtual" servers on a single machine. The default domain is appended to any unqualified userid.

A default domain for a particular interface can be defined by the `domain_map` XML element, located directly below the top level `ms_options` XML element (**mboxListenDomainMappings** or **mboxRemoteDomainMappings** LDAP attributes). The `domain_map` XML element has two mandatory attributes: `local_addr` and `domain`. The `domain` attribute specifies the default domain for the interface described by the `local_addr` attribute.

If a particular interface has a corresponding `domain_map` XML element, the value of its `domain` attribute is used as the default domain. If there is no corresponding `domain_map` element, the value of the `domain` option is used instead. If the `domain` option is not set, the fully-qualified hostname of the machine running M-Box is used.

2.2.3.1 Default domain

Option location: Edit Message Store Configuration page, **General** section.

Description: This option specifies the default domain value to be appended to an unqualified userid used for authentication and authorization, unless interface specific domain is specified using IP address to domain mapping (see Section 2.2.3, “Virtual servers and user aliases”). The default is None which causes the fully-qualified hostname to be appended to an unqualified userid. It is important to note that if a userid has a domain, the specified domain will always be used instead of the domain option value. See Section 2.2.3, “Virtual servers and user aliases” for detailed description on how this option is used.

Default value: -None-

Example: `example.com`

LDAP attribute name: mt-local-domain-site

XML option name: `domain`

Parent XML element: top level `<ms_options>`

2.2.4 Specifying service listeners

2.2.4.1 Listen URLs

Option location: Edit Message Store Configuration page, **General** section.

Description: This option describes IMAP4, POP3, Manage Sieve, LMTP and mseventd listeners. The option is multivalued, each value is a URL containing

- hostname/IP address and an optional port number;
- path to a Unix domain socket.

Currently the following URL types are recognized.

URL type	Description
imap	An IMAP4 listener.
lmtp	An LMTP listener.
pop	A POP3 listener.
sieve	A Manage Sieve (sieved) listener.
imaps	An IMAPS (IMAP4 over TLS/SSL) listener.
pop3s	An POP3S (POP3 over TLS/SSL) listener.
isode.event	A mseventd listener.
isode.log	A mlogd listener.

Note: Only the hostname/IP address and the port number parts of a URL are considered, the remainder of the URL is ignored.

Note: Multiple URLs of the same URL type are allowed.

In LDAP this option is stored as a single multivalued attribute. In XML file this option is stored as one of more pairs of options, as described by the following table:

URL type	Hostname/IP address option	Default hostname/IP address	Port number option	Default port number	Remarks
imap	`imap_host`	empty (*)	`imap_port`	143	-
lmtp	`lmtp_host`	UNIX: /var/run/lmtp Windows: empty (*)	`lmtp_port`	2003	LMTP port number option is not used when lmtp_host is a Unix domain socket
pop	`pop3_host`	empty (*)	`pop3_port`	110	-
sieve	`sieve_host`	empty (*)	`sieve_port`	4190	-
imaps	`imaps_host`	empty (*)	`imaps_port`	993	Both options are ignored unless enable_imaps option is set to true
pop3s	`pop3s_host`	empty (*)	`pop3s_port`	995	Both options are ignored unless enable_pop3s option is set to true
isode.event	`msevent_host`	UNIX: /var/run/mseventd Windows: 127.0.0.1	`msevent_port`	2004	msevent_port number option is not used when msevent_host is a Unix domain socket
isode.log	`mlogd_host`	UNIX: /var/run/mslogd Windows: 127.0.0.1	`mlogd_port`	2007	mlogd_port number option is not used when mlogd_host is a Unix domain socket

(*) – the empty value means “listen on all available interfaces”.

Default value: -See the table above-

Example (LDAP):

```
lmtp://mail.example.net
```

`imaps://mail.example.net:993`

`isode.event://127.0.0.1:2004/`

LDAP attribute name: mboxListeners

XML option name: -see the description above-

Parent XML element: top level `<ms_options>`

2.2.5 Restricting user access to certain services

M-Box provides the ability to specify which services are accessible to which users. This can be done globally and overridden on per-user basis. For example, it is possible to globally disable POP3 access and allow it on per user basis.

The following 2 options control access to different services.

2.2.5.1 Accessible services

Option location: Edit Message Store Configuration page, **General** section.

Description: Controls which services a user can access by default. It is used to control access to M-Box using POP, IMAP, and the ability to manage message filtering using SIEVE.

The value is a comma separated list of `<`*`service`*`>=<`*`access`*`>` pairs, where `<`*`access`*`>` is one of `allow`, `grant` (alternative name to `allow`, with identical meaning) or `deny`. Currently recognized services are imap, pop and sieve. Services not explicitly listed in this list are controlled by the “Default new service access” option described below.

Default value: empty string

Example: `imap=grant,sieve=allow,pop=deny`

LDAP attribute name: AccessibleServices

XML option name: `accessible_services`

Parent XML element: top level `<ms_options>`

2.2.6 Other general options

2.2.6.1 Runtime user id

Option location: Edit Message Store Configuration page, **General** section.

Description: This option specifies the M-Box runtime (OS) user. On Unix M-Box services start as “root” and then switch to the context of this OS user. In particular this means that all M-Box files are created as the runtime user.

(Windows) This option is ignored on Windows.

Default value: `mbox`

Example: `mailsrv`

LDAP attribute name: uid

XML option name: `ms_user`

Parent XML element: top level `<ms_options>`

2.2.6.2 Management of M-Box services

Option location: This option is not accessible through IMA.

(UNIX) Description: Specifies whether the M-Box services are to be managed internally. When managed internally, each M-Box process forks a child process that would perform the actual work, while the parent process keeps monitoring the child and will restart the child if it terminates abnormally. Turn off the internal manager if the services are to be managed by an external manager such as daemontools.

(Windows) Description: This option is ignored on Windows.

Default value (Linux): `true`

Example: `false`

LDAP attribute name: -None-

XML option name: `managed_services`

Parent XML element: top level `<ms_options>`

2.2.6.3 Users root directory

Option location: Edit Message Store Configuration page, **General** section.

Description: Specifies the default location for user mail. Each user's mail and other associated state will be stored in a subdirectory of this directory, named after the user. This value can be overridden on per-user basis by setting the **mboxRootUserDir** attribute in the user's entry.

(UNIX) Default value: `/var/isode/ms/user`

(Windows) Default value: `<`*`drive:`*`>\Isode\ms\user`

Example: `/var/imap/mail`

LDAP attribute name: mboxRootUserDir

XML option name: `userdir`

Parent XML element: top level `<ms_options>`

2.2.6.4 Shared folders root directory

Option location: Edit Message Store Configuration page, **General** section.

Description: Specifies the default filesystem location for Shared Folders. Mail for a Shared Folder named *SFolder* will be located in directory `<`*`shared_root_dir`*`>/SFolder`.

(UNIX) Default value: `/var/isode/ms/shared`

(Windows) Default value: `<`*`drive:`*`>\Isode\ms\shared`

Example: `/var/imap/shared_folders`

LDAP attribute name: mboxRootSharedDir

XML option name: `shared_root_dir`

Parent XML element: top level `<ms_options>`

2.2.6.5 Administrative user name

Option location: Edit Message Store Configuration page, **General** section.

Description: This option specifies username of the IMAP user who has administrative privileges. If SASL authentication is used in POP/IMAP, then a client can authenticate as the administrative user, but request access to mail owned by any other user.

> **Note:** This option is required for **msadm expire_mail** command.

Default value: -None-

Example: `administrator`

LDAP attribute name: isodeMboxAdminUser

XML option name: `admin_user`

Parent XML element: top level `<ms_options>`

2.2.7 Quota

M-Box supports quota limits on user mailboxes. Non-zero quota setting can restrict the total size of all messages in users' personal mailboxes. The following 2 options control global quota behaviour.

2.2.7.1 Mailbox size quota

Option location: Edit Message Store Configuration page, **Quota** section.

Description: Specifies the default per-user quota limit on total size of all messages in all user's mailboxes. The value applies to all users who don't have an explicit quota limit set on the **Edit User** page. This value is in Kilobytes. The default value is `0`, which means that there is no quota limit.

Default value: `0`

Example: `102400`

LDAP attribute name: mboxMessageSizeQuota

XML option name: `quota_limit`

Parent XML element: top level `<ms_options>`

2.2.7.2 Over quota is temp error

Option location: Edit Message Store Configuration page, **Quota** section.

Description: Specifies if `overquota` condition causes lmtpd to return the `452` temporary error code. If this option is false, the `552` permanent error code is sent.

Default value: `true`

Example: `false`

LDAP attribute name: mboxOverQuotaIsTempError

XML option name: `over_quota_temp_error`

Parent XML element: top level `<ms_options>`

2.2.8 POP3 and IMAP4 specific options

2.2.8.1 Clear text login disabled

Option location: Edit Message Store Configuration page, **POP3 and IMAP specific options** section.

Description: If this option is set to `true`, it disables all commands that send password in the clear to the server. This includes the IMAP LOGIN command, SASL PLAIN and LOGIN authentication mechanism and POP3 USER/PASS commands.

> **Note:** If TLS encryption is active then SASL PLAIN mechanism is allowed even if this option is set to `true`.

Default value: `true`

Example: `false`

LDAP attribute name: mboxCleartextLoginDisabled

XML option name: `login_disabled`

Parent XML element: top level `<ms_options>`

2.2.8.2 Authenticate against Directory that stores hashed passwords

Option location: Edit Message Store Configuration page, **POP3 and IMAP specific options** section.

Description: If this option is set to `true`, it enables special password verification mode which uses LDAP bind to verify users' passwords. This can be used with a Directory server that doesn't allow retrieving of the **userPassword** attribute, doesn't store it, or stores it in a hashed form. This option limits which SASL authentication mechanisms can be used by IMAP/POP clients, typically it limits the list of SASL mechanisms to SASL PLAIN.

Default value: `false`

Example: `true`

LDAP attribute name: -None-

XML option name: `use_hashed_passwords`

Parent XML element: top level `<ms_options>`

2.2.8.3 POP timeout

Option location: Edit Message Store Configuration page, **POP3 and IMAP specific options** section.

Description: Idle timeout for POP3 sessions in minutes. This value can’t be less than 5 minutes.

Default value: `10`

Example: `7`

LDAP attribute name: mboxPopTimeout

XML option name: `pop_timeout`

Parent XML element: top level `<ms_options>`

2.2.8.4 IMAP timeout

Option location: Edit Message Store Configuration page, **POP3 and IMAP specific options** section.

Description: Idle timeout for IMAP sessions in minutes. This value can't be less than 30 minutes.

Default value: `30`

Example: `60`

LDAP attribute name: mboxImapTimeout

XML option name: `imap_timeout`

Parent XML element: top level `<ms_options>`

2.2.8.5 TCP keep alive

Option location: Edit Message Store Configuration page, **POP3 and IMAP specific options** section.

Description: This option controls if the TCP Keep-Alive option is set on IMAP and IMAPS connections. The value is in minutes. If the server hasn't received any requests from the client for the specified number of minutes, a special TCP packet will be sent by the server to prompt the client to return another TCP packet to acknowledge receipt of the server's packet. This helps to keep the connection alive in presence of NATs and firewalls, it also helps to detect broken TCP connections earlier. The value `0` means that this option is disabled.

Note: On some platforms it is not possible to control frequency of TCP Keep-Alive packets. On such platforms any non `0` value will enable OS specific Keep-Alive frequency.

Default value: `0`

Example: `5`

LDAP attribute name: -None-

XML option name: `tcp_keepalive`

Parent XML element: top level `<ms_options>`

2.2.8.6 IMAP keep alive

Option location: Edit Message Store Configuration page, **POP3 and IMAP specific options** section.

Description: This option controls if imapd sends untagged IMAP responses to keep IMAP clients connected while imapd is executing a long running IMAP command. Currently this only affects the **COPY** and **UID COPY** commands. The value is in minutes. The value `0` means that this option is disabled.

Default value: `0`

Example: `5`

LDAP attribute name: -None-

XML option name: `imap_keepalive`

Parent XML element: top level `<ms_options>`

2.2.8.7 IMAP login alert

Option location: Edit Message Store Configuration page, **POP3 and IMAP specific options** section.

Description: This option specifies human readable text sent to a successfully logged in IMAP user using IMAP ALERT response code. Such text is typically shown in a pop-up dialog in IMAP clients.

Default value: -None-

Example: `System upgrade on April 8th`

LDAP attribute name: isodeMboxLoginAlert

XML option name: `login_alert`

Parent XML element: top level `<ms_options>`

2.2.8.8 Default mail expiration policy

Option location: Edit Message Store Configuration page, **POP3 and IMAP specific options** section.

Description: This option specifies the default mail expiration policy that would be applied by **msadm expire_mail** to users who don't have any explicitly set mail expiration policy. The value has the following syntax: *<mailbox-name>*$*<mail-retention-period>*, where *<mailbox-name>* is name of an IMAP mailbox where mail would be expired (typically *INBOX*) and *<mail-retention-period>* is the mail retention period in seconds. For example, if this option contains `"INBOX$259200"`, this means that any messages in *INBOX* older than 259200 seconds (3 days) will be automatically expired.

Default value: -None-

Example: `INBOX$259200`

LDAP attribute name: isodeMboxMailExpirationPolicy

XML option name: `expiration_policy`

Parent XML element: top level `<ms_options>`

2.2.8.9 msadmin IMAP client timeout

Option location: Edit Message Store Configuration page, **POP3 and IMAP specific options** section.

Description: msadm expire_mail IMAP timeout in seconds. If no response is received from IMAP server within the timeout, the IMAP connection is considered to be dead and the error is reported to the client.

Use the value `-1` to disable the client timeout.

Default value: `10`

Example: `120`

LDAP attribute name: -None-

XML option name: `client_timeout`

Parent XML element: top level <ms_options>M-Box Gateway and email migration

See Chapter 7, *M-Box POP to IMAP Gateway* for a description of M-Box in Gateway mode.

2.2.9 Gateway and migration mode specific options

2.2.9.1 Enable gateway mode

Option location: Edit Message Store Configuration page, **Email Synchronization** section.

Description: Specifies if imapd is operating as an IMAP gateway to one or more back-end POP3/IMAP servers. If this option is `false`, all other options described in Section 2.2.9, "Gateway and migration mode specific options" are ignored. Setting this option to `true` when `auto_migrate` option is also set to `true` is an error and would cause ms_syncd startup failure.

Default value: `false`

Example: `true`

LDAP attribute name: mboxGatewayMode

XML option name: `gateway_mode`

Parent XML element: top level `<ms_options>`

2.2.9.2 Backend POP3/IMAP

Option location: Edit Message Store Configuration page, **Email Synchronization** section.

Description: Specifies the default back-end POP3/IMAP hostname/IP address, port number and protocol to connect to. In LDAP this option is stored as a POP/POPS/IMAP/IMAPS URL for both gateway and migration modes. In the XML file this value is stored as three separate options. The back-end POP3/POPS/IMAP/IMAPS port number can be omitted, in which case it defaults to 110/995/143/993 respectively. The back-end protocol can also be omitted, in which case it defaults to `pop`.

The back-end protocol can be one of `pop`, `pop3` (same as `pop`), `pops` (POP3 over TLS on a separate port), `pop3s` (same as `pops`), `imap`, `imap4` (same as `imap`) or `imaps` (IMAP over TLS on a separate port). Unless configured otherwise (see `suppress_client_tls` option and below), `isode.ms_syncd` will always try to use STARTTLS/STLS to provide data confidentiality. If this fails, the connection to the POP/IMAP backend will not be protected from eavesdropping.

M-Box allows overriding this value on per-user basis. This information is stored in either LDAP, or in the XML user database (located in *`(ETCDIR)`*`/ms_sync.xml`), which contains mappings from frontend userids to backend POP3/IMAP userids and/or information about POP3/IMAP back-end to be used for the frontend user. The top level XML element in the database is `xmldb`. Each element below it must be a "user" XML element containing information about user's master backend. The "user" XML element has several attributes:

- Required `frontend_id` attribute contains the userid as accepted by the front-end. It might also contain a value in the format `@<`*`domain`*`>` (e.g. `@example.net`), in which case all users in the specified domain will be controlled by this record in the `xmldb`.
- An optional `backend_id` attribute contains the backend userid that should be used to talk to the POP3/IMAP back-end. If this attribute is not specified, the `backend_id` is assumed to be the same as the `frontend_id`. If `frontend_id` contains a value `@<`*`domain`*`>`, this attribute might contain a value in the format `@<`*`anotherdomain`*`>`. Such value will cause `ms_syncd` to substitute domain in the frontend username with the

domain specified in the `backend_id` attribute, before passing the username to the backend server.

- An optional `host` attribute specifies hostname or IP address for the POP3/IMAP back-end to be used for this user. If this attribute is not specified, the global POP3/IMAP back-end hostname is used.
- An optional `port` attribute specifies the port number for the POP3/IMAP backend to be used for this user. If this attribute is not specified, the global POP3/IMAP backend port number is used. If the global backend port is not specified (or set to `0`), then the protocol specific default is used.
- An optional `protocol` attribute specifies the protocol being used to access this backend. If the protocol is not specified, the global default is used.
- An optional `password` attribute specifies password to be used to authenticate to the backend. For a master backend (i.e. the backend used to authenticate frontend user) the password must match the password used on the frontend.
- An optional `mailbox` attribute specifies the frontend mailbox (for POP3 backends) or the frontend mailbox prefix (for IMAP backends) to be used for synchronizing messages. If this attribute is missing, then the value `INBOX` is used for POP3 backends (i.e. messages will be downloaded to the *INBOX* mailbox) and the empty string is used for IMAP backends (i.e. any mailbox on the backend is converted to a mailbox with the same name on the frontend).
- An optional `login_delay` attribute allows to control frequency of mail synchronization with POP3/IMAP backends. If this attribute is not specified, the global default is used. The value is in seconds. Note that ms_syncd obeys mail synchronization frequency advertised by POP3 backends (using the LOGIN-DELAY POP3 capability). So if the specified (or global) value is lower than the advertised limit, then it will be ignored.
- An optional `starttls` attribute allows to control use of IMAP STARTTLS/POP3 STLS when talking to backends. The global default is used, if this attribute is not specified. Use this attribute if you want to work around bugs in STARTTLS/STLS in backends.

Additional backends can be specified using the subordinate <backend> XML elements. Such elements contain the same attributes as used in the `<user>` element, except that the `frontend_id` is not allowed in them.

Example *`(ETCDIR)`/ms_sync.xml* file:

```
<xmldb>
  <user frontend_id='user1' backend_id='user1@example.ca'
host='pop3.example.ca' port='110'>
    <backend backend_id='second-account' host='127.0.0.1'
protocol='pop' password='123' mailbox='Other'/>
  </user>
</xmldb>
```

Default value: None.

Example (LDAP): `pop://pop.example.net:3110/`

LDAP attribute name: mboxBackends (multivalued) and **isodeMboxMasterBackend** (specifies the master backend)

XML option name: `ms_syncd_protocol`, `ms_syncd_host` and `ms_syncd_port`

Parent XML element: top level `<ms_options>`

2.2.9.3 Polling interval

Option location: Edit Message Store Configuration page, **Email Synchronization** section.

Description: Specifies the polling interval to the back-end POP3 server to check for new mail. The value is in seconds. The special value `-1` means that no polling would happen, so mail synchronization will only happen on initial connection and upon `msstat -e ms_sync:<user>` events. This option only affects M-Box running in Gateway mode.

Default value: `300`

Example: `600`

LDAP attribute name: **mboxSyncInterval**

XML option name: `ms_sync_interval`

Parent XML element: top level `<ms_options>`

2.2.9.4 Enable mail migration mode

Option location: Edit Message Store Configuration page, **Email Synchronization** section.

Description: Specifies if imapd/pop3d is operating in email migration mode from a POP3/IMAP back-end. Setting this option to `true` when `gateway_mode` option is also set to `true` is an error and would cause ms_syncd startup failure.

Default value: `false`

Example: `true`

LDAP attribute name: **mboxAutoMigrate**

XML option name: `auto_migrate`

Parent XML element: top level `<ms_options>`

2.2.9.5 Enable synchronization mode

Option location: Edit Message Store Configuration page, **Email Synchronization** section.

Description: Specifies if imapd should try to synchronize mail from one or more POP3/IMAP backends. Setting this option to true is similar to setting `gateway_mode` option to `true`, except that the local (stored on the frontend) password is used for user authentication. This option can be set to `true` even if one of `gateway_mode` or `auto_migrate` is set to `true`.

Default value: `false`

Example: `true`

LDAP attribute name: -None-

XML option name: `sync_mode`

Parent XML element: top level `<ms_options>`

2.2.9.6 Suppress use of IMAP STARTTLS/POP3 STLS when talking to Gateway backends

Option location: Edit Message Store Configuration page, **Email Synchronization** section.

Description: This option allows to disable use of STLS (in POP3)/STARTTLS (in IMAP) when talking to the POP3/IMAP back-end. It can be useful to workaround bugs in

STLS/STARTTLS implementations. When this option is `false`, STLS/STARTTLS would be negotiated automatically, if advertised by the backend server.

Default value: `false`

Example: `true`

LDAP attribute name: -None-

XML option name: `suppress_client_tls`

Parent XML element: top level `<ms_options>`

2.2.9.7 Fast login to backend

Option location: Edit Message Store Configuration page, **Email Synchronization** section.

Description: When this option is `false`, M-Box Gateway service (ms_syncd) won't let a user to log into M-Box frontend (imapd/pop3d) until mail synchronization or migration is complete. When this option is set to `true`, ms_syncd allows users to log into imapd/pop3d, as soon as it is able to successfully authenticate to the master backend. Setting this option to true prevents IMAP/POP3 clients from timing out while waiting for IMAP/POP3 authentication commands to complete.

Default value: `false`

Example: `true`

LDAP attribute name: -None-

XML option name: `backend_fast_login`

Parent XML element: top level `<ms_options>`

2.2.9.8 Backend fetch size

Option location: Edit Message Store Configuration page, **Email Synchronization** section.

Description: This option controls how many messages are downloaded from the IMAP backend in any IMAP operation. This is an advanced option that can affect performance of ms_syncd: bigger values will generate slightly less traffic to the backend IMAP server, but can result in higher utilitization of server resources, such as allocated memory.

Default value: `100`

Example: `1000000`

LDAP attribute name: -None-

XML option name: `backend_fetch_set_size`

Parent XML element: top level `<ms_options>`

2.2.9.9 Ignore frontend deletes

Option location: Edit Message Store Configuration page, **Email Synchronization** section.

Description: This option controls if messages deleted on the M-Box Gateway backend (using IMAP or POP3) will be deleted from the corresponding POP3 backend. Set this option to `true` if you want to periodically expire mail on the frontend, without deleting it from POP3 backends. This option is not used for IMAP backends.

Default value: `false`

Example: `true`

LDAP attribute name: -None-

XML option name: `ignore_frontend_deletes`

Parent XML element: top level `<ms_options>`

2.2.9.10 SMTP injection

Option location: Edit Message Store Configuration page, **Email Synchronization** section.

Description: This option controls how messages fetched from POP3/IMAP backends are delivered to M-Box. When this option is `false`, messages are injected directly to mailstore. When this option is `true`, messages are submitted using SMTP, located using `submit_host` and `submit_port` options. The latter allows you to force antispam processing or content conversion on fetched messages, before they are injected to the mailstore.

Default value: `false`

Example: `true`

LDAP attribute name: -None-

XML option name: `smtp_injection`

Parent XML element: top level `<ms_options>`

2.2.9.11 MAIL FROM sender for SMTP injection

Option location: Edit Message Store Configuration page, **Email Synchronization** section.

Description: This option controls the SMTP MAIL FROM sender for messages injected using SMTP. This option is only used when `smtp_injection` is set to `true`.

Default value: `mssync@<`*`domain`*`>`

Example: `gateway@example.com`

LDAP attribute name: -None-

XML option name: `smtp_injection_sender`

Parent XML element: top level `<ms_options>`

2.2.9.12 Message processing with Content Checking and Conversion Protocol (CCCP) before injection to mailstore

Option location: Edit Message Store Configuration page, **SIEVE mail filtering engine** section.

Description: `cccp_host` and `cccp_port` XML options specify the hostname/IP address and port number of the CCCP server to be used for converting messages downloaded by ms_syncd before they are injected into mailstore. CCCP conversion is only performed if `cccp_host` option is set.

Default value: -None- for cccp_host, `18003` for cccp_port

Example (LDAP):

LDAP attribute name: -None-

XML option name: `cccp_host` and `cccp_port`

Parent XML element: top level `<ms_options>`

2.2.9.13 Only fetch last N messages

Option location: Edit Message Store Configuration page, **Email Synchronization** section.

Description: This option controls the maximum number of backend messages that will appear in the frontend mailbox. As new messages get delivered into the backend mailstore, they will disappear from the frontend mailstore. The default value 0 means that all messages will be downloaded.

Default value: `0`

Example: `50`

LDAP attribute name: isodeMboxFetchLastN

XML option name: `fetch_last_n`

Parent XML element: top level `<ms_options>`

2.2.9.14 Only synchronise INBOX

Option location: Edit Message Store Configuration page, **Email Synchronization** section.

Description: When this option is `true`, M-Box Gateway service (ms_syncd) would only synchronize INBOX mailbox from any IMAP backend. Otherwise all backend mailboxes are synchronized.

Default value: `false`

Example: `true`

LDAP attribute name: -None-

XML option name: `sync_inbox_only`

Parent XML element: top level `<ms_options>`

2.2.10 LMTP delivery

This section describes additional options controlling LMTP delivery.

2.2.10.1 Duplicate suppression

Option location: Edit Message Store Configuration page, **LMTP delivery** section.

Description: Specifies if the LMTP server should perform duplicate message suppression. A message is considered to be a duplicate if it has the same RFC 2822 Message-Id header field and destined for the same mailbox as a previously received message.

Note: Duplicate suppression slows down delivery speed, as the LMTP server has to record some information about every message it receives.

Default value: `false`

Example: `true`

LDAP attribute name: mboxDuplicateSuppression

XML option name: `duplicate_suppression`

Parent XML element: top level `<ms_options>`

2.2.10.2 Autocreate mailboxes

Option location: Edit Message Store Configuration page, **LMTP delivery** section.

Description: Automatically create user´s INBOX when mail is delivered via LMTP. The default is `true`.

A user INBOX is always autocreated on successful login into IMAP or POP3 server.

Default value: `true`

Example: `false`

LDAP attribute name: mboxLmtpAutocreate

XML option name: `lmtp_autocreate`

Parent XML element: top level `<ms_options>`

2.2.10.3 LMTP database

Option location: Edit Message Store Configuration page, **SIEVE mail filtering engine** section.

Description: This option specifies the location of the SIEVE message tracking database, which is used to prevent generation of duplicated vacation notices, as well as to suppress email duplicates (see also Section 2.2.10.1, “Duplicate suppression”).

(UNIX) Default value: `/var/isode/ms/msg_track.db`

(Windows) Default value: `<drive:>\Isode\ms\ msg_track.db`

Example: `/var/imap/msg_track.db`

LDAP attribute name: mboxMessageTrackDatabase

XML option name: `msgtrack_db`

Parent XML element: top level `<ms_options>`

2.2.10.4 Duplicate history

Option location: Edit Message Store Configuration page, **LMTP Delivery** section.

Description: This option specifies how often records from the LMTP database (Section 2.2.10.3, “LMTP database”) will expire. This value is in days. If this value is less than 0, it is assumed to be 1.

Default value: `7`

Example: `14`

LDAP attribute name: mboxDuplicateHistory

XML option name: `duplicate_history`

Parent XML element: top level `<ms_options>`

2.2.10.5 Maximum LMTP message buffer size

Option location: Edit Message Store Configuration page, **LMTP Delivery** section.

Description: This option specifies the maximum memory buffer size (in Kb) used by lmtpd for storing received messages before delivery. If a received message is bigger than this limit, then every time the full buffer is accumulated, it will be written to disk. Increasing this value can improve performance for delivering big messages, but would increase memory usage.

Default value: `2048` (2 Mb)

Example: `64` (64 Kb)

LDAP attribute name: -None-

XML option name: `lmtp_data_size`

Parent XML element: top level `<ms_options>`

2.2.11 SIEVE mail filtering settings

See Chapter 6, *Filtering Messages Using Sieve* for the description of SIEVE and how to use it.

2.2.11.1 Enable SIEVE

Option location: Edit Message Store Configuration page, **SIEVE mail filtering engine** section.

Description: This option controls if SIEVE mail filtering language is used by the LMTP server to process messages before delivering them to the user's mailbox. If you are not using SIEVE you should change this option to `false`, as this will slightly improve mail delivery speed.

Default value: `true`

Example: `false`

LDAP attribute name: mboxEnableSieve

XML option name: `enable_sieve`

Parent XML element: top level `<ms_options>`

2.2.11.2 Maximum script size

Option location: Edit Message Store Configuration page, **SIEVE mail filtering engine** section.

Description: This option specifies the maximum allowed SIEVE script size that can be uploaded using the ManageSieve Protocol. This value is in Kilobytes. This option exists in order to prevent abuse by ManageSieve clients.

Default value: `32`

Example: `64`

LDAP attribute name: mboxMaxSieveScriptSize

XML option name: `sieve_maxscriptsize`

Parent XML element: top level `<ms_options>`

2.2.11.3 Maximum number of scripts

Option location: Edit Message Store Configuration page, **SIEVE mail filtering engine** section.

Description: This option specifies the maximum allowed number of different SIEVE scripts that can be uploaded using the ManageSieve Protocol. This option exists in order to prevent abuse by ManageSieve clients.

Default value: `5`

Example: `10`

LDAP attribute name: mboxMaxSieveScriptCount

XML option name: `sieve_maxscripts`

Parent XML element: top level `<ms_options>`

2.2.11.4 Submission servers

Option location: Edit Message Store Configuration page, **SIEVE mail filtering engine** section.

Description: Specifies the hostname/IP address and port number of the ESMTP server to be used to submit messages generated by SIEVE engine (e.g. vacation replies, rejects and redirects). Currently a single submit server is supported. In LDAP this option is stored as a SMTP URL. In XML file this value is stored as two separate options. The submission port number can be omitted, in which case it defaults to 587.

Default value: `smtp://localhost:587`

Example (LDAP): `smtp://out-smtp.example.net:25/`

LDAP attribute name: mboxConnectors

XML option name: `submit_host` and `submit_port`

Parent XML element: top level `<ms_options>`

2.2.11.5 Global Sieve script

Option location: Edit Message Store Configuration page, **SIEVE mail filtering engine** section.

Description: This option can define any Sieve script that would be executed on every email message before attempting mail delivery in lmtpd. This option may be useful for generic antispam processing or for mail archiving.

Default value: -None-

Example:

LDAP attribute name: mboxActiveScript

XML option name: `global_sieve_script`

Parent XML element: top level `<ms_options>`

2.2.11.6 Maximum allowed number of Sieve redirect actions per script

Option location: Edit Message Store Configuration page, **SIEVE mail filtering engine** section.

Description: This option specifies the maximum number of SIEVE “redirect” actions that can be executed by a SIEVE script. If a SIEVE script wants to execute more than the specified number of redirect actions, then all redirect actions after the specified number are ignored and an error message is logged for each. This option limits the user's ability to cause damage to a mail system by limiting the number of recipients a message can be redirected to.

Default value: 5

Example: 1

LDAP attribute name: -None-

XML option name: `sieve_maxredirects`

Parent XML element: top level `<ms_options>`

2.2.11.7 Default SMTP MAIL FROM sender for redirected messages

Option location: Edit Message Store Configuration page, **SIEVE mail filtering engine** section.

Description: This option specifies an email address that will be used as the SMTP MAIL FROM address of any message generated by the SIEVE redirect action. See description of the `sieve_redirect_mode` option.

Default value: -None-

Example: `bounce-owner@example.com`

LDAP attribute name: -None-

XML option name: `sieve_redirect_sender`

Parent XML element: top level `<ms_options>`

2.2.11.8 Controlling SMTP MAIL FROM sender for redirected messages

Option location: Edit Message Store Configuration page, **SIEVE mail filtering engine** section.

Description: This option controls how SMTP `MAIL FROM` value for messages being redirected (with the SIEVE redirect action) is selected by lmtpd.

Value `0` means that the value of the `sieve_redirect_sender` option is used as is. If the `sieve_redirect_sender` option is not defined, then `sieve@<defaultdomain>` is going to be used instead.

Value 1 means that the value of the `sieve_redirect_sender` option is used, but the owner of the script is added as a subaddress. For example, if the owner of the SIEVE script is `steve` and the `sieve_redirect_sender` option has the value `bounces@example.net`, then the `MAIL FROM` is going to be `bounces+steve@example.net`. If the `sieve_redirect_sender` option is not defined, then `sieve@<defaultdomain>` is going to be used as the base address instead.

Value 2 means that the original `MAIL FROM` sender of the message being redirected would be used as the `MAIL FROM`. This option has the advantage/disadvantage that if the redirected message bounces, then the original sender gets notified that the message was not delivered.

If a message only gets redirected to a single email address, this might be OK, but in some cases this might disclose too much information to the sender about the recipient's mail system.

Value 3 is the same as value 2, but subaddress `sieve` is added to the original sender's email address. E.g. If the original message is from `martin@example.org`, then the redirected message will use `martin+sieve@example.org` as the `MAIL FROM`.

Default value: `3`

Example: `1`

LDAP attribute name: -None-

XML option name: `sieve_redirect_mode`

Parent XML element: top level `<ms_options>`

2.2.12 TLS configuration options

The following options appear only in the Advanced mode. They control if IMAPS, IMAP STARTTLS, POP3S and POP3 STLS are available. Note that the default configuration has no certificates, nor anonymous ciphers, and in this situation M-Box will refuse to start if IMAPS and/or POP3S services are required.

2.2.12.1 Support for IMAPS (IMAP over TLS/SSL)

Description: This boolean option controls if IMAPS (IMAP over TLS) is enabled. This option has no corresponding LDAP attribute. Presence of an “imaps” URL in the **mboxListeners** LDAP attribute enables IMAPS.

Default value: `false`

Example: `true`

LDAP attribute name: -None-

XML option name: `enable_imaps`

Parent XML element: top level `<ms_options>`

2.2.12.2 Support for POP3S (POP3 over TLS/SSL)

Description: This boolean option controls if POP3S (POP3 over TLS) is enabled. This option has no corresponding LDAP attribute. Presence of a “pop3s” URL in the **mboxListeners** LDAP attribute enables POP3S.

Default value: `false`

Example: `true`

LDAP attribute name: -None-

XML option name: `enable_pop3s`

Parent XML element: top level `<ms_options>`

2.2.12.3 Cipher list

Option location: Edit Message Store Configuration page, **TLS** section.

Description: Specifies the list of space (or colon) separated TLS ciphers that the server is allowed to use. See Appendix D, *TLS Cipher List Formats* for more details.

Default value: `DEFAULT`

Example: `DHE-RSA-AES256-SHA DHE-DSS-AES256-SHA AES256-SHA`

LDAP attribute name: mboxTlsCipherList

XML option name: `tls_cipher_list`

Parent XML element: top level `<ms_options>`

2.2.12.4 CA file (PEM)

Option location: Edit Message Store Configuration page, **TLS** section.

Description: Specifies path to a PEM file containing one or more CA certificates used for clients' verification. Unless `tls_ca_path` is also specified, the CA certificate is also the CA that signed the server certificate contained in `tls_cert_file`. The CA certificates from this file are loaded on imapd/pop3d startup. On Windows this option can also contain a 'certstore:' URI (e.g. '`certstore:Root`'), so that the corresponding CA certificates can be retrieved from the Windows Certificate Store. If this option is not set, this might result in the server's inability to verify client certificates, however TLS will still be able to provide data encryption. See also Section 2.2.12.6, “Certificate file”.

Default value: -None-

Example: `/etc/isode/mbox-tls/ca_certificate.pem`

LDAP attribute name: mboxTlsCaFile

XML option name: `tls_ca_file`

Parent XML element: top level `<ms_options>`

2.2.12.5 CA path

Option location: Edit Message Store Configuration page, **TLS** section.

Description: Specifies a directory where multiple PEM files containing "trusted" CA certificates can be located. Should the server require that the client authenticates using a certificate, any client certificate will be checked to ensure that there is an unbroken chain of trust between the client's certificate and one of the "trusted" certificates. The trusted certificates are only read when they are needed during verification process.

Default value: -None-

Example: `/etc/isode/mbox-tls/extra-ca`

LDAP attribute name: mboxTlsCaPath

XML option name: `tls_ca_path`

Parent XML element: top level `<ms_options>`

2.2.12.6 Certificate file

Option location: Edit Message Store Configuration page, **TLS** section.

Description: Specifies the full path to a file containing the server's own certificate. This certificate will be sent by the server to any client that wishes to confirm the server's identity when negotiating secure communication. The certificate format can be either PEM, DER, or PKCS#12. The file extension has to match the format, i.e. *.pem*, *.der*, or *.p12*. The file extension *.crt* is also accepted here, in which case the file must contain a PEM certificate. Note that the *.p12* file may also contain CA certificates. On Windows this option can also

contain a 'certstore:' URI that points to a certificate in the Windows Certificate Store (e.g. 'certstore:My:sha1:09cfea1e5e5f2302bbd77f91c18ffb5e66135a01'). Without this option specified, TLS services will only be offered using anonymous cipher suites, which are disabled by default. Anonymous cipher suites are typically unsupported by client software, and therefore should be used with care.

Default value: -None-

Example: `/etc/isode/mbox-tls/mbox_certificate.pem`

LDAP attribute name: mboxTlsCertFile

XML option name: `tls_cert_file`

Parent XML element: top level `<ms_options>`

2.2.12.7 Key file

Option location: Edit Message Store Configuration page, **TLS** section.

Description: Specifies the full path to a PEM/DER file containing the private key belonging to the server certificate. The key file format is determined from the file extension (*.pem*, *.der*, *.crt* (PEM), *.key* (PEM)). If the file extension is not recognized, the file is assumed to be in the same format as the certificate file (`tls_cert_file`). If this option is not set, the value of the `tls_cert_file` is used. This value is not used when `tls_cert_file` option points to a PKCS 12 (*.p12*) file.

Default value: -None-

Example: `/etc/isode/mbox-tls/mbox-key.pem`

LDAP attribute name: mboxTlsKeyFile

XML option name: `tls_key_file`

Parent XML element: top level `<ms_options>`

2.2.12.8 Key password

Option location: Edit Message Store Configuration page, **TLS** section.

Description: Specifies the password used to decrypt the server's private key. This option is empty by default, which means that the private key is not protected by any password. This value can be obfuscated using the Service Key facility (see Section 2.1.5, “Use of ServPass for password obfuscation”).

Default value: -empty string-

Example: `SuperS0cret-Password`

LDAP attribute name: mboxTlsKeyPassword

XML option name: `tls_key_password`

Parent XML element: top level `<ms_options>`

2.2.12.9 Require client certificate

Option location: Edit Message Store Configuration page, **TLS** section.

Description: This boolean option specifies if a client certificate is required for TLS negotiation to succeed.

Default value: `false`

Example: `true`

LDAP attribute name: mboxTlsRequireClientCert

XML option name: `tls_require_client_cert`

Parent XML element: top level `<ms_options>`

2.2.12.10 Verify depth

Option location: Edit Message Store Configuration page, **TLS** section.

Description: Specifies the maximum depth of a certificate verification chain.

Default value: `5`

Example: `7`

LDAP attribute name: mboxTlsVerifyDepth

XML option name: `tls_verify_depth`

Parent XML element: top level `<ms_options>`

2.2.12.11 Allow use of certificates that can't be verified due to unrecognized CA certificate

Option location: Edit Message Store Configuration page, **TLS** section.

Description: If this option is set to `true`, TLS channels may still be established with peers which fail certificate verifications. Such certificates will not be considered for authentication (i.e. by SASL EXTERNAL).

Default value: `true`

Example: `false`

LDAP attribute name: -None-

XML option name: `tls_force_verify`

Parent XML element: top level `<ms_options>`

2.2.12.12 Use of TLS by mseventd clients

Option location: Edit Message Store Configuration page, **TLS** section.

Description: If this option is set to `true`, mseventd clients (such as msstat, imapd, etc.) will attempt to protect communication with mseventd with TLS. If TLS is unavailable (e.g. no TLS certificate is configured and ADH TLS cipher is not enabled) and this option is `true`, then mseventd clients would fail to start.

Default value: `false`

Example: `true`

LDAP attribute name: -None-

XML option name: `msevent_tls`

Parent XML element: top level `<ms_options>`

2.2.12.13 Authentication using TLS when talking to mseventd

Option location: Edit Message Store Configuration page, **TLS** section.

Description: If this option is set to `true`, mseventd clients (such as msstat, imapd, etc.) will attempt to authenticate to mseventd by making sure that the certificate presented by mseventd is the same as the one used by mseventd clients. If TLS is not configured to use certificates (e.g. ADH cipher is enabled), then this option is ignored.

Default value: `true`

Example: `false`

LDAP attribute name: -None-

XML option name: `msevent_tls_auth`

Parent XML element: top level `<ms_options>`

2.2.13 Other options controlling M-Box directories

The following options are only available in **WebAdmin Advanced** mode. They specify different directories used by M-Box internally. In most cases these values should not be changed.

2.2.13.1 Run directory

Option location: Edit Message Store Configuration page, **Directories** section.

Description: Specifies the runtime directory for M-Box servers. On Unix this is the current directory for a server before it switches to the daemon mode and where it saves the pid file.

(UNIX) Default value: `/var/run`

(Windows) This option is not used.

Example: `/var/isode/ms/run`

LDAP attribute name: mboxRunDir

XML option name: `run_dir`

Parent XML element: top level `<ms_options>`

2.2.13.2 Temporary file directory

Option location: Edit Message Store Configuration page, **Directories** section.

Description: Specifies the directory that imapd and pop3d servers will use for temporary files.

(UNIX) Default value: `/tmp`

(Windows) Default value: `<`*`drive:`*`>\Isode\tmp`

Example: `/var/imap/tmp`

LDAP attribute name: mboxCacheTmpDir

XML option name: `cache_tmpdir`

Parent XML element: top level `<ms_options>`

2.2.13.3 Telemetry log directory

Option location: Edit Message Store Configuration page, **Directories** section.

Description: Specifies top level directory for IMAP, POP3, ManageSieve and ms_syncd protocol trace logging. If this option is not specified, then trace logging is disabled. Trace logging for a user "jane" can be enabled by creating a *<telemetry_log>/jane* directory. Each IMAP/POP/ManageSieve session is logged in a separate file. Files for unauthenticated sessions are created in the `<telemetry_log>` directory itself. Once a session is authenticated, its telemetry file is moved under the *<telemetry_log>/<user>/<date>* directory, if one exists. Trace logs for sessions on the same day are created in a subdirectory named after the date, for example trace logs for all sessions that have occurred on 7th of December 2005 can be located in *<telemetry_log>/jane/2005/12/7*.

Default value: -None- (protocol trace logging disabled)

Example: `/var/isode/ms/log`

LDAP attribute name: mboxTelemetryLogDir

XML option name: `telemetry_log`

Parent XML element: top level `<ms_options>`

2.2.13.4 IMAP to POP3 mapping database

Option location: Edit Message Store Configuration page, **Directories** section.

Description: Specifies the location of the user mapping database, used by IMAP-to-POP3 gateway and automatic migration from IMAP/POP. The mapping database can describe POP3/IMAP back-end specific to a user, as well as which POP3/IMAP back-end username corresponds to the IMAP front-end username.

(UNIX) Default value: `/etc/isode/ms_sync.xml`

(Windows) Default value: *<drive:>*`\Isode\etc\ms_sync.xml`

Example: /opt/isode/share/ms_sync.xml

LDAP attribute name: mboxSyncUserdb

XML option name: `ms_sync_userdb`

Parent XML element: top level `<ms_options>`

2.2.13.5 Using LDAP for storing IMAP to POP3 mapping database

Option location: Edit Message Store Configuration page, **Directories** section.

Description: This option controls if LDAP is used for storing user mapping database, used by IMAP-to-POP3 gateway and automatic migration from IMAP/POP. The mapping database can describe POP3/IMAP back-end specific to a user, as well as which POP3/IMAP back-end username corresponds to the IMAP front-end username.

Default value: `true`

Example: `false`

LDAP attribute name: -None-

XML option name: `ms_sync_sasl_lookup`

Parent XML element: top level `<ms_options>`

2.2.14 Advanced options controlling M-Box performance

Changing options specified in this section is not recommended, as it can degrade performance.

2.2.14.1 Minimal number of worker threads

Description: This option specifies the minimal number of worker threads that will be created for any work queue. This value can't be less than 1.

The exact number of worker threads depends on the number of CPU cores and the value of the `thread_pool_max` option. If the number of CPU cores is less than or equal to the value of the `thread_pool_min`, then the `thread_pool_min` threads will be created. If the number of CPU cores is greater than or equal to the value of the `thread_pool_max`, then the `thread_pool_max` threads will be created. Otherwise the number of threads is equal to the number of CPU cores.

Default value: -None-

Example: `4`

LDAP attribute name: -None-

XML option name: `thread_pool_min`

Parent XML element: top level `<ms_options>`

2.2.14.2 Maximal number of worker threads

Description: This option specifies the maximal number of worker threads that will be created for any work queue.

The exact number of worker threads depends on the number of CPU cores and the value of the `thread_pool_min` option. If the number of CPU cores is less or equal to the value of the `thread_pool_min`, then the `thread_pool_min` threads will be created. If the number of CPU cores is greater or equal to the value of the `thread_pool_max`, then the `thread_pool_max` threads will be created. Otherwise the number of threads is equal to the number of CPU cores.

Default value: `8`

Example: `16`

LDAP attribute name: -None-

XML option name: `thread_pool_max`

Parent XML element: top level `<ms_options>`

2.2.14.3 Use or memory maps

Description: This option controls if messages will be memory mapped when a POP3/IMAP client is requesting message content download.

This option should be set to `false` if M-Box is running on Windows and accessing mailboxes stored on CIFS network filesystem.

Default value: `true`

Example: `false`

LDAP attribute name: -None-

XML option name: `use_mmap`

Parent XML element: top level `<ms_options>`

2.2.14.4 Maximum write buffer size

Description: This option controls the maximum size of the write buffer (in bytes) used by M-Box services. If a client is not reading data quickly enough, the write buffer will grow in size and can cause scalability problems. If for a particular TCP session the write buffer size reaches the specified value (or the default value), the TCP connection is closed to prevent Denial of Service attacks.

Default value: `131072`

Example: `1048576`

LDAP attribute name: -None-

XML option name: `max_net_buffer`

Parent XML element: top level `<ms_options>`

2.2.15 Integration with SMTP server

2.2.15.1 Automatic addition of LASER routing attributes in "msadm add"

Description: This option controls whether **msadm add** command automatically adds **mail** and **mailLocalAddress** attributes with values equal to full username to user's LDAP entry. This option is enabled (`true`) by default.

Default value: `true`

Example: `false`

LDAP attribute name: -None-

XML option name: `auto_add_laser_routing`

Parent XML element: top level `<ms_options>`

2.2.16 SASL options

SASL options are described in Chapter 4, *User Authentication*.

2.3 Logging

2.3.1 Getting started

By default M-Box sends log messages to the "mail" syslog facility. The severity levels used by M-Box are:

- CRIT - Critical errors which require prompt administrator action
- ERR - I/O and System errors. The syslog message includes the specific file and OS specific error
- WARNING - Protection mechanism failures, client inactivity timeouts
- NOTICE - Authentications, both successful and unsuccessful

- INFO - LMTP delivery information, new and closed connections
- DEBUG - Debug information including BAD protocol traces.

If you wish to modify the default logging settings for the M-Box application, you should do the following:

Copy the file mboxlogging.xml from *(SHAREDIR)* into *(ETCDIR)*.

On Windows, a shortcut to the Log Configuration Tool will have been set up in the Isode folder on your Start menu.

On Unix, run */opt/isode/sbin/logconfig*.

Once the GUI is running, open mboxlogging.xml from *(ETCDIR)*. You will see a display of a number of predefined logging streams used by the M-Box, which can be modified as required. For full details of the options available, Section 5.3 of *GENSERV: General Services Administration Guide*.

2.3.2 How logging works

2.3.2.1 Record types

Isode server programs can write two types of log records during normal execution - Audit records and Event records.

Audit records are used to record “auditable events” - message submission, for example. They do not have a severity level associated with them, and have a well defined format, so that they can be easily parsed. Audit records normally consist of an event-type indicator, followed by a list of key=value pairs.

Event records are used to record errors, normal program operation, or to provide debugging information. They are associated with a particular severity level, and contain freeform text with substituted data items. The freeform text is contained in a separate dynamically-loaded library (on Windows) or a message catalog (on Unix), which makes it possible to replace the standard set of English messages with equivalent text in other languages simply by substituting a suitable message file.

No output mechanism is directly associated with log records. When an event or audit record is generated by an application, then whether or not it is logged, where it is logged to, and what the output of the log looks like, depends on what output streams have been configured.

Currently M-Box processes don't generate any Audit events.

2.3.2.2 Output streams

An output stream is a description of how a particular set of event and audit records should be recorded or displayed. Multiple output streams may be configured for an application, and whenever an event or audit record is generated, the logging subsystem checks to see which, if any, of the available output streams is eligible to process it.

As well as defining which records are eligible to be logged, the configuration of an output stream also determines the format of the messages that are produced by the stream.

This means that a single event or audit record may be processed by one or more separate streams (or by no stream at all), and that, in the case of multiple streams, the messages output by the streams may be of differing formats, containing more or less detail. For example, it would be possible to configure one output stream to generate a brief message about all "warning" level events, and another to generate a detailed message about a specific "warning" event which is of particular interest.

Three stream types are currently available: the file type, where the records are output to a file, the system type, where the records are passed to the system event log (syslog on

Unix-type systems and the Application Event Log on Windows), and the `tty` type, which is identical to file type, except that the records are written to either `stdout` or stderr.

2.3.2.3 Configuration storage and loading

Information about output stream configuration is stored as XML. All Isode applications will load the XML contained in the file *logtailor.xml*, located in *`(ETCDIR)`* or *`(SHAREDIR)`*, if it exists, at startup. This filename and location can be overridden if required by defining the environment variable `LOGTAILOR` to be an alternative filename or filepath.

An application may then load a private stream configuration. In the case of the M-Box, this is contained in the *mboxlogging.xml* file. A default version of this file is located in *`(SHAREDIR)`* - if you wish to make changes, copy this file into *`(ETCDIR)`* and modify this version. If the configuration file exists in both *`(ETCDIR)`* and *`(SHAREDIR)`*, the version in *`(ETCDIR)`* will be used.

2.3.2.4 Format of messages in output streams

When a given audit or event is generated, then for each output stream that is configured to process records of that type, the settings for the output stream determine the format of the message that is output. In the case of file and tty streams, the stream may be configured to contain any combination (including none) of the following fields:

date and time
: The format of date and time is configurable on a per-stream basis.

program name
: The name of the program generating the message. Any "isode" prefix will have been removed, and the program name will be truncated to 8 characters.

process id
: Identifies the process.

thread id
: This field may be useful to distinguish separate threads in the same process.

username
: The username of the process which generated the record. This field is only meaningful on Unix systems. If the username cannot be established, then a numeric UID is logged.

severity
: Audit records have no associated severity, but event records always have a severity, which, if displayed, is represented using one of the following single letters, as follows:

  - `I` - Info
  - `N` - Notice
  - `S` - Success
  - `D` - Detail
  - `W` - Warning
  - `E` - Error
  - `F` - Fatal
  - `C` - Critical
  - `L` - AuthOK
  - `A` - Authfail
  - `X` - Debug
  - `P` - PDU

facility code
: The name of the facility which generated the message. Audit records are not associated with a particular facility.

message identifier
: An identifier representing the event. Audit records do not have a message identifier

text
: The formatted text describing this event. Audit records do not have a text field

supplementary audit record parameters
: For certain types of audit records, extra information may be associated with the record, and if the stream is suitably configured, this will be included as a sequence of *`key`*`:`*`value`* pairs on the end of the message.

windows event log category
: Configures which of the predefined event categories will be used for the event log.

syslog config
: This allows control over various aspects of messages written to syslog. Available options are:

  - `console`: Write directly to system console if there is an error while sending to system logger.
  - `stderr`: Print to `stderr` as well.
  - `pid`: Include process ID.
  - `severity`: Include a single letter indicating the severity level of the event.
  - `facility`: Include the name of the facility for the message.
  - `ident`: Include the string identification of the event.
  - `text`: Include the formatted text for the event.
  - `firstonly`: If a message set is being logged, only log the first message in the set.

syslog facility
: The facility which should be used to log events.

2.4 Upgrading from a previous version

Before upgrading M-Box from a previous version, you need to run *`(SBIN)`*`/mbox` **uninstall**. This will remove information about M-Box services, but will not affect other configuration (like *`(ETCDIR)`*`/ms.conf`) or existing email.

Chapter 3 M-Box User Management

This section talks about how user information is stored and what kind of information about user accounts can be stored.

3.1 How M-Box stores users

Users, and information about their mailboxes, are stored in an LDAP directory in entries which have the **mboxUser** object class. This information may be downloaded from an external provisioning system, or managed directly in the LDAP directory. Cobalt provides a convenient user interface for managing this information using the web.

3.2 User administration using the Cobalt web application

Use **Users View** screen to select a user to edit or to add a user. The information is divided into several sections.

3.2.1 Account name

Option location: Personal tab.

Description: The "Primary Email Address and XMPP JID" is a required attribute and its value is used to uniquely identify each email account.

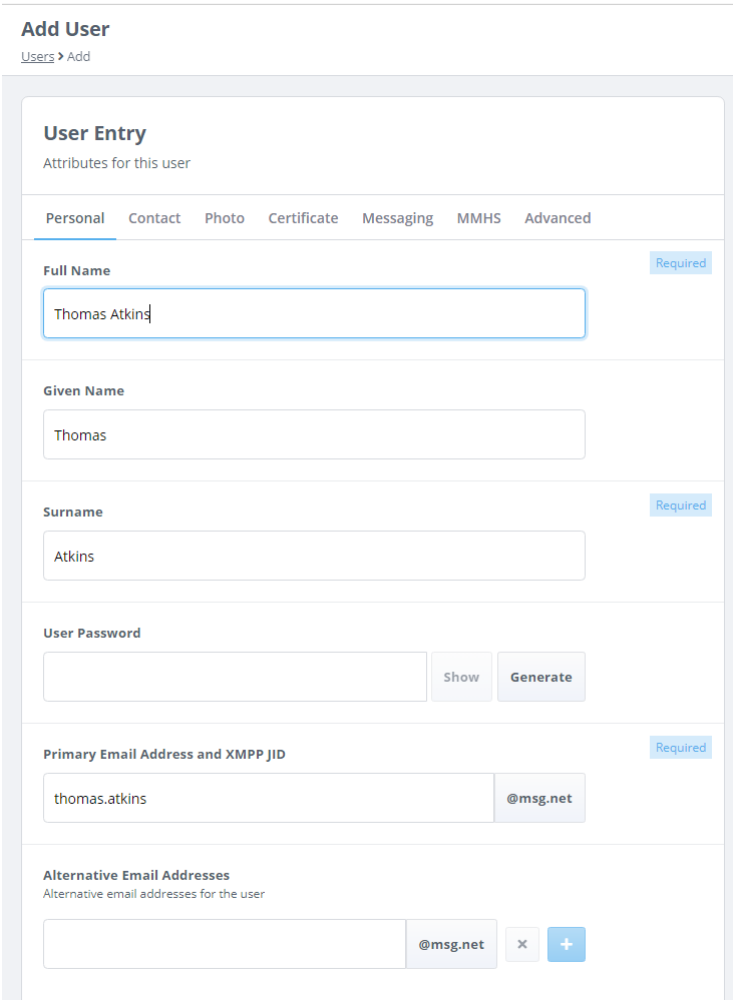
The account name is also used as the username when accessing IMAP, POP3 and ManageSieve services.

Please enter the left hand side (e.g. `joe` for `joe@example.com`) of the account name in the "Primary Email Address and XMPP JID" field. The domain part is either fixed or can be selected from a drop down list.

Example: Jack.Smith

LDAP attribute name: left hand side of the account name is stored in the **CN** attribute.

> **Note:** The full account name is also stored in **UID** attribute and the attribute specified in the **saslUsernameAttribute** attribute in the DSA's own entry.



3.2.1.1 User Password

Option location: Personal tab.

Description: Each account with IMAP, POP and/or ManageSieve access must have a password.

Example: `mysecret`

LDAP attribute name: userPassword

3.2.1.2 Account email addresses

Option location: Personal tab.

Description: This multivalued attribute contains all email addresses associated with the account specified in the **Alternative Email Addresses** field. This value is used for mail routing, i.e. a message sent to any of the listed email addresses will end up in the account mailbox or will be sent to the mailing list.

Example: `jack@example.com`

LDAP attribute name: mailLocalAddress

3.2.2 User information

Options described in this section are purely informational. They are only used to help administrators to uniquely identify users.

3.2.2.1 Given Name

Option location: Personal tab.

Description: First name of the person. This field is used to help administrators locate and identify an account.

Example: `Jack`

LDAP attribute name: givenName

3.2.2.2 Surname

Option location: Personal tab.

Description: Last name of the person. This field is used to help administrators to locate and identify an account.

Example: `Smith`

LDAP attribute name: surname

3.2.2.3 Full Name

Option location: Personal tab.

Description: This field lets administrators to locate and identify an account for cases when **First Name** + **Surname** are not unique.

Example: `Jack Smith Jr.`

LDAP attribute name: displayName

3.2.3 M-Box specific attributes

Option described in this section affect M-Box user account and associated mailboxes.

3.2.3.1 Message Quota

Option location: Messaging tab.

Description: Specifies the per-user quota limit on total size of all messages in all user mailboxes. If this value is not specified, the global default applies. This value is in Kilobytes.

Example: `102400`

LDAP attribute name: mboxMessageSizeQuota

3.2.4 User administration using command line tools

The **msadm** utility can administer both user information configured in the directory and local mailbox information. **msadm** is primarily used for performing operations that can't be done by managing data in the Directory, for example calculating or rebuilding quota usage. Modification of user and account information will usually be done directly using Cobalt or an LDAP tool.

You must always use **msadm** to rename or delete the user's mail volume in the mail database. It is possible to configure **msadm** to only manage the mail database.

Command line switches for **msadm**:

-c *configuration_file*
Specifies the name of the configuration file. The default is *(ETCDIR)/ms.conf*. **msadm** will start and run with defaults if there is no configuration file.

msadm can execute a single subcommand, if it is specified on the command line (.e.g. **msadm path user1@example.org**). If no subcommand is specified on the command line, **msadm** will read commands from standard input and output results to standard output.

Subcommands for **msadm**:

add {-p password|-m|-d} [{-r|-n}] [-f] userid [<properties>]
Adds a new user to the mailstore by creating the userid in the SASL database. The account is either created enabled (with the password specified after the `-p`), in 'being migrated' mode (`-m`) or disabled (`-d`). If the account already exists (e.g. `Disabled`), the command fails unless the `-f` flag is also specified. `-r` (the default) will automatically add a default LASER routing attribute. `-n` can be used to avoid addition of LASER attributes.

An optional list of extra properties can be specified after the `userid`. Each element has a syntax of *<name>=<value>*. For example, the following example would create a user with 2 email aliases, both of which can be used to log into the user's account:

```
add -p pass1 user1@example.co.uk mailLocalAddress=jaz@example.com
mailLocalAddress=user1@example.com
mailRoutingAddress=user1@example.co.uk ir-userName=user1
```

add -d userid
Adds a new disabled user to the mailstore.

add -m userid
Adds a new disabled (email migration pending) user to the mailstore.

del [-k] userid
Deletes `userid` from the SASL database and its corresponding mail volume. When the `-k` flag is specified, user's mail is not deleted.

ren -l new_userid userid
Renames the `userid` to the new_userid in the SASL database and renames the `userid` mail volume to the new name.

passwd -p password userid
Changes the password for `userid`.

list [-d domain_reg] [-u user_reg] [-v] [-p]
List all users in the SASL database using the optional domain and user regular expression to match users against. `-u` can be used to specify the left hand side of username to match against. `-d` can be used to specify the right hand side of a username (domain) to match against. `-v` can be used to display values of properties, except for the user password property. `-p` can be used to display password property values.

du [-r] userid
Gets mailstore disk usage for a userid. The optional `-r` switch tells msadm to recalculate disk usage (can be slow!).

path [-e] userid
Returns full path to the userid's directory. This command would perform user canonicalization, unless `-e` option is specified. If the account doesn't exist, the command return path to user's directory if the user is created.

quota {get|set} userid [<quota>]
Retrieve or change per-user quota. **quota set** can be used to set per-user quota. **quota get** can be used to retrieve the current quota value. Quota values are in Kilobytes. The value `0` means no quota.

acl {list <owner> <mailbox> [<userid>] | set <owner> <mailbox> <userid> <rights> | delete <owner> <mailbox> [<userid>]}

List/grant/change/delete permission for other users to access <*owner*>'s personal mailbox <mailbox>. This command has 3 subcommands:

acl list <owner> <mailbox> [<userid>]

This command allows listing ACLs for a personal mailbox. If <*userid*> is specified, then only rights for the specified user will be listed, otherwise rights for all users will be listed.

acl set <owner> <mailbox> <userid> <rights>

This command allows granting/changing rights to the mailbox for the specified userid. Here <*rights*> is a string containing one or more RFC 4314 rights:

- `r` – open/read the mailbox
- `s` – modify the Seen state of messages in the mailbox (\Seen flag)
- `w` – modify all flags other than \Seen and \Deleted
- `i` – insert new messages to the mailbox using IMAP
- `k` – create submailboxes below the mailbox or rename existing mailboxes below the mailbox
- `x` – delete the mailbox, rename the mailbox to a new name
- `t` – mark messages as deleted (before they can be expunged or undeleted)
- `e` – perform expunge on the mailbox

Note: The ability to list the mailbox (the 'l' right described in RFC 4314) is always granted to a mailbox that has any other right described above. Also note that the ability to administer mailbox (set/change/delete ACL on a mailbox) is always granted to the owner of a mailbox and is never granted to anybody else.

---

acl delete <owner> <mailbox> [<userid>]

This command allows deleting of rights granted to the specified userid, or any rights granted to any user (if userid is not specified). If the specified <*userid*> has no right to the mailbox, the operation has no effect.

expire_mail -p admin_passwd [{-f input_file [-c] [-n]|-l|<userid>}]

Expire user mailboxes according to mailbox expiration policy. Here <*admin_password*> is a password for a designed user account, which was created using msadm and which is specified in *ms.conf* in `<admin_user>` XML element, for example:

```
<admin_user>admin@momail.org</admin_user>
```

Mail expiration is performed over IMAP, logging in as the `admin_user`, but then authorizing as the desired user (<*userid*>). Mail for a particular user can be expired when <*userid*> is specified. It is also possible to expire mail for all enabled users by calling **msadm** without the `userid` parameter.

Per-user expiration policy is stored in user's LDAP entry in the **isodeMboxMailExpirationPolicy** multivalued attribute. Each value has the following syntax:

```
<mailbox>$<retention_period>
```

where `<retention_period>` is in seconds. So such value can be interpreted as "expire mail in mailbox <*mailbox*> which is older than <*retention_period*> seconds". This attribute can be specified when a user is added with **msadm add**, for example:

```
add -p pass test@example.ca
isodeMboxMailExpirationPolicy=INBOX$86400
```

In order to prevent expiration of mail in a mailbox, a special value "`NONE` is used, e.g. `INBOX$NONE`. Alternatively a large value, such as `946080000` (30 years), can be specified in this attribute.

It is also possible to specify a global mail expiration policy that would apply to any user that doesn't have the **isodeMboxMailExpirationPolicy** attribute in his entry. This can be done using the `<expiration_policy>` XML element in *ms.conf*, e.g.:

```
<expiration_policy>INBOX$7777777</expiration_policy>
```

`expire_mail -l` only outputs the list of enabled users. This can be saved to a file, processed later on with `expire_mail -f <filename>`. When `-c` option is specified together with `-f`, it restarts a previous expiration attempt. When `-f <filename>` is used without `-c`, any previously created file with expiration progress state is replaced.

`-n <numconnections>` option can also be specified with `-f <filename>`. It controls the number of parallel IMAP connections used for expiration. If `<numconnections>` is 0, the default number of connections is used (currently 4).

migrate -p password userid [<*properties*>]
Initiate immediate mail migration for the user with the specified password. Upon successful migration the user will be created in the mailstore (with the specified password) and will be enabled.

An optional list of extra properties can be specified after the userid. They have the same syntax and semantics as the extra properties specified in the add command. See the description of the **add** command above.

dump vacation userid
Allows a system administrator to show the content of Sieve vacation database.

service service-name {grant|allow|deny|default|query} userid
Allows to manage and query which services are available to a userid. The following service types (service-name) are currently recognized:

Service name	Service description
imap	Controls access to email messages over IMAP4 protocol
pop	Controls access to email messages over POP3 protocol
sieve	Controls access to SIEVE scripts over ManageSieve protocol

A userid can have service access record. If the record specifies that access to a particular service is granted or prohibited, the specified access is used by M-Box applications to control access to the service. This is called "explicit access rule". If the record doesn't contain information about the service, or the record is missing, the default access rule specified in the `accessible_services` option is used. The latter is called "implicit access rule".

The "**service ... grant**" subcommand allows explicitly granting a userid access to a particular service. The "**service ... allow**" subcommand is a synonym for the "**service ... grant**". The "**service ... deny**" subcommand explicitly revokes userid's access to a particular service.

The "**service ... default**" subcommand removes all explicitly specified access by userid to a service. When a user has no explicitly specified access to a service, the default access rule specified in the `accessible_services` option is used.

The **"service ... query"** subcommand can be used to check what kind of access (whether implicit or explicit) a userid has to a service.

Chapter 4 User Authentication

This section talks about configuration options used to control how user entries are located and how authentication is performed.

4.1 SASL authentication

IMAP uses the Simple Authentication and Security Layer (SASL) [RFC 4422] framework for authentication. Isode M-Box services use extended version of the Cyrus SASL library to implement SASL.

SASL provides a method for adding authentication support with an optional security layer to connection-based protocols. It also describes a structure for authentication mechanisms. The result is an abstraction layer between protocols and authentication mechanisms such that any SASL-compatible authentication mechanism can be used with any SASL-compatible protocol. See RFC 4422 for more information.

4.1.1 Generic SASL options

4.1.1.1 List of SASL mechanisms

Option location: This option is not accessible through IMA.

Description: This option contains comma or space separated list of allowed SASL mechanisms. This option allows limiting which mechanisms are advertised by the IMAP server. The intersection of the set of available mechanisms with this list is returned in the IMAP `CAPABILITY` response: e.g. if "`PLAIN,DIGEST-MD5,GSSAPI`" are available and the value of this option is "`SRP,GSSAPI,DIGEST-MD5`", the `CAPABILITY` response will list at most DIGEST-MD5 and GSSAPI. "At most", because other SASL options like `min_ssf`, `max_ssf` and a global option `login_disabled` (see Section 2.2.8.1, “Clear text login disabled”) affect the final list of available options as well. If this option is not set, all installed SASL mechanisms are allowed. See Section 4.1.2, “SASL mechanisms” for detailed discussion of different SASL mechanisms.

Default value: -None-

Example: `GSSAPI,SRP,DIGEST-MD5`

XML option name: `mech_list`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.1.1.2 Minimal and maximal strength security factors

Option location: This option is not accessible through IMA.

Description: `min_ssf` and `max_ssf` options contain minimal and maximal SSF (strength security factor) respectively. SSF is an unsigned integer (with values from 0 to 255) usable by the caller to specify approximate security layer strength desired. It roughly corresponds to the effective key length for encryption, e.g:

- `0` = no protection (no security layer)
- `1` = integrity protection only
- `>1` = key length of the cipher

The default value is 0 for both options.

Default value:

Example: `1`

XML option name: `min_ssf` and `max_ssf`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.1.1.3 Location of SASL plugins

Option location: This option is not accessible through IMA.

Description: This option specifies the location of SASL plugins in filesystem.

Default value: `(LIBDIR)/sasl2`

Example (UNIX): `/usr/local/lib/sasl2`

XML option name: `plugin_dir`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.1.1.4 Password verification method

Option location: This option is not accessible through IMA.

Description: This option contains the name of the password verification method. Currently two password verification methods are supported: `auxprop` (read cleartext password) and `auxprop-hashed` (treat the password as hashed in the password database). The specified password verification method is used to verify passwords during SASL PLAIN authentication, as well as in IMAP LOGIN and POP3 PASS command.

Note: If this option is set to an invalid value, this will prevent users from authenticating used the aforementioned mechanisms.

Default value: `auxprop`

Example: `auxprop-hashed`

XML option name: `pwcheck_method`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.1.1.5 List of auxprop plugins

Option location: This option is not accessible through IMA.

Description: This option contains a space separated list of auxiliary property (`auxprop`) plugins used for password verification by SASL plugins. The default is `None`, i.e. use all installed auxprop plugins. If multiple plugins are specified, they will all be queried in the specified order.

Note: Auxprop plugins (such as LDAPDB) retrieve raw passwords from the remote end and then pass them to SASL library for performing password verification. In order to protect cleartext passwords from people who might be snooping on the wire, each auxprop plugin should be configured to use a secure communication channel, such as communication over physically secure network (e.g. Unix domain socket) or TLS encrypted connection.

Default value: `auxprop`

Example: `userdb_lookup ldapdb userdb_cache`

XML option name: `auxprop_plugin`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.1.2 SASL mechanisms

The M-Box supports multiple SASL mechanisms via a plugin system. When the IMAP server starts up it loads all the plugins installed in *(LIBDIR)/sasl2*. This makes it simple to completely disable certain mechanisms (by removing the plugin file and restarting the corresponding service, such as the IMAP server) or to add additional mechanisms (by copying in the new plugin and restarting the corresponding service).

Each mechanism supplied has different characteristics that might make it more or less useful for a given M-Box installation:

4.1.2.1 SASL mechanism characteristics

Mechanism	Approach	Security
PLAIN	Sends plaintext passwords across the network.	Very weak.
LOGIN	Sends plaintext passwords across the network.	Very weak.
CRAM-MD5	Basic challenge/response, but vulnerable to server spoofing attacks	Weak
NTLM	Basic challenge/response, using a Microsoft specific algorithm	Weak
DIGEST-MD5	Challenge/response	Better
GSSAPI	Kerberos V5 authentication	Good
SCRAM-SHA-1	Challenge/response	Good

4.1.2.2 Shared secret mechanisms

For CRAM-MD5, DIGEST-MD5 and SCRAM-SHA-1 there is a shared secret between the server and client (e.g. a password). However, in this case the password itself does not travel on the wire. Instead, the client passes a server a token that proves that it knows the secret (without actually sending the secret across the wire). For these mechanisms, the server generally needs a plaintext equivalent of the secret to be in local storage.

4.1.3 SASL options controlling user management

The following options are only available in the Internet Messaging Administrator Web Application Advanced mode. They specify options controlling how user entries are located in the LDAP server and how new users are created. In most cases those values should not be changed.

4.1.3.1 User object classes

Option location: Edit Message Store Configuration page, **SASL options controlling user management** section.

Description: This option specifies a comma separated list of object classes that would be used when the Internet Messaging Administrator or M-Box LDAPDB plugin need to create a new user entry.

Default value: `top,person,organizationalPerson,inetOrgPerson, inetUser,mboxUser,cmuSaslUser`

Example:
`top,person,organizationalPerson,inetOrgPerson,inetUser,mboxUser, extensibleObject`

LDAP attribute name: saslUserObjectClasses

XML option name: `ldapdb_user_ocs`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>` or the top level `<ms_options>` element.

4.1.3.2 Domain object classes

Option location: Edit Message Store Configuration page, **SASL options controlling user management** section.

Description: This option specifies a comma separated list of object classes that would be used when the Internet Messaging Administrator Web Application or M-Box LDAPDB plugin need to create a new domain entry.

Default value: `top,domain`

Example: `top,domain,dNSDomain`

LDAP attribute name: saslDomainObjectClasses

XML option name: `ldapdb_domain_ocs`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>` or the top level `<ms_options>` element.

4.1.3.3 Container object classes

Option location: Edit Message Store Configuration page, **SASL options controlling user management** section.

Description: This option specifies a comma separated list of object classes that would be used when the Internet Messaging Administrator Web Application or M-Box LDAPDB plugin need to create a new container entry. A container entry usually contains user or domain entries below it.

Default value: `top,untypedObject`

Example: `top,organizationalUnit`

LDAP attribute name: saslContainerObjectClasses

XML option name: `ldapdb_container_ocs`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>` or the top level `<ms_options>` element.

4.1.3.4 User entry filter

Option location: Edit Message Store Configuration page, **SASL options controlling user management** section.

Description: This option specifies an LDAP filter [RFC 4515] used to select M-Box user entries.

Default value: `(objectclass=mboxUser)`

Example: `(&(objectclass=mboxUser)(!(inetUserStatus=Deleted)))`

LDAP attribute name: saslUserEntryFilter

XML option name: `ldapdb_user_filter`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>` or the top level `<ms_options>` element.

4.1.3.5 Controlling if users with no passwords should be listed

Option location: Edit Message Store Configuration page, **SASL options controlling user management** section.

Description: This option specifies whether an additional check for presence of **userPassword**/**inetUserStatus** attributes should be done before returning user entry in "msadm list" output. Note that this option is useful when M-Box is configured to authenticate users against Microsoft AD.

Default value: `true`

Example: `false`

LDAP attribute name: -None-

XML option name: `ldapdb_user_check_attrs`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>` or the top level `<ms_options>` element.

4.1.3.6 Domain entry filter

Option location: Edit Message Store Configuration page, **SASL options controlling user management** section.

Description: This option specifies an LDAP filter [RFC 4515] used to select M-Box domain entries by LDAPDB plugin. The first `%s` is replaced with the LDAP attribute used to name domain entries (as specified in the **saslDomainAttribute** attribute in DSA's own entry), the second `%s` is replaced with the domain search mask or specific domain name.

Default value: `(&(%s=%s)(!(objectclass=mboxUser)))`

Example: `(objectclass=domain)`

LDAP attribute name: saslDomainEntryFilter

XML option name: `ldapdb_domain_filter`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>` or the top level `<ms_options>` element.

4.1.4 Advanced SASL configuration options

LDAPDB is a SASL plugin responsible for verifying user password and retrieving other information about users from an LDAP server. In order to use the LDAPDB plugin for user information, the *`(ETCDIR)`/ms.conf* must have the `auxprop_plugin` SASL option containing value `ldapdb`. LDAPDB-specific SASL options are described below. They can be used if M-Box users are stored in LDAP, but the M-Box server configuration and shared folders are not, or if users and M-Box server configuration are stored in two different LDAP directories.

Most of the options described in this section control how LDAPDB plugin binds and searches the Directory.

4.1.4.1 LDAPDB URI

Option location: This option is not accessible through Internet Messaging Administration Web Application.

Description: Specifies LDAP server URL(s). Multiple URLs can be specified as a space separated list of URLs. Recognized LDAP URL schema types are:

- `ldap://` (connection over TCP)
- `ldapi://` (connection over UNIX domain socket) [Not supported on Windows]
- `ldaps://` (connection over TCP with required TLS).

If this option is not specified, the value of the `config_location` option (or the value constructed from `ldap_server`/`ldap_port` options, if it is not specified) is used by default.

Default value: -None-

Example: `ldaps://secure.example.com`

XML option name: `ldapdb_uri`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.1.4.2 Bind DN

Option location: Directory Configuration Browser page, **LDAP options** section.

Description: Specifies the LDAP Bind DN. If both `ldapdb_dn` and `ldapdb_id` are not specified, the value of `ldap_bind_dn` option is used by default. This option is ignored if SASL bind is used is used to authenticate to the LDAP server. See also `ldapdb_mech`.

Default value: -None-

Example: `cn=Manager, o=Corp, c=US`

XML option name: `ldapdb_dn`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.1.4.3 Bind user ID

Option location: Directory Configuration Browser page, **LDAP options** section.

Description: Specifies the LDAP Bind userid. If both `ldapdb_dn` and `ldapdb_id` are not specified, the value of `ldap_bind_id` option is used by default. This option is ignored if simple bind is used. See also `ldapdb_mech`.

Default value: -None-

Example: `frank@example.com`

XML option name: `ldapdb_id`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.1.4.4 Bind password

Description: Specifies the LDAP Bind password that is used to bind to the LDAP server together with `ldapdb_dn` or `ldapdb_id`. If both `ldapdb_dn` and `ldapdb_id` are not specified, and this option is not specified, the value of `ldap_bind_pwd` option is used instead.

Default value: -None-

Example: `supersecret`

XML option name: `ldapdb_pw`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.1.4.5 Bind method

Description: Specifies the LDAP authentication method used by the LDAPDB plugin. If this option has the value `SIMPLE`, then LDAP simple bind is used with DN defined by the `ldapdb_dn` option. Otherwise it contains the name of a SASL mechanism to use in LDAP SASL Bind (and the userid defined by the `ldapdb_id` option is used). If both `ldapdb_dn` and `ldapdb_id` are not specified, and this option is not set, the value of `ldap_bind_method` option is used instead.

Default value: -None-

Example: `SIMPLE`

XML option name: `ldapdb_mech`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.1.4.6 TLS server certificate verification mode

Description: This option tells LDAPDB how to verify TLS server identity. The option may be set to "`demand`", "`hard`", "`try`", "`allow`" or "`never`". The meaning of various options is as follows:

`never`
: The client will not request or check any server certificate.

`allow`
: The server certificate is requested. If no certificate is provided, the session proceeds normally. If a “bad”(*) certificate is provided, it will be ignored and the session proceeds normally.

`try`
: The server certificate is requested. If no certificate is provided, the session proceeds normally. If a “bad”(*) certificate is provided, the LDAP session is immediately terminated.

`demand` or `hard`
: These keywords are equivalent. The server certificate is requested. If no certificate is provided, or a “bad”(*) certificate is provided, the LDAP session is immediately terminated.

(*) A certificate is considered “bad” if it is expired, revoked, untrusted (e.g. not signed by a known CA), or which contains **SubjectAltName** or **Subject DN** which don't cover the LDAP server. The default is “`never`”, i.e. don't verify TLS server identity.

Default value: `never`

Example: `demand`

XML option name: `ldapdb_tls_server_cert`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.1.4.7 STARTTLE

Description: This option tells LDAPDB to use STARTTLS and can optionally specify how LDAP server side TLS certificates should be verified. The option may be set to

"demand", "hard", “try”, "allow" or "never". It can also contain a value “yes”, “true” or “1”, all of which just enable use of STARTTLS. The meaning of various other options is as follows:

never
: The client will not use STARTTLS.

allow
: The server certificate is requested. If no certificate is provided, the session proceeds normally. If a “bad”(*) certificate is provided, it will be ignored and the session proceeds normally.

try
: The server certificate is requested. If no certificate is provided, the session proceeds normally. If a “bad”(*) certificate is provided, the LDAP session is immediately terminated.

demand or hard
: These keywords are equivalent. The server certificate is requested. If no certificate is provided, or a “bad”(*) certificate is provided, the LDAP session is immediately terminated.

(*) A certificate is considered “bad” if it is expired, revoked, untrusted (e.g. not signed by a known CA), or which contains SubjectAltName or Subject DN which don't cover the LDAP server.

The default is “never”, i.e. don't use STARTTLS.

Default value: -None-

Example: demand

XML option name: ldapdb_starttls

Parent XML element: <sasl> XML element below the top level <ms_options>

4.1.4.8 Cipher list

Description: Specifies the list of space (or colon) separated TLS ciphers that the LDAP client is allowed to use. See Appendix D, *TLS Cipher List Formats* for more details.

If this option is not specified, the value of the tls_cipher_list *ms.conf* option is used instead.

Default value: DEFAULT

Example: DHE-RSA-AES256-SHA DHE-DSS-AES256-SHA AES256-SHA

LDAP attribute name: mboxTlsCipherList

XML option name: tls_cipher_list

Parent XML element: <sasl> XML element below the top level <ms_options>

4.1.4.9 CA file (PEM)

Description: Specifies path to a PEM file containing one or more CA certificate used for servers' identity verification. The CA certificate is also the CA that signed the clients certificate contained in ldapdb_client_cert SASL option. On Windows this option can also contain a 'certstore:' URI (e.g. 'certstore:Root'), so that the corresponding CA certificates can be retrieved from the Windows Certificate Store. The CA certificates from this file are loaded on M-Box application startup. If this option is not set, this might result in a client's inability to verify server certificates, however TLS will still be able to provide data encryption.

See also Section 2.2.12.6, “Certificate file”.

Default value: -None-

Example: `/etc/isode/mbox-tls/ca_certificate.pem`

LDAP attribute name: -None-

XML option name: `ldapdb_ca_cert`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.1.4.10 Certificate file

Description: Specifies the full path to a file containing the client's own certificate. This certificate will be sent by the client to the LDAP server that wishes to confirm the client's identity when negotiating secure communication. The certificate format can be either PEM, DER, or PKCS12. The file extension has to match the format, i.e. *.pem*, *.der*, or *.p12*. The file extension *.crt* is also accepted here, in which case the file must contain a PEM certificate.

On Windows this option can also contain a '`certstore:`' URI that points to a certificate in the Windows Certificate Store (e.g. '`certstore:My:sha1:09cfea1e5e5f2302bbd77f91c18ffb5e66135a01`').

Note: The *.p12* file may also contain CA certificates.

Default value: -None-

Example: `/etc/isode/mbox-tls/mbox_certificate.pem`

LDAP attribute name: -None-

XML option name: `ldapdb_client_cert`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.1.4.11 Key file

Description: Specifies the full path to a PEM/DER file containing the private key belonging to the client's certificate. The key file format is determined from the file extension (*.pem*, *.der*, *.crt* (PEM), *.key* (PEM)). If the file extension is not recognized, the file is assumed to be in the same format as the certificate file (`ldapdb_client_cert`). If this option is not set, the value of the `ldapdb_client_cert` is used.

This value is not used when `ldapdb_client_cert` option points to a PKCS 12 (*.p12*) file.

Default value: -None-

Example: `/etc/isode/mbox-tls/mbox-key.pem`

LDAP attribute name: -None-

XML option name: `ldapdb_client_key`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.1.4.12 Key password

Description: Specifies the password used to decrypt the client's private key. This option is empty by default, which means that the private key is not protected by any password. This option is not used on Windows when the corresponding certificate/private key is stored

in the Windows Certificate Store (i.e. a certstore: URI is used). This value can be obfuscated using the Service Key facility (see Section 2.1.5, “Use of ServPass for password obfuscation”).

Default value: -empty string-

Example: `SuperS0cret-Password`

LDAP attribute name: -None-

XML option name: `ldapdb_client_key_passwd`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.1.4.13 Controlling how user account existence is verified by msadm list command

Description: This option specifies whether **msadm list** should check for presence of the **userPassword**, the **inetUserStatus** or any attribute which has name starting with **cmusasl** for a user before returning the user information. By default (`true`) no user account is considered to exist unless it contains one of these attributes. When using Active Directory for storing user account information, this option should be set to `false`.

Default value: `true`

Example: `false`

LDAP attribute name: -None-

XML option name: `ldapdb_user_check_attrs`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.1.4.14 FIPS 140 mode

Description: The 140 series of Federal Information Processing Standards (FIPS) are U.S. Government computer security standards that specify requirements for cryptography modules. When this option is set to `true`, it enables FIPS-140 compliance mode, which will restrict which hash and encryption algorithms are allowed in TLS and SASL.

Default value: constructed from Windows FIPS 140 registry settings, `false` on Unix platforms

Example: `true`

LDAP attribute name: -None-

XML option name: `fips140_mode`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.1.4.15 Configuration file with additional LDAP options

Description: The filename specified in this option will be put into the server's `LDAPRC` environment variable, and `libldap-specific` config options may be set in that *ldaprc* file. The main purpose behind this option is to allow a client TLS certificate to be configured, so that SASL/EXTERNAL may be used between the LDAPDB and the LDAP server. This is the most optimal way to use this plugin when the servers are on separate machines.

Default value: -None-

Example: `demand`

XML option name: `ldapdb_rc`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.1.4.16 Username to DN translation method

Description: This option controls how translation from SASL username to Directory DNs is performed. Allowed values are: `proxyauth` (use LDAP “Who Am I” extended operation with proxy authorization control), `emulate` (perform translation in LDAPDB) and `fallback` (try `proxyauth`, use `emulate` if this fails).

Default value: `emulate`

Example: `proxyauth`

XML option name: `ldapdb_map_method`

Parent XML element: `<sasl>` XML element below the top level `<ms_options>`

4.2 Userid canonicalization during authentication process

For user convenience M-Box servers support virtual servers. The following diagram shows how these capabilities affect authentication process:

- Default domain is added to userid, if it is not fully qualified (i.e. doesn't contain @domain part)
- Credentials for the canonicalized userid are retrieved and verified.

If M-Box servers are configured to listen on multiple IP addresses (interfaces), one or more of them can be configured to be a virtual server. Each virtual server can have own default domain. The default domain is used to fully qualify any unqualified userid.

When M-Box server receives an unqualified userid it first checks if there is a virtual server entry described using the `domain_map` XML element of the configuration file (or **mboxListenDomainMappings** or **mboxRemoteDomainMappings** attributes if M-Box server configuration is stored in LDAP). If such entry is found, the default domain specified there is used. If there is no corresponding entry, the value of the `domain` option is used instead. If the domain option is not set, the fully-qualified hostname of the machine running M-Box is used instead.

Example:

An M-Box server is configured to listen on 2 interfaces, one is 1.1.1.1 with default domain ISP.NET and another one is 2.2.2.2 with default domain EXAMPLE.COM If a user is trying to connect to 1.1.1.1 with userid "test", the M-Box server will canonicalize it to "test@ISP.NET".

If a user is trying to connect to 2.2.2.2 with userid "test", the M-Box server will canonicalize it to "test@EXAMPLE.COM". If a user is trying to connect to 1.1.1.1 or 2.2.2.2 with userid "test@example.net", the M-Box server will canonicalize it to " test@example.net", i.e. it will not change the provided userid.

Chapter 5 M-Box Shared Folders

M-Box supports shared folders for ease of sharing of email messages between a group of IMAP users.

M-Box supports shared folders. A hierarchy of mailboxes can be shared by defining a shared root and specifying what kind of access different users have to the hierarchy. In LDAP a shared root hierarchy is represented as an entry with **mboxSharedMailboxRoot** object class. In *(ETCDIR)/ms.conf* configuration file a shared root hierarchy is represented by a `shared_root` XML element.

Each shared folder root object has 2 mandatory attributes: shared root filesystem location and user friendly name. Also the shared folder root contains access control list and can have an (optional) email address. The attributes are described in details in subsequent sections.

5.1 Name

Option location: Edit Shared Folder Root page.

Description: This attribute specifies a user's friendly name that will be used when listing the shared folder hierarchy in IMAP. The actual IMAP mailbox name for the root of the shared folder hierarchy would be *Shared Folders/<name>*. This attribute must be unique across all shared folders.

Example: `Email Archive`

LDAP attribute name: CN

XML attribute name: `name`

XML element: `<shared_root>` XML element below the top level `<ms_options>`

5.2 Root directory

Option location: Edit Shared Folder Root page.

Description: Value of this attribute is a file name, it specifies physical location of the mailbox root. This attribute must be unique among all shared folders.

> **Note:** A shared root is autocreated on disk once any user that has access to it logs in.

Example: `C:\Isode\ms\shared\archive`

LDAP attribute name: mailboxRoot

XML attribute name: `path`

XML element: `<shared_root>` XML element below the top level `<ms_options>`

5.3 Email address

Option location: Edit Shared Folder Root page.

Description: This optional attribute allows for specifying an email address that can be used to deliver mail to the shared folder using LMTP. LMTP plus addressing can be used to deliver mail to a submailbox. For example, if the email attribute is `archive@example.com`, sending email to `archive@example.com` will deliver mail to the top level mailbox of the shared mailbox hierarchy. Sending email to `archive+subfolder@example.com` will deliver mail to the submailbox called "`subfolder`" of the top mailbox. And sending email to `archive+leads/john@example.com` will deliver mail to the submailbox called "`john`" of the mailbox "`leads`".

Example: `archive@example.com`

LDAP attribute name: mail

XML attribute name: `email`

XML element: `<shared_root>` XML element below the top level `<ms_options>`

5.4 Quota

Option location: Edit Shared Folder Root page.

Description: Specifies the default quota limit on total size of all messages in the shared root hierarchy. This value is in Kilobytes. The default value is `0`, which means that there is no quota limit.

Default value: `0`

Example: `102400`

LDAP attribute name: mboxMessageSizeQuota

XML attribute name: -no corresponding attribute-

> **Note:** This option is currently ignored by M-Box.

5.5 Access Control list

Option location: Edit Shared Folder Root page.

Description: In LDAP access permissions are stored in the multivalued **mboxAccessControlList** attribute, each value has the following syntax: "**<username>**

<permissions>", where **<permissions>** is a comma separated list of permissions as described in Section 5.8, “Permissions”. User names starting with "`user:`" reference to users, the "`user:`" prefix can be omitted. User names starting with "`group:`" reference to groups (see below). A special user name "`anyone`" is reserved and applies to all users that don't have explicit permissions.

In XML each element of the access control list is described as a child element for the `shared_root` XML element. A child element named "`user`" describes rights a user has. A child element named "`group`" describes rights a user belonging to the group has, unless the user has a corresponding "`user`" element. Only one defined group can be used in any access control list, plus a special group called "`anyone`" which includes all users. Name of a user/group is specified in the "`name`" attribute and access permissions are specified in the "`access`" attribute. The access attribute contains a comma separated list of words that describe different access permissions. Currently recognized permissions are described in Section 5.8, “Permissions”.

See Section 5.9, “Administrator defined groups” for description of how groups of users are stored in LDAP.

LDAP attribute name: mboxAccessControlList

XML element: `<user>` or `<group>` XML element below the second level `<shared_root>`

Example (XML):

```
<ms_options>
 …
<shared_root name='Email Archive' path='/work/ms/shared/archive'
   email='archive@example.com'>
<group name='anyone' access='read,write'/>
<user name='brian' access=''/>
<user name='natalia' access='read,write,manage'>
</shared_root>
…
</ms_options>
```

The user "brian" is explicitly denied any access rights to the mailbox hierarchy, so he will not see it. The user "olga" will have "read,write" rights, because she doesn't have an explicit “user” element with name attribute containing her username.

Example (LDIF): Below is representation of the example given above in LDIF

```
dn: cn=email-archive, cn=Shared Folders, cn=mail.example.com,
 cn=Servers, cn=Internet Mailstore, cn=Messaging Configuration,
 ou=MHS, c=GB
objectClass: mboxSharedMailboxRoot
objectClass: top
cn: email-archive
description: Email Archive
mail: archive@example.com
mailboxRoot: /work/ms/shared/archive
mboxAccessControlList: anyone read,write
mboxAccessControlList: brian none
mboxAccessControlList: natalia read,write,manage
```

5.6 Description

Option location: Edit Shared Folder Root page.

Description: This optional attribute contains human readable text describing the shared folder hierarchy.

Example: Archive of internal mailing lists

LDAP attribute name: description

XML attribute name: -no corresponding attribute-

5.7 Status

Option location: Edit Shared Folder Root page.

Description: This optional attribute is used internally by M-Box. If the value "`Deleted`" is selected, the shared folder hierarchy is scheduled to be erased from filesystem. If the value "`Active`" is selected, the shared folder hierarchy is accessible to users. If this attribute is not set the default value is "`Active`".

Example: `Deleted`

LDAP attribute name: mboxFolderStatus

XML attribute name: -no corresponding attribute

5.8 Permissions

Currently recognized permissions are:

`none`
: explicitly deny any access to the mailboxes (the same as the empty value. Note that the empty value can't be used in LDAP)

`read`
: allow to open the mailbox and read/search messages

`write`
: allow to append/copy messages to the mailboxes, expunge messages and change flags

`manage`
: allow to create/rename/delete mailboxes

`all`
: a synonym for "`read,write,manage`"

5.9 Administrator defined groups

An administrator can define groups. In XML a group is represented as a "`group`" XML element. The group name is defined in the "`name`" attribute of this XML element. An optional "`type`" attribute can be specified. Currently the value of this attribute is ignored. The group XML element contains one or more nested "`user`" element, which define group members. A member name is defined in the "`name`" attribute of the "`user`" element.

> **Note:** Groups can't include other groups. Also, it is not possible to redefine the special group "`anyone`".

Groups of users are stored in an LDAP directory as entries belonging to the **mboxGroup** object class. Such entries must be located below the "cn=Groups" entry which is below the M-Box configuration entry. Members of a **mboxGroup** group are listed in the **mboxGroupMember** multivalued attribute.

IMA supports editing of groups of users. This can be done on the **Groups** page.

Example (XML):

```
<ms_options>
 …
<group name='staff'>
<user name='tim'/>
<user name='kurt'/>
<user name='marry'/>
</group>
…
</ms_options>
```

5.10 Special groups used by IMAP URLAUTH extension (RFC 4467)

M-Box imapd recognizes a special group called "`submit`". This group defines users who are allowed to access "`submit+`" access identifiers (as defined in RFC 4467) using URLFETCH command.

Chapter 6 Filtering Messages Using Sieve

This section talks about using Sieve Mail Filtering Language for automatic processing of emails on delivery.

Sieve is an Internet standards-track language for processing messages on delivery [RFC 5228]. The Isode Sieve implementation is based on the Cyrus Sieve implementation, and supports:

- Filing messages into specific folders [RFC 5228]
- Forwarding messages [RFC 5228]
- Rejecting messages [RFC 3028]
- Sending vacation replies [RFC 5230]
- Marking messages with IMAP flags [RFC 5232]

It can perform those actions based on headers or envelope information. The following Sieve extensions are also supported:

- Relational tests [RFC 5231]
- Subaddress extension [RFC 5233]
- Copying without side effects [RFC 3894]
- Date extension [RFC 5260]
- Extension for accessing mailbox metadata [RFC 5490]

Note that if a user has a Sieve script, the Sieve script runs authorized as *that* user, and the rights of the posting user are ignored for the purposes of determining the outcome of the Sieve script.

Users can manage SIEVE scripts using ManageSieve protocol. By default the ManageSieve server (isode.sieved) listens on port 4190. Web Mail clients like AvelSieve (companion to Squirrelmail), Ingo (companion to IMP WebMail) and Mail clients like Polymer support ManageSieve protocol.

SIEVE filtering in isode.lmtpd is enabled by default. In order to disable SIEVE filtering you should set the `mboxEnableSieve` option (see Section 2.2.11.1, “Enable SIEVE”) to `false`.

Chapter 7 M-Box POP to IMAP Gateway

M-Box POP to IMAP Gateway service allows M-Box to download and synchronize email from other POP or IMAP servers.

M-Box POP to IMAP Gateway enables mail clients to access Internet Standard POP3 mailboxes (located on other POP3 servers) using IMAP.

The back-end of the gateway accesses a POP server, the front end of the gateway is identical to that that of M-Box and provides an IMAP (and POP) interface. Messages are stored by the M-Box POP to IMAP Gateway to optimize access performance for an IMAP client. Operation is as follows:

1. When the IMAP Client connects, the back-end of the M-Box POP to IMAP Gateway connects to the POP server and downloads all messages which are not already held by M-Box POP to IMAP Gateway.
2. M-Box POP to IMAP Gateway is now in a position to provide IMAP access to those messages from IMAP capable clients.
3. While the IMAP connection remains open, the M-Box POP to IMAP Gateway back-end will poll the POP server (at configurable intervals) to retrieve new messages.
4. If the client deletes a message from M-Box POP to IMAP Gateway INBOX mailbox and the POP3 backend doesn't automatically expire downloaded mail, the message will then be deleted on the POP server, so that both servers remain in synchronization. This behaviour can be turned off with the `ignore_frontend_deletes` *ms.conf* option.
5. If a message is added to *INBOX* using IMAP **APPEND** or **COPY** command, it will remain there, unless explicitly deleted by a mail client. The POP3 protocol does not allow uploading messages, so the message cannot be added to the POP3 backend.
6. The M-Box POP to IMAP Gateway can notify the IMAP client when new email messages arrive using the IMAP **IDLE** command.

You can find detailed information about configuration options that affect M-Box running as a POP to IMAP Gateway in Section 2.2.9, “Gateway and migration mode specific options” of this administration guide.

7.1 Configuring M-Box to run as a POP to IMAP gateway

In order to make M-Box run as a POP to IMAP Gateway, you needs to perform the following 2 steps:

1. Configure and enable Gateway mode in *(`ETCDIR`)/ms.conf*
2. Enable isode.ms_syncd before starting M-Box

7.1.1 Enabling/configuring the Gateway mode

The Gateway mode is enabled by setting the `gateway_mode` option to `true`. A hostname or IP address of the default POP3/IMAP backend is specified in the `ms_syncd_host` option. The default IMAP/POP3 backend port number can be specified in the `ms_syncd_port` option, it will default to the default protocol port if not specified. The

default value of the delay between two subsequent synchronizations for a user can be specified in the `ms_sync_interval` option.

> **Note:** IMAP server doesn't allow for the **AUTHENTICATE** command when running in the Gateway mode. To authenticate to the server clients must use the LOGIN command instead. If your clients don't support **STARTTLS**, you should consider setting the `login_disabled` option to `false`.

A fragment of the *(ETCDIR)/ms.conf*for M-Box in the Gateway mode can look like the following:

```
<login_disabled>false</login_disabled>

<gateway_mode>true</gateway_mode>
<ms_syncd_host>pop.myisp.com</ms_syncd_host>
<ms_syncd_port>3110</ms_syncd_port>
<ms_sync_interval>600</ms_sync_interval>
```

7.1.2 Enabling mail migration service (isode.ms_syncd)

7.1.2.1 On Windows

M-Box in Gateway mode requires that an additional service is started: isode.ms_syncd. By default this service is disabled on all platforms.

To enable isode.ms_syncd, you need to run: *(SBINDIR)*`\mbox enable ms_syncd`

Once this is done, you will see that isode.ms_syncd status will change in the "*(SBINDIR)*`\mbox status`" output.

Once this is done M-Box needs to be restarted (or started for the first time).

7.1.2.2 On Linux

On RedHat Linux, isode.ms_syncd can be enabled by editing one line in the *(SBINDIR)/mbox.sh* script. Add "`ms_syncd`" to the end of the following line:

```
DAEMONS="mseventd imapd pop3d lmtpd sieved"
```

I.e. it should become:

```
DAEMONS="mseventd imapd pop3d lmtpd sieved ms_syncd"
```

Chapter 8 M-Box Migration Mode

M-Box Migration mode allows for migration of user's mail and authentication information from an existing POP or IMAP system to M-Box.

M-Box Migration mode allows for migration of user's mail and authentication information from an existing POP or IMAP system to M-Box. Unless the migration happens from a POP3 backend which expires all downloaded mail, all migrated mail is preserved intact on the backend server.

The back-end of the gateway accesses a POP or IMAP server, the front end of the gateway is identical to that that of M-Box and provides an IMAP (and POP) interface. M-Box in Migration mode operates as follows:

1. When the IMAP Client connects, isode.imapd verifies if the username/password provided by the client can be found in M-Box's authentication database. If the username is found and password matches, then the user can access its mail as usual.
2. If the username is not found, then the isode.imapd requests from the isode.ms_syncd to migrate user's mail. The isode.ms_syncd server connects to the backend POP/IMAP server and downloads all messages from all personal mailboxes. Mailbox subscription is also migrated (only when migrating from an IMAP backend). Once migration is successful, the username/password are added to the M-Box's authentication database.
3. An administrator can force manual migration by running **msadm migrate** command. This can be used to restart a failed migration, but note that any mail **APPEND**ed to any user's mailbox since the previous migration attempt will be lost.

Note: A manual attempt to migrate an email account (using **msadm migrate**) may delete all existing mail from M-Box for the user being migrated, if such user already exists in M-Box.

You can find detailed information about configuration options that affect M-Box running in Migration mode in Section 2.2.9, “Gateway and migration mode specific options” of this administration guide.

Chapter 9 Live Monitoring

This section talks about using the msstat command line utility for remote monitoring of up/down status of M-Box services and monitoring of logged in IMAP/POP users.

M-Box includes the *(BINDIR)/msstat* utility for live monitoring of M-Box services. The **msstat** utility acts as a client to mseventd, which requests the list of logged in users and server status.

When *(BINDIR)/msstat* is called without parameters, it reports the list of running services (this currently doesn't include the ManageSieve server). For example:

```
(BINDIR)/msstat

online ~ pop3 test-rhel5-64   02:48
online ~ lmtp test-rhel5-64   02:48
online ~ imap test-rhel5-64   02:48
online ~ gw test-rhel5-64   02:48
online ~ pop3 test-win2k3-64   02:49
online ~ lmtp test-win2k3-64   02:49
online ~ imap test-win2k3-64   02:49
online ~ gw test-win2k3-64   02:49
```

With the `-p` option (“pretty output”), it reports a bit more information:

```
online ~   pop3  test-rhel5-64            Fri 23 Apr 12:54(02:52)
online ~   lmtp  test-rhel5-64            Fri 23 Apr 12:54(02:52)
online ~   imap  test-rhel5-64            Fri 23 Apr 12:54(02:52)
online ~   gw    test-rhel5-64            Fri 23 Apr 12:54(02:52)
online ~   pop3  test-win2k3-64           Fri 23 Apr 12:53(02:53)
online ~   lmtp  test-win2k3-64           Fri 23 Apr 12:53(02:54)
online ~   imap  test-win2k3-64           Fri 23 Apr 12:53(02:54)
online ~   gw    test-win2k3-64           Fri 23 Apr 12:53(02:53)
```

The first column displays `online ~` for rows showing running servers. The second column shows protocol name (`gw` corresponds to ms_syncd). The third column shows the hostname of a running server. The last column shows when the the server was started, with the up time in brackets `()`.

In order to display status of client connections belonging to a particular user (or a set of users), the `-u <regex>` option is used. If no special regular expression characters is used, the *<regex>* is interpreted as a part of username. For example `-u m` might report such users as “`milk`” and “`om`”. In order to report information about a specific user, use “`-u ^<username>`”.

For example:

```
at imap dhcp-261.example.com idling INBOX 07:16
ael imap 62.3.217.253 idling INBOX 00:52
ast imap esper.example.com idling INBOX 04:00
at imap dhcp-261.example.com idling Shared Folders/Spam Filing/false-negative
```

The first column displays name of the logged in user (empty for unauthenticated sessions). The second column shows protocol name. The third column shows the hostname of the client. The fourth column contains extra information, such as what the client is doing and which mailbox it has opened (for IMAP). The last column shows when the client was connected, with the connection duration in brackets `()`.

msstat has a number of other optional command line options:

By default **msstat** tries to do reverse DNS lookups for client's IP addresses. The `-n` option will force **msstat** not to use reverse DNS lookups when showing hostnames, i.e. it will show IP addresses of the connected clients.

`-d` adds an extra column in the default **msstat** (or **msstat -u** ...) output to show the server host to which the user is connected.

The `-f <event>` option will request continuous monitoring. `-f joe` will get events for user “joe”. `-f mbox:joe` will get mailbox events for user “joe”. If `-f ALL` is used, then all events will be reported. Each login and logout will be reported as separate lines. This option also uses slightly different output format and provides more information on active sessions:

```
service start: service=imap server_fqdn=andrew.example.com
time_stamp=1176108777.265625
conn open:     service=imap server_fqdn=andrew.example.com
session_id=456 client_ip=127.0.0.1 time_stamp=1176116686.546875
```

Appendix A Command Line Parameters for M-Box Service Applications

Details of common command line options for M-Box servers are listed in this section.

All M-Box service application accept standard set of command line parameters described below:

`-d`
: Run service application in debug mode. When this parameter is specified on UNIX, the service application will not detach and does not become a daemon. This allows for easy monitoring of the service application.

`-c configuration_file`
: Specifies the name of the configuration file. The default configuration file is *`(ETCDIR)`/ms.conf*. Note that if this parameter is not specified and the default configuration file doesn't exist, then the service application use hardcoded defaults.

`-s label`
: Specifies the logging label that is going to be used to identify this instance of the service application.

Appendix B Example XML Configuration

Example XML configuration is shown in this appendix.

This section provides an example of *(ETCDIR)/ms.conf* XML configuration file where user information is stored in LDAP, but M-Box server configuration and information about shared folders are not stored in LDAP. The given example is for Windows installation of M-Box:

```
<ms_options>
  <config_location>local</config_location>
  <ms_user>pp</ms_user>
  <login_disabled>false</login_disabled>
  <domain>imap.myisp.net</domain>
  <userdir>c:\Mailstore\users</userdir>
  <lmtp_host>mail.example.com</lmtp_host>

  <shared_root name='staff' path='c:\Mailstore\shared\staff'>
    <user name='usera' access='all'/>
    <user name='userb' access='read,write'/>
  </shared_root>

  <sasl>
    <auxprop_plugin>ldapdb</auxprop_plugin>
    <ldapdb_uri>ldap://ldap.myisp.net:19389</ldapdb_uri>
    <ldapdb_dn>cn=DSA Manager, cn=dsa, o=myisp</ldapdb_dn>
    <ldapdb_pw>secret</ldapdb_pw>
    <ldapdb_mech>SIMPLE</ldapdb_mech>

    <saslDCMappingSuffix>ou=users, o=myisp</saslDCMappingSuffix>
    <saslSearchSuffix>ou=users, o=myisp</saslSearchSuffix>
    <saslDefaultDomain>imap.myisp.net</saslDefaultDomain>
  </sasl>
</ms_options>
```

Appendix C Example LDAP Configuration

This appendix shows an example M-Box LDAP configuration in LDIF format.

This section provides examples of M-Box server configuration entry and Shared Folder entry. The two examples show a possible LDAP representation of the XML configuration described in Appendix B, *Example XML Configuration*.

```
dn: cn=mail.myisp.net, cn=Servers, cn=Internet Mailstore, cn=Messaging Config
objectClass: mboxVirtualDomain
objectClass: mboxStoreTailoringObject
objectClass: top
cn: mail.myisp.net
uid: pp
mt-local-domain-site: imap.myisp.net
mboxRootUserDir: c:\Mailstore\users
mboxRootSharedDir: c:\Mailstore\shared
mboxCleartextLoginDisabled: FALSE
mboxListeners: lmtp://mail.example.com:2003

dn: cn=Shared Folders, cn=mail.myisp.net, cn=Servers, cn=Internet Mailstore,
objectClass: untypedObject
objectClass: top
cn: Shared Folders

dn: cn=staff, cn=Shared Folders, cn=mail.myisp.net, cn=Servers, cn=Internet M
objectClass: mboxSharedMailboxRoot
objectClass: top
cn: staff
description: Myisp.net internal shared folder
mailboxRoot: c:\Mailstore\shared\staff
mboxAccessControlList: usera all
mboxAccessControlList: userb read,write
Below you can see an example user entry:
dn: cn=usera, dc=imap, dc=myisp, dc=net, ou=users, o=myisp
objectClass: mboxUser
objectClass: inetUser
objectClass: inetLocalMailRecipient
objectClass: cmuSaslUser

objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
cn: usera
sn: Smith
userPassword: pot4secret
givenName: Jack
uid: usera@imap.myisp.net
displayName: Jack Smith Jr.
mail: usera@imap.myisp.net
mailLocalAddress: usera@imap.myisp.net
mailLocalAddress: jsjr@imap.myisp.net
```

Appendix D TLS Cipher List Formats

This appendix describes the format of the TLS cipher list option and possible ciphers that can be specified.

The cipher list consists of one or more cipher strings separated by spaces. Commas or colons are also acceptable separators.

The actual cipher string can take several different forms.

It can consist of a single cipher suite such as **RC4-SHA**.

It can represent a list of cipher suites containing a certain algorithm, or cipher suites of a certain type. For example **SHA1** represents all ciphers suites using the digest algorithm **SHA1** and **SSLv3** represents all **SSL v3** algorithms.

Lists of cipher suites can be combined in a single cipher string using the + character. This is used as a logical and operation. For example SHA1+DES represents all cipher suites containing the **SHA1** and the **DES** algorithms.

Each cipher string can be optionally preceded by the characters !, - or +.

If ! is used then the ciphers are permanently deleted from the list. The ciphers deleted can never reappear in the list even if they are explicitly stated.

If - is used then the ciphers are deleted from the list, but some or all of the ciphers can be added again by later options.

If + is used then the ciphers are moved to the end of the list. This option doesn't add any new ciphers it just moves matching existing ones.

If none of these characters is present then the string is just interpreted as a list of ciphers to be appended to the current preference list. If the list includes any ciphers already present they will be ignored: that is they will not moved to the end of the list.

Additionally the cipher string @STRENGTH can be used at any point to sort the current cipher list in order of encryption algorithm key length.

D.1 Cipher strings

The following is a list of all permitted cipher strings and their meanings.

DEFAULT
: the default cipher list. This is determined at compile time and is normally ALL:!ADH:RC4+RSA:+SSLv2:@STRENGTH. This must be the first cipher string specified.

COMPLEMENTOFDEFAULT
: the ciphers included in ALL, but not enabled by default. Currently this is ADH. Note that this rule does not cover eNULL, which is not included by ALL (use COMPLEMENTOFALL if necessary).

ALL
: all ciphers suites except the eNULL ciphers which must be explicitly enabled.

COMPLEMENTOFALL
: the cipher suites not enabled by ALL, currently being eNULL.

`HIGH`
: ``high" encryption cipher suites. This currently means those with key lengths larger than 128 bits.

`MEDIUM`
: ``medium" encryption cipher suites, currently those using 128 bit encryption.

`LOW`
: ``low" encryption cipher suites, currently those using 64 or 56 bit encryption algorithms but excluding export cipher suites.

`EXP, EXPORT`
: export encryption algorithms. Including 40 and 56 bits algorithms.

`EXPORT40`
: 40 bit export encryption algorithms

`EXPORT56`
: 56 bit export encryption algorithms.

`eNULL, NULL`
: the ``NULL" ciphers that is those offering no encryption. Because these offer no encryption at all and are a security risk they are disabled unless explicitly included.

`aNULL`
: the cipher suites offering no authentication. This is currently the anonymous DH algorithms. These cipher suites are vulnerable to a ``man in the middle" attack and so their use is normally discouraged.

`kRSA, RSA`
: cipher suites using RSA key exchange.

`kEDH`
: cipher suites using ephemeral DH key agreement.

`kDHr, kDHd`
: cipher suites using DH key agreement and DH certificates signed by CAs with RSA and DSS keys respectively. Not implemented.

`aRSA`
: cipher suites using RSA authentication, i.e. the certificates carry RSA keys.

`aDSS, DSS`
: cipher suites using DSS authentication, i.e. the certificates carry DSS keys.

`aDH`
: cipher suites effectively using DH authentication, i.e. the certificates carry DH keys. Not implemented.

`TLSv1, SSLv3, SSLv2`
: TLS v1.0, SSL v3.0 or SSL v2.0 cipher suites respectively.

`DH`
: cipher suites using DH, including anonymous DH.

`ADH`
: anonymous DH cipher suites.

`AES`
: cipher suites using AES.

`3DES`
: cipher suites using triple DES.

`DES`
: cipher suites using DES (not triple DES).

`RC4`
: cipher suites using RC4.

`RC2`
: cipher suites using RC2.

`MD5`
: cipher suites using MD5.

`SHA1, SHA`
: cipher suites using SHA1.

D.2 Cipher suite names

The following lists give the SSL or TLS cipher suites names from the relevant specification and their OpenSSL equivalents. It should be noted, that several cipher suite names do not include the authentication used, e.g. `DES-CBC3-SHA`. In these cases, RSA authentication is used.

D.2.1 SSL v3.0 cipher suites

SSL_RSA_WITH_NULL_MD5	NULL-MD5
SSL_RSA_WITH_NULL_SHA	NULL-SHA
SSL_RSA_EXPORT_WITH_RC4_40_MD5	EXP-RC4-MD5
SSL_RSA_WITH_RC4_128_MD5	RC4-MD5
SSL_RSA_WITH_RC4_128_SHA	RC4-SHA
SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5	EXP-RC2-CBC-MD5
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA	EXP-DES-CBC-SHA
SSL_RSA_WITH_DES_CBC_SHA	DES-CBC-SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA	DES-CBC3-SHA
SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA	EXP-EDH-DSS-DES-CBC-SHA
SSL_DHE_DSS_WITH_DES_CBC_SHA	EDH-DSS-CBC-SHA
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA	EDH-DSS-DES-CBC3-SHA
SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	EXP-EDH-RSA-DES-CBC-SHA
SSL_DHE_RSA_WITH_DES_CBC_SHA	EDH-RSA-DES-CBC-SHA
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA	EDH-RSA-DES-CBC3-SHA
SSL_DH_anon_EXPORT_WITH_RC4_40_MD5	EXP-ADH-RC4-MD5
SSL_DH_anon_WITH_RC4_128_MD5	ADH-RC4-MD5
SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA	EXP-ADH-DES-CBC-SHA
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA	ADH-DES-CBC3-SHA

D.2.2 TLS v1.0 cipher suites

TLS_RSA_WITH_NULL_MD5	NULL-MD5
TLS_RSA_WITH_NULL_SHA	NULL-SHA
TLS_RSA_EXPORT_WITH_RC4_40_MD5	EXP-RC4-MD5
TLS_RSA_WITH_RC4_128_MD5	RC4-MD5
TLS_RSA_WITH_RC4_128_SHA	RC4-SHA
TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5	EXP-RC2-CBC-MD5
TLS_RSA_EXPORT_WITH_DES40_CBC_SHA	EXP-DES-CBC-SHA
TLS_RSA_WITH_DES_CBC_SHA	DES-CBC-SHA

TLS_RSA_WITH_3DES_EDE_CBC_SHA	DES-CBC3-SHA
TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA	EXP-EDH-DSS-DES-CBC-SHA
TLS_DHE_DSS_WITH_DES_CBC_SHA	EDH-DSS-CBC-SHA
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA	EDH-DSS-DES-CBC3-SHA
TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	EXP-EDH-RSA-DES-CBC-SHA
TLS_DHE_RSA_WITH_DES_CBC_SHA	EDH-RSA-DES-CBC-SHA
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA	EDH-RSA-DES-CBC3-SHA
TLS_DH_anon_EXPORT_WITH_RC4_40_MD5	EXP-ADH-RC4-MD5
TLS_DH_anon_WITH_RC4_128_MD5	ADH-RC4-MD5
TLS_DH_anon_EXPORT_WITH_DES40_CBC_SHA	EXP-ADH-DES-CBC-SHA
TLS_DH_anon_WITH_DES_CBC_SHA	ADH-DES-CBC-SHA
TLS_DH_anon_WITH_3DES_EDE_CBC_SHA	ADH-DES-CBC3-SHA

D.2.3 AES cipher suites from RFC3268, extending TLS v1.0

TLS_RSA_WITH_AES_128_CBC_SHA	AES128-SHA
TLS_RSA_WITH_AES_256_CBC_SHA	AES256-SHA
TLS_DH_DSS_WITH_AES_128_CBC_SHA	DH-DSS-AES128-SHA
TLS_DH_DSS_WITH_AES_256_CBC_SHA	DH-DSS-AES256-SHA
TLS_DH_RSA_WITH_AES_128_CBC_SHA	DH-RSA-AES128-SHA
TLS_DH_RSA_WITH_AES_256_CBC_SHA	DH-RSA-AES256-SHA
TLS_DHE_DSS_WITH_AES_128_CBC_SHA	DHE-DSS-AES128-SHA
TLS_DHE_DSS_WITH_AES_256_CBC_SHA	DHE-DSS-AES256-SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA	DHE-RSA-AES128-SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA	DHE-RSA-AES256-SHA
TLS_DH_anon_WITH_AES_128_CBC_SHA	ADH-AES128-SHA
TLS_DH_anon_WITH_AES_256_CBC_SHA	ADH-AES256-SHA

D.2.4 Additional export 1024 and other cipher suites

Note: These ciphers can also be used in SSL v3.

TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA	EXP1024-DES-CBC-SHA
TLS_RSA_EXPORT1024_WITH_RC4_56_SHA	EXP1024-RC4-SHA
TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA	EXP1024-DHE-DSS-DES-CBC-SHA
TLS_DHE_DSS_EXPORT1024_WITH_RC4_56_SHA	EXP1024-DHE-DSS-RC4-SHA
TLS_DHE_DSS_WITH_RC4_128_SHA	DHE-DSS-RC4-SHA

D.2.5 SSL v2.0 cipher suites

SSL_CK_RC4_128_WITH_MD5	RC4-MD5
SSL_CK_RC4_128_EXPORT40_WITH_MD5	EXP-RC4-MD5
SSL_CK_RC2_128_CBC_WITH_MD5	RC2-MD5
SSL_CK_RC2_128_CBC_EXPORT40_WITH_MD5	EXP-RC2-MD5
SSL_CK_DES_64_CBC_WITH_MD5	DES-CBC-MD5
SSL_CK_DES_192_EDE3_CBC_WITH_MD5	DES-CBC3-MD5

D.3 Examples

All ciphers including NULL ciphers:

```
ALL:eNULL
```

Include all ciphers except NULL and anonymous DH then sort by strength:

```
ALL:!ADH:@STRENGTH
```

Include only 3DES ciphers and then place RSA ciphers last:

```
3DES:+RSA
```

Include all RC4 ciphers but leave out those without authentication:

```
RC4:!COMPLEMENTOFDEFAULT
```

Include all ciphers with RSA authentication but leave out ciphers without encryption.

```
RSA:!COMPLEMENTOFALL
```

Appendix E M-Box Redundancy to LDAP Service Failures

This section talks about how M-Box services can cope with LDAP server outages.

When configured to use LDAP, M-Box can cope with some temporary LDAP server outages. When an M-Box service starts up it needs to successfully read M-Box configuration. Resilience to a failure to connect to the LDAP server during configuration loading is provided using retries, by listing multiple LDAP servers and using configuration caching.

When multiple LDAP servers are specified and the first (the second, etc.) LDAP server is not responding, an M-Box service will automatically try to connect to the second (third, ...) server in the list, unless the connection is successful or there are no more LDAP servers listed. Multiple LDAP servers can be specified by listing several space separated LDAP URLs in the `config_location` option.

An M-Box service will try to connect to the specified LDAP servers one or more times. The number of retries is controlled by the `ldap_connect_retries` option. The delay before different retry attempts can be specified using the `ldap_connect_retry_pause` option.

And finally, an M-Box service will start even if all connection attempts to all LDAP servers fails, as long as there is a valid cached configuration. The cached configuration is created/updated on any successful LDAP configuration loading by any M-Box service and most M-Box utilities. If the M-Box service failed to refresh the LDAP configuration, it will keep trying to reconnect to the LDAP server and refresh the configuration.

Limited caching of user authentication information is provided by userdb_lookup and userdb_cache auxprop plugins.

Appendix F : MPP Protocol Specification

This appendix describes the MPP protocol, designed by Isode, which is used to communicate events between lmtpd, imapd, pop3d and ms_syncd.

Various components of M-Box are using Message Passing Protocol (MPP) for talking to each other. This appendix describes the MPP protocol.

MPP is a simple application level protocol that provides the basis for passing message payloads between two end points. The payload defines the real work to be done. Example of the payload can be XMLRPC, SOAP, or any other data representation formats. MPP doesn't provide any authentication and it doesn't negotiate TLS, but it can be configured over a TLS protected channel with optional certificate based authentication.

Ideas for the protocol design were taken from RFC 3080, RFC 3117, and XMPP, except that the protocol is simpler, because there are no sequence numbers, sliding windows, and it does not force the use of XML and MIME for encoding.

The following protocol requirements were considered while designing MPP:

1. support asynchrony to allow either to process and to receive multiple requests in any order
2. easy to implement in any programming/scripting language
3. don't enforce the use of any data encoding standard
4. easy for the application to decide if it can handle and how to handle any message sent to it
5. easy to parse for applications using nonblocking I/O.

Protocol syntax. Each message send by the client or the server have the same format and include the following components:

Header
: The header consists of three components: an application message type string, message number (`msgno`) and payload size. A single space character (decimal code 32, " ") separates each component. The header is terminated with a CRLF pair. This makes it every easy to parse a message.

Message Type
: The message type is a XML namespace style string which describes the type of message. Format should be `<service>:<payload type>`. Message Type makes it easy for an application to decide if it can and how to handle message. Example. `pcms:auto_dial`.

Message Number
: The message number (`msgno`) must be a non-negative integer (in the range 0..2147483647). It is application defined how the message number should be used. All requests should have a different value from any other outstanding session requests. Both Message Type and Message Number are provided for application use, this allows an application can handle a large number of concurrent requests.

Payload Size
: The payload size (`size`) must be a non-negative integer (in the range 0..2147483647) and specifies the exact number of octets in the payload. (This does not include either the header or the trailer.)

Payload
: The payload is treated as a blob by message parser. The application handles all encoding and decoding of the payload.

Trailer
: `END` CR LF is used to mark the end of the payload. Trailer and Payload Size make very easier to have robust error checking.

Message Syntax using ABNF (RFC 5234):

```
msg = header payload trailer
header = msg_type SP msgno SP size CR LF
msgno = 0..2147483647
size = 0..2147483647
msg_type = 1*<any CHAR except SP / CTL>
payload = *OCTET
trailer = "END" CR LF
```