

Isode M-Link Administration Guide

M-Link and Isode are trade and service marks of Isode Limited.

All products and services mentioned in this document are identified by the trademarks or service marks of their respective companies or organizations, and Isode Limited disclaims any responsibility for specifying which marks are owned by which companies or organizations.

Isode software is © copyright Isode Limited 2002-2010, All rights reserved.

Isode software is a compilation of software of which Isode Limited is either the copyright holder or licensee.

Acquisition and use of this software and related materials for any purpose requires a written licence agreement from Isode Limited, or a written licence from an organization licensed by Isode Limited to grant such a licence.

This manual is © copyright Isode Limited 2010

Chapter 1: Preface	5
1.1 Software Version	5
1.2 Readership	5
1.3 What This Guide Covers	5
1.4 Related Publications	5
1.5 Typographical Conventions	5
1.6 Support Queries and Bug Reporting	6
1.7 Conformance	6
Chapter 2: Introduction to M-Link	7
2.1 Overview	7
2.1.1 Connections to other servers	7
2.1.2 Clustering	7
2.1.3 Multiple "Virtual" domains	7
2.1.4 Chatrooms and multiparty chat sessions	7
2.1.5 Policy-driven Security Labels	7
2.1.6 Management through Ad-Hoc Commands	8
2.2 Configuration Overview	8
Chapter 3: M-Link server Configuration	9
3.1 Initial Configuration	9
3.1.1 M-Link Configuration File	9
3.1.2 M-Link Runtime User	9
3.1.3 M-Link Filestore	10
3.1.4 Installing M-Link license file	10
3.1.5 Configuring LDAP server and configuring M-Link to use LDAP	10
3.1.6 Starting and Stopping M-Link	12
3.1.7 Configuring Internet Messaging Administration Web Application	14
3.1.8 M-Link Users	15
3.2 Server configuration options	15
3.2.1 How M-Link applications read configuration	15
3.2.2 LDAP configuration options	16
3.2.3 Virtual servers and user aliases	19
3.2.4 Specifying service listeners	20
3.2.5 Restricting user access to certain services	21
3.2.6 Other general options	22
3.2.7 XMPP specific options	23
3.2.8 TLS configuration options	26
3.2.9 Security Label support	29
3.2.10 Peer Controls	31
3.2.11 Multiple Domain support	34
3.2.12 Clustering	37
3.2.13 Advanced options controlling M-Link performance	38
3.2.14 SASL options	39
3.3 Logging	39
3.3.1 Getting Started	39
3.3.2 How Logging Works	40
3.4 Upgrading from a previous version	43
Chapter 4: M-Link User management	44

4.1 How M-Link Stores Users.....	44
4.2 User Administration using Internet Messaging Administration Web Application.....	44
4.2.1 Account Name.....	44
4.2.2 User information.....	45
4.2.3 M-Link specific attributes.....	46
4.2.4 User Administration using command line tools.....	48
Chapter 5: User Authentication.....	51
5.1 SASL Authentication.....	51
5.1.1 Generic SASL options.....	51
5.1.2 SASL Mechanisms.....	53
5.1.3 SASL options controlling user management.....	53
5.1.4 Advanced SASL configuration options.....	55
5.2 Userid canonicalization during authentication process.....	58
Chapter 6: Multi-User Chat.....	60
6.1 Persistent Rooms.....	60
6.2 Creating and configuring rooms.....	60
6.2.1 Room Configuration Options.....	60
6.2.2 Affiliations.....	64
Chapter 7: Groups for Rosters and Authorization.....	65
7.1 Overview.....	65
7.1.1 Roster Groups.....	65
7.1.2 Chatroom permissions, etc.....	65
7.2 Types and ObjectClasses.....	65
7.2.1 Common.....	65
7.2.2 M-Link Group.....	66
7.2.3 Isode SASL Group.....	66
7.2.4 Group of Names.....	67
7.2.5 Isode M-Link Group Reference.....	67
7.2.6 Isode M-Link Group Search.....	67
Appendix A: Command line parameters for M-Link service applications	69
Appendix B: Example XML configuration.....	70
Appendix C: Example LDAP configuration.....	71
Appendix D: TLS Cipher List Format.....	73
Appendix E: M-Link redundancy to LDAP service failures.....	80

Chapter 1: Preface

1.1 Software Version

This guide is published in support of Isode M-Link R14.6v0. It may also be pertinent to later releases - please consult the release notes for further details.

1.2 Readership

This guide is intended for administrators who plan to configure M-Link, a high performance Jabber/XMPP server.

1.3 What This Guide Covers

Chapter 2 is a brief introduction to Isode M-Link.

Chapter 3 covers M-Link server configuration.

Chapter 4 describes M-Link user configuration

Chapter 5 describes user authentication

Chapter 6 describes tools that can be used for live M-Link monitoring

1.4 Related Publications

SWADM: Isode M-Switch Administration Guide

VAUDM: Isode M-Vault Administration Guide

1.5 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	Identified by	Example
File and directory names	<i>italic</i>	<i>isoentities</i>
Program and macro names	bold	mkpasswd
Input to the system	Courier	cd newdir
Required user input	<i>italic</i> Courier	<i>filename</i>
Cross references	<i>italic bold</i>	see <i>Section 5.3.2</i>
Additional information to note, or a warning that the system could be damaged by certain action	BOLD CAPS	NOTE: WARNING:

In addition, text which refers to only one platform, UNIX or Windows NT, has markers in the margin indicating the differentiated text.

1.6 Support Queries and Bug Reporting

customer-service@isode.com for all account-related inquiries and issues. 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.

license@isode.com for all licensing related issues.

support@isode.com for all technical inquiries and problem reports, including documentation issues from customers with support contracts. 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.

sales@isode.com for all sales inquiries and similar communication.

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/contact/email-signup.html>

1.7 Conformance

The main specifications Isode M-Link conforms to are listed on <http://www.isode.com/products/m-link-standards.html>

For details on any specific conformance issues, please contact Isode Support.

Chapter 2: Introduction to M-Link

2.1 Overview

M-Link is a standard XMPP server, supporting both a range of Instant Messaging services, PEP (in support of Rich Presence applications), and Multi-User Chat. Finally, it supports hosting of multiple Instant Messaging and Multi-User Chat domains, each with independent configuration.

Like all XMPP servers, externally it listens on two ports, one for clients, and one for servers. It supports multiple connections from each user – typically used to have multiple clients on desktop computers as well as mobile phones.

2.1.1 Connections to other servers

By default, it forms a mesh-like network with peer XMPP services on remote machines. Connections to peer services in general, as well as to specific remote servers, can be controlled in detail via the Peer Controls described in [3.2.10 Peer Controls](#).

2.1.2 Clustering

Multiple M-Link processes can themselves be linked via a mesh networking system, and they will automatically share data inside this network to provide a single service apparently hosted on multiple machines.

All M-Link services are fully clustered, and the removal of a single node will leave all services, and the other nodes, undisturbed.

Configuration of clusters is described in [3.2.12 Clustering](#)

2.1.3 Multiple “Virtual” domains

Any M-Link instance, whether a single server or a cluster, may handle many domains, both for IM and for other services such as Multi-User Chat. This is described in [3.3 Multiple Domain support](#)

2.1.4 Chatrooms and multiparty chat sessions

M-Link supports the configuration of one or more Multi-User Chat service domains, described in [3.3.3 Multi-User Chat Domains](#) and [Chapter 6: Multi-User Chat](#), allowing several users or clients to join a single conversation in a controlled environment.

2.1.5 Policy-driven Security Labels

M-Link supports security labels both for XEP-0258 clients as well as the US CDCIE CCP capable clients such as TransVerse. Security Labels are described in [3.2.9 Security Label support](#) for the entire server, Section for individual

service domains, and [6.2.1 Room Configuration Options](#) for individual Multi-User Chat rooms.

2.1.6 Management through Ad-Hoc Commands

M-Link supports management through XEP-0050, and in particular a large subset of the XEP-0133 commands, available to those in the special “operator” group. The “operator” group is described in [7.2.3 Isode SASL Group](#)

2.2 Configuration Overview

M-Link holds server configuration information, and configuration information on M-Link users and shared folders in an LDAP directory, such as Isode M-Vault.

Most server configuration may be managed using Isode’s Internet Message Administration (IMA) Web interface, which accesses information stored in the directory. This is described in detail in this Manual. Server configuration may also be managed in the directory by use of other tools.

User information may be directly managed using IMA. It may also be loaded into M-Vault from a provisioning system, or managed via Sodium.

Chapter 3:M-Link Server Configuration

M-Link 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, described in [Chapter 4: M-Link User management](#).

The management of server and user data takes place using the Internet Messaging Administration Web Application. Once `(ETCDIR)/ms.conf` is set up to use LDAP configuration as described in [3.1.5.2 Setting up M-Link to use LDAP configuration](#), you can manage M-Link server configuration using the Internet Messaging Administration Web Application. The M-Link server configuration entry specified in the `ldap_basedn` option doesn't have to exist, it will be created by the Internet Messaging Administration Web Application if its parent entry exists in the Directory.

Information may also be managed directly with LDAP. Relevant LDAP attributes are described in [3.2 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-Link server configuration information removing the need for an LDAP directory. Details are given in [3.2 Server configuration options](#).

3.1 Initial Configuration

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

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

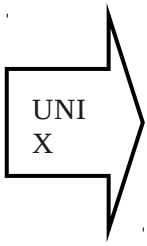
3.1.1 M-Link Configuration File

The default M-Link 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-Link not to use LDAP.

3.1.2 M-Link Runtime User

Create the M-Link 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.





3.1.3 M-Link Filestore

(UNIX only) Create the working directory used by M-Link for storing user's roster, pending subscriptions, trace logs and other information. This directory must be owned by the M-Link runtime user.

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

Create a mail partition owned by the M-Link 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.

3.1.4 Installing M-Link license file

Copy the license file provided to you by Isode to `(ETCDIR)/license.dat`.

3.1.5 Configuring LDAP server and configuring M-Link to use LDAP

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

3.1.5.1 Configuring the LDAP Server (M-Vault)

M-Link uses LDAP for user authentication, and passes XMPP credentials to the Directory. Basic LDAP uses Directory Names for authentication, and so there is a need to map from the XMPP JID to a directory name. Isode's approach is to use SASL (Simple Authentication and Security Layer) authentication with LDAP. With SASL, the XMPP JID (with no resource part) 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, XMPP, 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 6.2.3 of the M-Vault Administrator's 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 DSA's own entry. In order for M-Link to share SASL configuration with M-Vault, the M-Link `dsa_config_dn` option will have to point to this entry, as described in **Section 3.1.5.2**.

Once you have initialized the M-Vault server using EDM, you must configure SASL in M-Vault. Select the DSA properties in EDM and then select the SASL tab:

- Make sure that the Default Domain value is set to the proper value. The default domain is the domain that is appended to an unqualified username, typically it is a fully qualified hostname, e.g. "services.example.com".
- If you want to configure M-Link to bind securely as DSA Manager, you need to set the DSA Manager's username.
- Select the "Generic mappings" tab:
 - Set the mapping rule to "Active Directory compatible".
 - AD DN suffix to be the DN to be appended to DNs of Internet Users in the directory, for example "C=US".
 - AD glue container to be any valid RDN, typically it is "cn=Users" or "ou=Users". This value can be left blank if not needed.
 - Optionally, change the Username attribute to be "CN".

Example: If AD DN suffix is set to "C=US", AD glue container is set to "ou=Users" and Username attribute is "CN", then the entry for M-Link user "fred@example.com" is going to be "CN=fred, ou=Users, dc=example, dc=com, c=US".

3.1.5.2 Setting up M-Link to use LDAP configuration

In order to use M-Link with Isode M-Vault, M-Vault must be configured to use SASL authentication using EDM (see **Section 3.1.5.1**). In order for M-Link to bind securely to M-Vault, you also need to make sure that DSA Manager SASL username is properly set in EDM.

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

1. `config_location` must point to the LDAP server that contains M-Link configuration.
2. One of `ldap_bind_id` or `ldap_bind_dn` must be specified. `ldap_bind_id` is a SASL username for DSA Manager's account, `ldap_bind_dn` is its DN. (Note that if you want to use Internet Messaging Administration Web Application then you must specify `ldap_bind_dn`, as currently the Web Application doesn't support SASL binds to Directory)

3. `ldap_bind_pwd` must contain DSA Manager's password.
4. 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 the Internet Messaging Administration Web Application)
5. `dsa_config_dn` option contains DN of the entry containing SASL configuration for M-Link. Typically this configuration is shared between M-Link and M-Vault (see also **Section 3.1.5.1**), so this option would contain DN of the DSA's own entry.
6. `ldap_basedn` option contains DN of the M-Link configuration entry. This entry contains the entire M-Link 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-Link server entry contains the hostname of the M-Link server.
7. `auxprop_plugin` option located below the `sasl` XML element must contain value `ldapdb`.

Example: If M-Link is responsible for domain "myisp.net" (running on "xmpp.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=dsa, o=myisp, dc=net</dsa_config_dn>
<ldap_basedn>dc=xmpp, 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>
```

3.1.6 Starting and Stopping M-Link

3.1.6.1 M-Link processes

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

Command line parameters are described in 7.2.6.4.

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

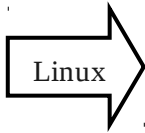
1. `isode.msarchived`

This service is used by M-Link to provide message archiving facilities. It should be started if archiving is required.

2. isode.xmppd

This service implements the XMPP/Jabber server. It should always be running to provide XMPP service.

3.1.6.2 Starting/stopping M-Link on UNIX



This section is specific to Linux.

An example startup/shutdown script, `/opt/isode/sbin/mlink.sh`, is included in the M-Link package.

The script can start, stop and query all of the M-Link services: `isode.xmppd`. A symbolic link "mlink" to the script is created in the rc directory specific to the platform.

For example, on Red Hat Enterprise Linux 3.0 this will be

```
/etc/init.d/mlink
```

In order to start M-Link run

```
/etc/init.d/mlink start
```

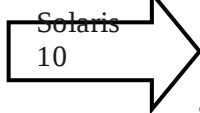
In order to stop it run

```
/etc/init.d/mlink stop
```

To check which M-Link services are running:

```
/etc/init.d/mlink status
```

3.1.6.3 Starting/stopping M-Link on Solaris 10



Solaris 10 uses SVC to manage services. The M-Link package automatically registers its services with SVC. Upon installation of the M-Link package services are marked as *disabled*, i.e. they will not start automatically upon OS start.

In order to enable and start M-Link services run

```
/usr/sbin/svcadm enable xmpp
```

Once enabled, services will be started automatically upon OS start.

In order to start an M-Link service temporarily (i.e. without making the service start automatically when OS starts) use

```
/usr/sbin/svcadm enable -t <service>
```

In order to stop M-Link services run

```
/usr/sbin/svcadm disable -t xmpp
```

In order to stop and disable M-Link services run

```
/usr/sbin/svcdm disable xmpp
```

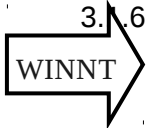
To check which M-Link services are running:

```
svcs -a | grep xmpp
```

To see detailed status of a specific service:

```
svcs -l <service>
```

3.1.6.4 Installing M-Link on Windows



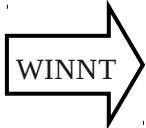
On Windows each M-Link process is installed as a Windows service. These processes run under the Local System account.

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

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

from a command line.

3.1.6.5 Starting/stopping M-Link on Windows



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

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

In order to start M-Link run

```
(SBINDIR)/mbox start
```

In order to stop it run

```
(SBINDIR)/mbox stop
```

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

```
(SBINDIR)/mbox status
```

Use

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

to enable a particular service, for example

```
(SBINDIR)/mbox enable msarchived
```

will change msarchived to Automatic Start.

3.1.7 *Configuring Internet Messaging Administration Web Application*

You must first install and configure the Web Application.

To do so, see the instructions in the Web Applications Guide for installing and configuring Tomcat and the Internet Messaging Administration Web Application.

3.1.8 M-Link Users

Users may be added in several ways. See [Chapter 4: M-Link User management](#) for more details.

3.2 Server configuration options

The M-Link 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.

3.2.1 How M-Link applications read configuration

3.2.1.1 Configuration Location

Option location: “Directory Configuration Browser” page, “General” section.

Description: Specifies where M-Link 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-Link 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>

3.2.1.2 How M-Link 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-Link configuration entry, which contains the entire M-Link configuration, except for information on SASL Configuration, Shared Folders and Users. Most of the attributes map one-to-one to M-Link options (see the subsequent section for the complete list), with the following exceptions:

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

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

The `mboxBackends` and `mboxConnectors` attributes are singlevalued attributes, each containing an URL with hostname/IP address and an optional port number.

Once the M-Link configuration entry (as specified in the “`ldap_basedn`” option) is read, M-Link will try to read DSA’s own entry (as specified in the “`dsa_config_dn`” option), that 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, but authentication will need manual configuration via the `<sasl/>` section.

Finally, M-Link will locate Multi-User Chat domains below the “`ldap_basedn`”, and beneath each, locate persistent chatrooms, which may be preconfigured by Sodium, or setup via XMPP. Future versions will use the information in these entries to configure complete Multidomain support.

3.2.2 LDAP configuration options

This section describes configuration options that are needed to connect to an LDAP server that contains M-Link 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-Link 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 that if an LDAP configuration is in force all options specified in Section 3.2 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-Link service or application is starting up, the error message is printed to `stderr` and the service/application will refuse to run.

3.2.2.1 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>`

3.2.2.2 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>`

3.2.2.3 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>`

3.2.2.4 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>`

3.2.2.5 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).

Default value: -None-

Example: supersecret

XML option name: *ldap_bind_pwd*

Parent XML element: top level <ms_options>

Note that 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.

3.2.2.6 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>

3.2.2.7 M-Link configuration entry DN

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

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

Default value: -None-

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

XML option name: *ldap_basedn*

Parent XML element: top level <ms_options>

3.2.2.8 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-Link users in the Directory. If M-Link server configuration is stored in LDAP, this entry would be read together with M-Link server configuration.

Default value: -None-

Example: cn=dsa, c=US

XML option name: *dsa_config_dn*

Parent XML element: top level <ms_options>

3.2.2.9 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>

3.2.2.10 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>

3.2.3 Virtual servers and user aliases

M-Link 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, and is automatically added to the multidomain configuration – see Section Error: Reference source not found

3.2.4 Specifying service listeners

3.2.4.1 Listen URLs

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

Description: This option describes XMPP and *mseventd* listeners. The option is multivalued, each value is an 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
xmpp	A XMPP listener.
xmpp-server	XMPP server-to-server listener.
isode.event	A mseventd listener.
isode.log	A mlogd listener.

Note 1: only the hostname/IP address and the port number parts of an URL are considered, the remainder of the URL is ignored.

Note 2: 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
<i>xmpp</i>	<i>xmpp_client_host</i>	empty (*) - none?	<i>xmpp_client_port</i>	5222	
<i>xmpp-server</i>	<i>xmpp_server_host</i>		<i>xmpp_server_port</i>	5269	
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/mlogd 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): *xmpp://mail.example.net*

xmpp-server://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>

3.2.5 Restricting user access to certain services

M-Link 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 XMPP access and allow it on per user basis.

The following 2 options control access to different services.

The screenshot shows a configuration panel with two rows. The first row is labeled 'Accessible services' and has a text input field to its right. The second row is labeled 'Default new service access' and has three radio button options: 'Granted' (which is selected), 'Forbidden', and 'Unset (ms.conf: Granted)'. Each label has a small question mark icon to its right.

3.2.5.1 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-Link using XMPP, and other Isode services.

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 **xmpp**. Services not explicitly listed in this list are controlled by the “Default new service access” option described below, although each domain may have an explicit access list, see *Section Error: Reference source not found*.

Default value: empty string

Example: xmpp=grant

LDAP attribute name: *AccessibleServices*

XML option name: *accessible_services*

Parent XML element: top level <ms_options>

3.2.5.2 Default new service access

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

Description: This boolean option controls default access to all services not listed in the “Accessible services” option. The value “true” means that the access is granted, the value “false” means that the access is forbidden.

See *Section 4.2.3.2* on how to specify accessible services on per-user basis.

Default value: true

Example: false

LDAP attribute name: *defaultNewServiceAccess*

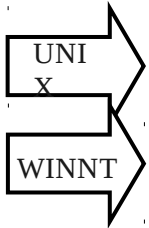
XML option name: *default_service_access*

Parent XML element: top level <ms_options>

3.2.6 Other general options

3.2.6.1 Runtime user ID

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



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

(Windows) This option is ignored on Windows.

Default value: mbox

Example: xmppsrv

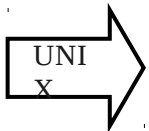
LDAP attribute name: *uid*

XML option name: *ms_user*

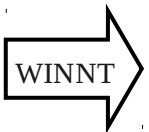
Parent XML element: top level <ms_options>

3.2.6.2 Management of M-Link services

Option location: This option is not accessible through IMA.



(UNIX) Description: Specifies whether the M-Link services are to be managed internally. When managed internally, each M-Link 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 or Solaris svc..



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

Default value (Linux): true

Default value (Solaris): false

Example: false

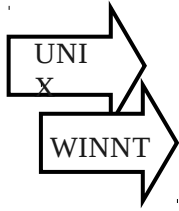
LDAP attribute name: *-None-*

XML option name: *managed_services*

Parent XML element: top level <ms_options>

3.2.6.3 Users root directory

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



Description: Specifies the default location for user mail. Each user's mail will be located in a subdirectory of this directory, named after the user.

(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>

3.2.7 XMPP specific options

3.2.7.1 Clear text Login Disabled

Description: If this option is set to true, it disables all commands that send password in the clear to the server. This includes plaintext authentication using XEP 0078 and SASL PLAIN and LOGIN authentication mechanism. Note that if TLS encryption is active then the SASL PLAIN mechanism and/or XEP-0078 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>

3.2.7.2 Auto Accept

Description: If this option is set to true, then by default allsubscription requests between users of the same domain will be automatically accepted, without passing the request onto the user.

This setting may be overridden by the user's entry, see Section 4.2.3.4.

Default value: false

Example: true

LDAP attribute name: N/A

XML option name: *auto_accept_default*

Parent XML element: top level <ms_options>

3.2.7.3 Auto Subscribe

Description: If this option is set to true, then by default all subscription requests between users of the same domain will result in an automatic mutual subscription request.

This setting may be overridden by the user's entry, see Section 4.2.3.5.

Default value: false

Example: true

LDAP attribute name: N/A

XML option name: *auto_subscribe_default*

Parent XML element: top level <ms_options>

3.2.7.4 Offline Messaging

Description: When this option is set to true, then offline messaging is enabled across the server. Whether a message to an individual user who is offline will be held until the user is online depends on several settings. Firstly, this setting must be on. Secondly, the user must have offline messaging enabled, either by the default, or by a user specific setting. Thirdly, the user's offline messaging store must contain fewer than the maximum number of offline messages.

See Section 4.2.3.6 for the user settings relating to this option.

Default value: true

Example: false

LDAP attribute name: *isodeMlinkOffline*

XML option name: *offline_messaging*

Parent XML element: top level <ms_options>

3.2.7.5 Offline Message Limit

Description: This option defines the maximum number of offline messages the server will store for a user.

Default value: 100

Example: 2500

LDAP attribute name: *isodeMlinkOfflineMax*

XML option name: *offline_message_max*

Parent XML element: top level <ms_options>

3.2.7.6 Offline Message Default

Description: This option defines the default for a user's offline messaging behaviour.

It can be overridden by user settings, see Section 4.2.3.6.

Default value: false

Example: true

LDAP attribute name: N/A

XML option name: *offline_messaging_default*

Parent XML element: top level <ms_options>

3.2.7.7 Password Changing

Description: This option can be used to disable XEP-0077 based password changing. By default, passwords may be changed, bypassing any password policy, but only over an encrypted session.

Default value: true

Example: false

LDAP attribute name: N/A

XML option name: *xep_77_change_password*

Parent XML element: top level <ms_options>

3.2.7.8 Password Changing SSF

Description: This option can be used to allow XEP-0077 based password changing over insecure channels. By default, an SSF of 64 is used, which forces export-grade encryption.

Default value: 64

Example: 0

LDAP attribute name: N/A

XML option name: *xep_77_ssf*

Parent XML element: top level <ms_options>

3.2.8 **TLS configuration options**

The following options appear only in the Advanced mode. They control if XMPP TLS is available. Note that the default configuration has no certificates, nor anonymous ciphers, and therefore TLS will not be enabled.

3.2.8.1 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 7.2.6.4 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>

3.2.8.2 CA File (PEM)

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

Description: Specifies path to a PEM file containing one or more CA certificate 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 xmppd startup. If this option is not set, this might result in server's inability to verify client certificates, however TLS will still be able to provide data encryption.

See also 3.2.8.4.

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>

3.2.8.3 CA Path

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

Description: Specifies a directory where multiple PEM files containing "trusted" CA certificates could 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>

3.2.8.4 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 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.

Note that the .p12 file may also contain CA certificates.

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>

3.2.8.5 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>

3.2.8.6 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.

Default value: -empty string-

Example: SuperS0cret-Password

LDAP attribute name: *mbxTlsKeyPassword*

XML option name: *tls_key_password*

Parent XML element: top level <ms_options>

3.2.8.7 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: *mbxTlsRequireClientCert*

XML option name: *tls_require_client_cert*

Parent XML element: top level <ms_options>

3.2.8.8 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: *mbxTlsVerifyDepth*

XML option name: *tls_verify_depth*

Parent XML element: top level <ms_options>

3.2.8.9 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. It is recommended that this option is left at the default value of false unless specifically required. Cases where this is required is for open federation of XMPP servers.

Default value: false

Example: true

LDAP attribute name: -None-

XML option name: *tls_force_verify*

Parent XML element: *top level <ms_options>*

3.2.9 Security Label support

M-Link supports both the newer, XSF standards-track XEP-0258, and the older form of security labels used by the CDCIE system, including the TransVerse client. Setting up both requires a number of items to be configured. A policy is required for the server to evaluate labels and clearances, and a server itself will most likely need a clearance, and may be labelled.

Applying a clearance to a server (or service, see below) enables it to handle labelled information. Typically, clearances are used to prevent messages labelled with a high sensitivity label from being accepted by the server, and to restrict a “top secret” server to only handling top secret data.

Applying a label to a server restricts those who can connect to those with sufficient clearance.

Catalogues are used to provide a working set of useful labelling options to XEP-0258 capable clients which are not policy-aware. If no catalogue is provided, then clients will be unable to label messages unless they are policy-aware – typical clients are not.

Finally, the CDCIE method requires an additional mapping to allow clients such as the TransVerse client to specify labels by reference. This is only required to support CDCIE clients such as TransVerse.

All Security Label facilities are configured either by referencing an XML file containing the label, clearance, SPIF, or other Security Information Object (SIO) in the XML configuration file, or by entering the XML data describing these object directly into the LDAP attributes in the appropriate directory entry.

3.2.9.1 Security Policy

Description: This option defines the Security Policy of the server. Any use of labelling, whether XEP-0258 or CDCIE, will require a SPIF to be entered or uploaded.

Default value: -None-

Example: */etc/isode/sio/policy.xml*

XML option name: *sio_policy_file*

Parent XML element: top level <ms_options>

LDAP attribute name: sioPolicy

LDAP attribute value syntax: SIO (XML) SPIF

3.2.9.2 Security Label Catalog

Description: This option specifies the Security Label Catalog for the server.

Default value: -None-

Example: /etc/isode/sio/label_catalog.xml

XML option name: *sio_label_catalog_file*

Parent XML element: top level <ms_options>

LDAP attribute name: sioLabelCatalog

LDAP attribute value syntax: SIO Label Catalog

3.2.9.3 XEP 258 Security Label Catalog

Description: This option specifies the Security Label Catalog for the server. This option is an alternative to the SIO label catalog option above.

Default value: -None-

Example: /etc/isode/sio/xep258_label_catalog.xml

XML option name: *xep258_label_catalog_file*

Parent XML element: top level <ms_options>

LDAP attribute name: xep258LabelCatalog

LDAP attribute value syntax: XEP 258 Label Catalog

3.2.9.4 XEP 258 Security Label Format

Description: This option specifies the default encoding to use when generating XEP 258 labels. Two choices are supported: xep258ess (ESS) and isode (Isode XML).

Default value: ESS

Example: Isode

XML option name: *xep258_format*

Parent XML element: top level <ms_options>

LDAP attribute name: isodeMlinkXep258Format

LDAP attribute value syntax: string (“xep258ess” or “isode”)

3.2.9.5 FLOT Injection

Description: This option enables/disables First-Line-Of-Text (FLOT) injection in stanzas sent to clients which do not advertise support for XEP 258 (or CDCIE CCP).

Default value: TRUE

Example: FALSE

XML option name: *flot*

Parent XML element: top level <ms_options>

LDAP attribute name: isodeMlinkFLOT

LDAP attribute value syntax: boolean

3.2.9.6 Security Label

Description: This option defines the Security Label of the server. This controls who may connect to the server.

Default value: -None-

Example: /etc/isode/sio/lConfidential.xml

XML option name: *sio_label_file*

Parent XML element: top level <ms_options>

LDAP attribute name: sioLabel

LDAP attribute value syntax: SIO (XML) Security Label

3.2.9.7 Security Clearance

Description: This option defines the Security Clearance of the server, which controls what messages may be processed by the server.

Default Value: -None-

Example: /etc/isode/sio/cRestricted.xml

XML option name: *sio_clearance_file*

Parent XML element: top level <ms_options>

LDAP attribute name: sioClearance

LDAP attribute value syntax: SIO (XML) ClearanceSecurity Label Catalogue

Description: This option specifies a catalogue of security labels, which the server will then adapt, depending on the user requesting, and the target they specify.

Default Value: -None-

Example: /etc/isode/sio/slcBasic.xml

XML option name: *sio_label_catalog_file*

Parent XML element: top level <ms_options>

LDAP attribute name: sioLabelCatalog

LDAP attribute value syntax: SIO (XML) Label Catalog

3.2.9.8 Default Security Label

Description: This option defines the Default Security Label of the server. This label is used instead of the SPIF default label in cases where the stanza does not contain a security label.

Default value: -None-

Example: /etc/isode/sio/lConfidential.xml

XML option name: *sio_label_file*

Parent XML element: top level <ms_options>

LDAP attribute name: sioLabel

LDAP attribute value syntax: SIO (XML) Security Label

3.2.9.9 CDCIE CCP

Description: This option enables/disables enables transparent support for locally connected CDCIE CCP clients such as TransVerse.

Default value: TRUE

Example: FALSE

XML option name: *cdcie_ccp*

Parent XML element: top level <ms_options>

LDAP attribute name: isodeMlinkCdcieCcp

LDAP attribute value syntax: boolean

3.2.9.10 CDCIE Map

Description: This option enables/disables specifies a mapping between CDCIE CCP labels and XEP 258 labels used to support remote systems using CDCIE CPP labels.

Default value: - none-

Example: /etc/isode/sio/cdciecpp_map.xml

XML option name: *cdcie_map*

Parent XML element: top level <ms_options>

LDAP attribute name: isodeMlinkCdcieMap

LDAP attribute value syntax: Isode (XML) CDCIE CPP Map

3.2.10 Peer Controls

Peer controls allows for processing to applied on a per remote XMPP domain basis. For instance, one can use peer controls to disallow XMPP communication with a particular XMPP domain.

Peering controls may be contained in either the configuration file or the directory.

Configuration Example:

```
<peering>
    <peer domain='abusive.example'></deny></peer>
</peering>
```

In the directory, the peer controls are stored in a container, generally named cn=Peers, with each peer control stored in a isodeMlinkPeer object.

Directory Configuration Example:

```
dn: cn=abusive.example,cn=Peers,o=example.net
objectClass: isodeMlinkPeer
cn: abusive.example
isodeMlinkPeerDeny: TRUE
```

All the controls for a particular peer must reside in a single configuration or directory entry.

A simple wildcarding mechanism is supported.

- A domain value of *.example.com applies to all subdomains of example.com but not to example.com itself.
- A domain value of +.example.com applies to example.com and all of its subdomains.
- Default peer controls may be specified with a <peer/> configuration entry with no domain= attribute or with a 'cn=*' peer control directory object.

3.2.10.1 Deny Control

Description: Disallow all XMPP traffic to peer. Deny connection attempts from peer.

Default Value: False

XML option name: `<deny/>`

Parent XML element: `<peer>`

LDAP attribute name: `isodeMlinkPeerDeny`

LDAP attribute value syntax: `BOOLEAN`

3.2.10.2 Accept From Control

Description: Refuse connections from domain unless made from specified address.

Default Value: Absent

Example:

```
<accept address="10.0.0.1"/>
```

XML option name: `<accept/>`

Parent XML element: `<peer>`

LDAP attribute name: `isodeMlinkPeerAcceptFrom`

3.2.10.3 Connect To Host/Link Control

Description: Route traffic for peer to specified host or link.

Default Value: Absent

Example:

```
<connect host="edge.example.com"/>
```

or:

```
<connect link="link name"/>
```

XML option name: `<connect/>`

Parent XML element: `<peer>`

LDAP attribute name: `isodeMlinkPeerConnectTo`

LDAP attribute value syntax: `host name`

LDAP attribute name: `isodeMlinkPeerConnectLink`

LDAP attribute value syntax: `link name`

3.2.10.4 Message Folding

Description: Specifies `<message/>` folding. The mode may be either “keep” or “strip”. If “keep”, the match elements specify which elements to pass to the remote server. If “strip”, the match elements specify which elements to remove before it is passed to the remote server. Additionally, required elements may be

specified. If a folded stanza does not contain one of required elements, the whole stanza will be discarded.

Default Value: Absent

Example:

```
<message_fold mode="strip">
  <require element="body"/>
  <strip element="html"
ns="http://jabber.org/protocol/xhtml-im"/>
</message_fold>
```

XML option name: `<message_fold/>`

Parent XML element: `<peer>`

LDAP attribute name: `isodeMlinkPeerMessageFoldMode`

LDAP attribute value syntax: string (“keep” or “strip”)

LDAP attribute name: `isodeMlinkPeerMessageFoldRequire`

LDAP attribute value syntax: element-name name-space pair

LDAP attribute name: `isodeMlinkPeerMessageFoldMatch`

LDAP attribute value syntax: element-name name-space pair

Example:

```
isodeMlinkPeerMessageFoldMode: strip
isodeMlinkPeerMessageFoldRequire: body
isodeMlinkPeerMessageFoldMatch: html
http://jabber.org/protocol/xhtml-im
```

3.2.10.5 Presence Folding

Description: Specifies `<presence/>` folding. The mode may be either “keep” or “strip”. If “keep”, the match elements specify which elements to passed to the remote server. If “strip”, the match elements specify which elements to remove before it is passed to the remote server.

Default Value: Absent

Example:

```
<presence_fold mode="keep">
  <match element="show"/>
  <match element="priority"/>
  <match element="c" ns="http://example"/>
</presence_fold>
```

XML option name: `<presence_fold/>`

Parent XML element: `<peer>`

LDAP attribute name: isodeMlinkPeerPresenceFoldMode

LDAP attribute value syntax: string (“keep” or “strip”)

LDAP attribute name: isodeMlinkPeerPresenceFoldMatch

LDAP attribute value syntax: element-name name-space pair

Example:

```
isodeMlinkPeerPresenceFoldMode: keep
isodeMlinkPeerPresenceFoldMatch: show
isodeMlinkPeerPresenceFoldMatch: priority
isodeMlinkPeerPresenceFoldMatch: c http://example
```

Relay Control

Description: Specifies relay zones for controlling relaying. A stanza will only be forwarded between two remote systems if those two systems are in different zones.

Default Value: -none-

Configuration Example: <relay zone="Example"/>

XML option name: <relay/>

Parent XML element: <peer>

LDAP attribute name: isodeMlinkPeerRelay

LDAP attribute value: string

3.2.10.6 Clearance Control

Description: Specifies peer clearance for security label checks

Default Value: security policy default

Configuration Example: /etc/isode/sio/cRestricted.xml

XML option name: <sio_clearance_file/>

Parent XML element: <peer>

LDAP attribute name: sioClearance

LDAP attribute value: SIO (XML) clearance

3.2.10.7 Default Label Control

Description: Specifies peer default label for security label checks

Default Value: security policy default

Configuration Example: /etc/isode/sio/lRestricted.xml

XML option name: <sio_clearance_file/>

Parent XML element: <peer>

LDAP attribute name: sioDefaultLabel

LDAP attribute value: SIO (XML) label

3.2.10.8 Relabel Out XEP 258

Description: Disables inclusion of XEP 258 security labels in outbound stanzas.

Configuration Example: <no_xep258/>

XML option name: <no_xep258/>

Parent XML element: <relabel_out> of peer

LDAP attribute name: isodeMlinkPeerRelabelOutXep258

LDAP attribute value: BOOLEAN (default TRUE, set to FALSE to disable XEP 258 security label inclusion)

3.2.10.9 Relabel Out XEP 258 Format

Description: This option specifies the default encoding to use when generating XEP 258 labels. Two choices are supported: xep258ess (ESS) and isode (Isode XML).

Default value: ESS

Example: Isode

XML option name: *xep258_format*

Parent XML element: <relabel_out> of peer

LDAP attribute name: isodeMlinkPeerRelabelOutXep258Format

LDAP attribute value syntax: string (“xep258ess” or “isode”)

3.2.10.10 Relabel Out Provide Default

Description: Enables insertion of a default security label in stanzas which have no security label.

Configuration Example: <provide_default/>

XML option name: <provide_default/>

Parent XML element: <relabel_out> of peer

LDAP attribute name: isodeMlinkPeerRelabelOutProvideDefault

LDAP attribute value syntax: BOOLEAN (default FALSE, set to TRUE to enable insertion of a default security label)

3.2.10.11 Relabel Out Use Equivs

Description: Enables replacement of security label based upon security policy equivalences of a default security label in stanzas which have no security label.

Configuration Example: <provide_default/>

XML option name: <provide_default/>

Parent XML element: <relabel_out> of peer

LDAP attribute name: isodeMlinkPeerRelabelOutProvideDefault

LDAP attribute value syntax: BOOLEAN (default FALSE, set to TRUE to enable insertion of a default security label)

3.2.10.12 Relabel Out SIO Policy

Description: Provides peer's policy for use. Used with provide default, use eqivs, and update controls.

Default value: -None-

Example: /etc/isode/sio/policy.xml

XML option name: *sio_policy_file*

Parent XML element: <relabel_out> of peer

LDAP attribute name: isodeMlinkRelabelOutSioPolicy

LDAP attribute value syntax: SIO (XML) SPIF

3.2.10.13 Relabel Out SIO Label

Description: If set, relabel to the specified label.

Default value: -None-

Example: /etc/isode/sio/lSecret.xml

XML option name: *sio_label_file*

Parent XML element: <relabel_out> of peer

LDAP attribute name: isodeMlinkRelabelOutSioLabel

LDAP attribute value syntax: SIO (XML) SPIF

3.2.10.14 Relabel Out XEP 131 Classification

Description: This option enables sending of XEP 131 Classification elements to the remote server.

XML option name: *xep131_classification*

Parent XML element: <relabel_out> of peer

LDAP attribute name: isodeMlinkPeerRelabelOutXep313Cls

LDAP attribute value syntax: boolean (Default 'FALSE', set to 'TRUE' to enable)

3.2.10.15 Relabel Out FLOT

Description: This option enables sending of First-Line-Of-Text (FLOT) labels to the remote server.

XML option name: *flot*

Parent XML element: <relabel_out> of peer

LDAP attribute name: isodeMlinkPeerRelabelOutFlot

LDAP attribute value syntax: boolean (Default 'FALSE', set to 'TRUE' to enable)

3.2.10.16 CDCIE CCP

Description: This option enables/disables sending of CDCIE CPP labels to the remote server.

Default value: FALSE

Example: TRUE

XML option name: *cdcie_ccp*

Parent XML element: <relabel_out> of peer

LDAP attribute name: isodeMlinkPeerRelabelOutCdcieCcp

LDAP attribute value syntax: boolean

3.2.10.17 CDCIE Map

Description: This option specifies a CDCIE Map to use for outbound relabeling.

Default value: - none-

Example: /etc/isode/sio/cdciecpp_map.xml

XML option name: *cdcie_map*

Parent XML element: <relabel_out> of peer

LDAP attribute name: isodeMlinkPeerRelabelOutCdcieMap

LDAP attribute value syntax: Isode (XML) CDCIE CPP Map

3.2.11 Links

Peers control, via the <connect/> element described above, may specify that traffic to/from a set of peers is to be sent over a particular I/O link instead of

using XMPP Server-to-server facilities. Link elements, like peer, elements placed in the <peering/> element or, if directory configuration is used, under the cn=Peers entry.

Configuration Example:

```
<peering>
    <peer domain='+.special.example'><connect
link='special'></peer>
    <link name='special' type='direct_io'>
        <connect_to host='special.example.com' port='9999'>
        <listen_on port='9999'>
    </link>
</peering>
```

In the directory, the links are stored in a container, generally named cn=Peers, with each peer control stored in a isodeMlinkLink object.

Directory Configuration Example:

```
dn: cn=\+.special.example,cn=Peers,o=example.net
objectClass: isodeMlinkPeer
cn:+.special.example
isodeMlinkPeerConnectLink: special

dn: cn=special,cn=Peers,o=example.net
objectClass: isodeMlinkLink
cn: special
isodeMlinkLinkType: direct_io
isodeMlinkLinkConnectTo: special.example.com:9999
isodeMlinkLinkListenOn: :9999
```

Multiple peer controls may reference a single link object.

The type must be either 'direct_io', 'connection_manager', 'stanza_pipe', or 's5066'.

3.2.11.1 Connect To Information

Description: Specifies the host and port to connect to.

Default Value: Absent

Example:

```
<connect_to host="remote.example.com" port="9999"/>
```

XML option name: `<connect_to/>`

Parent XML element: `<Link>`

LDAP attribute name: `isodeMlinkLinkConnectTo`

LDAP attribute value syntax: string of format `host:port`

3.2.11.2 Listen On Information

Description: Specifies the host and port to listen on. The host field may be absent.

Default Value: Absent

Example:

```
<listen_on host="example.com" port="9999"/>
```

XML option name: `<listen_on/>`

Parent XML element: `<link>`

LDAP attribute name: `isodeMlinkLinkListenOn`

LDAP attribute value syntax: string of format `host:port` with the host part optional.

3.2.12 Clustering

Clustering, in this instance, is connecting multiple M-Link processes together to run a single domain or set of domains. Each process within the cluster is known as a “node”. In M-Link’s clustering implementation, each node is made aware through configuration of the addresses of all its peers.

Nodes can communicate using a variety of encodings, and a variety of transports. These are specified using a URI-like syntax. Current transports are TCP and UDP, and the only encoding is TLV.

The URI’s scheme is made up of “tcp” or “udp”, a period, and “tlv”.

3.2.12.1 Specifying the Node URI

Description: All nodes must have a node URI configured for their own local address, as well as all peers URIs, configured as below.

Default Value: -None-

Example: `tlvmpp.tcp://192.168.37.212:3999/`

LDAP attribute name: -None-

XML option name: `cell_xmpp_uri`

Parent XML element: top level `<ms_options>`

3.2.12.2 Specifying Peer URIs

Description: Each node within the cluster is required to know the locations of all peers. This option should appear as many times as there are peers.

Default Value: -None-

Example: `tlvmpp.tcp://192.168.37.212:3999/`

LDAP attribute name: -None-

XML option name: `cell_xmpp`

Parent XML element: top level `<ms_options>`

3.2.12.3 Specifying Dialback Secret

Description: All nodes within the cluster should share a common dialback secret. A randomly generated password is best for this.

Default Value: -None-

Example: `d141b4ck-s3cr3t`

LDAP attribute name: -None-

XML option name: `xmpp_db_secret`

Parent XML element: top level `<ms_options>`

Other options controlling M-Link directories

The following options are only available in WebAdmin Advanced mode. They specify different directories used by M-Link internally, in most cases those values should not be changed.

3.2.12.4 Run

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

Description: Specifies the runtime directory for M-Link 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>`

3.2.12.5 Telemetry log directory

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

Description: Specifies top level directory for XMPP trace logging. Trace logging for a user "jane" can be enabled by creating a "<telemetry_log>/jane" directory. Each XMPP 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: -- Disabled --

Example: /var/imap/tracelogs

LDAP attribute name: mboxTelemetryLogDir

XML option name: telemetry_log

Parent XML element: top level <ms_options>

3.2.13 Advanced options controlling M-Link performance

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

3.2.13.1 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 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: -None-

Example: 4

LDAP attribute name: -None-

XML option name: thread_pool_min

Parent XML element: top level <ms_options>

3.2.13.2 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>*

3.2.14 SASL options

SASL options are described in Section 5

3.3 Multiple Domain support

3.3.1 Overview

In XMPP, domains are used to identify both instant messaging services, such as the “example.com” in “joe@example.com”, and other services, such as Multi-User Chat. Each instant messaging domain typically has at least one Multi-User Chat which is considered local to, or subordinate to, that service, and clients may discover this hierarchy by querying the server using XEP-0030, known as Service Discovery or simply “Disco”.

Domain objects are located in the directory below the M-Link entry itself, using the “Isode M-Link Instant Messaging Service” Sodium template, which will create an object of class “mlinkIMDomain”. As with the file-based configuration below, the default domain is automatically assumed to exist as an IM domain.

Below these objects in turn can be placed objects for Multi-User Chat domains and XEP-0114 components, as detailed below.

If using file-based configuration, each domain must be mentioned in a section of the configuration file represented by the `<multidomain/>` element. The default domain is automatically added to this section, so single domain deployments can omit this. Each Instant Messaging domain should be placed immediately beneath the `<multidomain/>` element, in a `<domain/>` element, with a name attribute defining the domain name – for example, `<domain name='example.com'/>`.

Multi-User Chat domains, and XEP-0114 components – also identified by domain names – are placed immediately below a `<domain/>`, and themselves can contain various options.

3.3.2 Common options

All types of domains may contain a number of configuration settings:

3.3.2.1 Security Label

Description: This option defines the Security Label of the domain. This controls who may see and use the service. It should be set to the filename of the XML-based security label.

Default value: -None-

Example: /etc/isode/sio/lConfidential.xml

XML option name: *sio_label_file*

Parent XML element: top level <ms_options><multidomain><domain>

LDAP attribute name: sioLabel

LDAP attribute value syntax: SIO (XML) Security Label

3.3.2.2 Security Clearance

Description: This option defines the Security Clearance of the domain, which controls what messages may be processed by the service. It should have a value of the filename of an XML-based clearance file.

Default Value: -None-

Example: /etc/isode/sio/cRestricted.xml

XML option name: *sio_clearance_file*

Parent XML element: top level <ms_options><multidomain><domain>

LDAP attribute name: sioClearance

LDAP attribute value syntax: SIO (XML) Clearance

3.3.2.3 TLS Certificate File

Description: This option defines TLS Certificate file used by the domain. If left out completely, the TLS certificate of the parent domain will be used (if a parent domain exists), or else the global TLS certificate will be used. However, if present, but empty – such as the construct <tls_cert_file/> - then no TLS certificate will be used.

Default Value: -None-

Example: /etc/isode/extra-domain.p12

LDAP attribute name: mboxTlsCertFile

XML option name: *tls_cert_file*

Parent XML element: top level <ms_options><multidomain><domain>

3.3.2.4 TLS Key File

Description: This option defines TLS Key file used by the domain, if any.

Default Value: -None-

Example: /etc/isode/extra-domain.key

LDAP attribute name: mboxTlsKeyFile

XML option name: *tls_key_file*

Parent XML element: top level <ms_options><multidomain><domain>

3.3.2.5 TLS Password File

Description: This option defines the password for the private key, or PKCS#12 file used. Note that this is primarily in support of PKCS#12, which mandates a password, and is not for security purposes.

Default Value: -None-

Example: s3cret!

LDAP attribute name: mboxTlsKeyPassword

XML option name: *tls_key_password*

Parent XML element: top level <ms_options><multidomain><domain>

3.3.3 Multi-User Chat Domains

Multi-User Chat Domains should be added beneath the holding domain, using the “Isode M-Link Multi-User Chat Domain” Sodium template, which creates entries with an objectClass ofmlinkMUCDomain.

In a file based configuration, Multi-User chat domains should be specified beneath a holding <domain/>, using a <muc_domain/> element. Like the <domain/> element, this must have a name attribute declaring the domain name, and may contain the configuration settings above.

3.3.4 Components

Components should be added beneath the holding domain, using the “Isode M-Link Component Domain” Sodium template, which creates entries with an objectClass ofmlinkComponentDomain.

Components may alternately be specified beneath a holding <domain/>, using a <component/> element. Again, as with the <domain/> element, this must have a name attribute declaring the domain name, and may contain any of the common configuration elements above.

Components also have specific configuration elements – in particular, they require a component key set for XEP-0114 access, and some components are required to be trusted by the server.

3.3.4.1 Component Key

Description: This option defines the key, or password, for XEP-0114 component authentication.

Default Value: -None-

Example: s3cret!

LDAP attribute name: isodeMlinkComponentKey

XML option name: *component_key*

Parent XML element: top level

```
<ms_options><multidomain><domain><component>
```

3.3.4.2 Trusted Components

Description: This boolean option, if set to true, allows components to send stanzas with the from attribute set to a different, local, domain.

Default Value: False

Example: TRUE

LDAP attribute name: isodeMlinkComponentTrusted

XML option name: *component_trusted*

Parent XML element: top level

```
<ms_options><multidomain><domain><component>
```

3.3.5 **Group Disco Service**

In order to explore groups, it may be useful to tell M-Link to provide a service which can be used with an XMPP client supporting Service Discovery to view group information. This can be created using the “Isode M-Link Groups Domain” template, which will create an entry with an object class of `mLinkGroupsDomain`.

Alternately, in a file-based configuration, the domain must be specified with a `<groups name='..'/>` element beneath the holding domain.

In either case, the domain used must match the `<xmpp_group_domain/>` directive set in the `ms.conf` file.

3.3.6 **Publish Subscribe Service (Future)**

Isode M-Link does not yet support a full PubSub service, however configuration for one is possible. The resultant service is incomplete and not supported.

3.4 Logging

3.4.1 Getting Started

By default M-Link sends log messages to the "mail" syslog facility. The severity levels used by M-Link 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-Link application, you should do the following:

Copy the file *mboxtailor.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, you should ensure that **/opt/isode/bin/tixwish** is in your path (or define `TIXWISH=/opt/isode/bin/tixwish`) and then run **/opt/isode/sbin/logconfig**.

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

3.4.2 How Logging Works

3.4.2.1 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-Link processes don't generate any Audit events.

3.4.2.2 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.

3.4.2.3 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-Link, this is contained in the *mboxtailor.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.

3.4.2.4 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**

- **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:

1. **console:** Write directly to system console if there is an error while sending to system logger.
2. **stderr:** Print to stderr as well.
3. **pid:** Include process ID.
4. **severity:** Include a single letter indicating the severity level of the event.
5. **facility:** Include the name of the facility for the message.
6. **ident:** Include the string identification of the event.
7. **text:** Include the formatted text for the event.
8. **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

3.5 Upgrading from a previous version



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

Chapter 4:M-Link User management

4.1 How M-Link Stores Users

Users and information about their mailboxes is 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. Internet Messaging Administration Web Application provides a convenient user interface for managing this information using web.

4.2 User Administration using Internet Messaging Administration Web Application

Use "User Manager - Edit user" screen to edit user information. The information is divided into several sections.

4.2.1 Account Name

Option location: "Edit User" page, "Attributes for the Internet User" section.

Description: The account name is a required attribute and its value is used to uniquely identify each email account.

The account name is also used as the user JID.

Please enter the left hand side (e.g. "joe" for "joe@example.com") of the account name in the "" field. The domain part can be entered in the field below. An existing domain can be selected by pressing on the "..." button, or a new domain can be entered here.

Example: *Jack.Smith*

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

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

4.2.1.1 Password

Option location: "Edit User" page, "Attributes for the Internet User" section.

Description: Each Jabber account must have a password.

Example: *mysecret*

LDAP attribute name: *userPassword*

4.2.2 User information

Options described in this section are purely informational. They are only used to help administrators to uniquely identify users, and will be made available in users' vCards via XEP-0054.

4.2.2.1 First name

Option location: “Edit User” page, “User information” section.

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

Example: *Jack*

LDAP attribute name: *givenName*

4.2.2.2 Surname

Option location: “Edit User” page, “User information” section.

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*

4.2.2.3 Display name

Option location: “Edit User” page, “User information” section.

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*

4.2.2.4 Email

Option location: “Edit User” page, “User information” section.

Description: The official email address of the account as displayed in public Directory. This value is **not** used for mail routing.

Typically this field contains one of the email addresses specified in the Account Email Addresses field.

This option only appears in the Advanced view.

Example: *Jack.Smith@example.com*

LDAP attribute name: *mail*

4.2.3 M-Link specific attributes

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

4.2.3.1 Account Status

Option location: “Edit User” page, “M-Link specific attributes” section.

Description: This field can have one of three values: “Active” (the account is active, the default), “Inactive” (temporary disabled account, the user can’t log into M-Link services) or “Deleted” (account is deleted, this record is used internally by M-Link).

Example: *Inactive*

LDAP attribute name: *inetUserStatus*

4.2.3.2 Available services

Option location: “Edit User” page, “M-Link specific attributes” section.

Description: Controls which services a user can access by default. It is used to control access to M-Link using XMPP.

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 **xmpp**.. Services not explicitly listed in this listed are controlled by the global “*Default new service access*” option.

If this attribute is not set, the global default is used instead.

This option only appears in the Advanced view.

Default value: empty string

Example: `xmpp=grant`

LDAP attribute name: *AccessibleServices*

4.2.3.3 Clearance

Description: Specifies the user's clearance for use in Security Label processing.

The value is an XML representation of the user's clearance.

If this attribute is not set, the user will have 'default' access under the configured security policy.

Default value: not present

LDAP attribute name: *sioClearance*

4.2.3.4 Auto Accept

Description: If set, any inbound subscription request will automatically be accepted, without consultation with the user. There are two attributes – one setting affects all subscription requests – including local – the other affects only local. If the global attribute is set, then the local attribute is ignored.

See also the server defaults, in Section 3.2.7.2.

Default values: unset

LDAP attribute names: *isodeMlinkAutoAccept, isodeMlinkAutoAcceptLocal*

4.2.3.5 Auto Subscribe

Description: If set, any inbound subscription request will automatically cause a corresponding outbound subscription request. There are two attributes – one setting affects all subscription requests – including local – the other affects only local. If the global attribute is set, then the local attribute is ignored.

See also the server defaults, in Section 3.2.7.3.

Default values: unset

LDAP attribute names: *isodeMlinkAutoSubscribe, isodeMlinkAutoSubscribeLocal*

4.2.3.6 Offline Messages

Description: If set, messages received when the user is disconnected (or, more accurately, has no matching resource nor any resource with non-negative presence), will be held by the server until such time as they reconnect.

A second attribute extends this behaviour from just messages of type Normal or Chat to Headline messages as well, which are not typically intended for offline storage.

See also the server controls, in Section 3.2.7.4 and onward.

Default values: unset

LDAP attribute names: *isodeMlinkOffline, isodeMlinkOfflineHeadline*

4.2.4 **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 IMA 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 (ETC/DIR)/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|-d}* *{-r|-n}* *[-f]* *userid* [*<properties>*]: Adds a new XMPP user account by creating the userid in the SASL database. The account is either created enabled (with the password specified after the -p) 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, suitable for email accounts. -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 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 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*: Creates a new disabled user account.

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*.

enable *userid*: Enables a disabled *userid*.

disable *userid*: Disables a *userid*.

status *userid*: Get the status of *userid*.

list *[-d domain_reg]* *[-u user_reg]* *[-v]*: 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 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.

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.

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 by M-Link – other service names are recognized by M-Box, Isode's internet message store:

Service name	Service description
xmpp	Controls access over XMPP 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-Link 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 5: User Authentication

5.1 SASL Authentication

XMPP uses the Simple Authentication and Security Layer (SASL) [RFC 4422] framework for authentication. The Isode XMPP server uses 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.

5.1.1 Generic SASL options

5.1.1.1 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 XMPP server. The intersection of the set of available mechanisms with this list is returned in the XMPP stream `<mechanisms/>` element: e.g. if "PLAIN,DIGEST-MD5,GSSAPI" are available and the value of this option is "SRP,GSSAPI,DIGEST-MD5", the `<mechanisms/>` 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 3.2.7.1](#)) affect the final list of available options as well. If this option is not set, all installed SASL mechanisms are allowed. See [Section 5.1.2](#) 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>`

5.1.1.2 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: 0

Example: 1

XML option name: *min_ssf* and *max_ssf*

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

5.1.1.3 List of SASL mechanisms

Option location: This option is not accessible through IMA.

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

Default value: (LIBDIR)/sas12

(UNIX) Example: */usr/local/lib/sas12*

XML option name: *plugin_dir*

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

5.1.1.4 Password verification method

Option location: This option is not accessible through IMA.

Description: This option contains the name of the password verification method. Currently only a single password verification method called "auxprop" is supported. The specified password verification method is used to verify passwords during SASL PLAIN authentication, as well as when using XEP 0078 plaintext authentication.

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

Default value: *auxprop*

Example: *auxprop*

XML option name: *pwcheck_method*

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

5.1.1.5 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: *ldapdb*

XML option name: *auxprop_plugin*

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

5.1.2 SASL Mechanisms

The M-Link supports multiple SASL mechanisms via a plugin system. When the XMPP server starts up it loads all the plugins installed in (LIBDIR)/sas2. This makes it simple to completely disable certain mechanisms (by removing the plugin file and restarting the XMPP server) or to add additional mechanisms (by copying in the new plugin and restarting the XMPP server).

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

5.1.2.1 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
OTP	One-time password	Good

5.1.2.2 Shared Secret Mechanisms

For CRAM-MD5 and DIGEST-MD5 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.

5.1.3 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.

5.1.3.1 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-Link 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.

5.1.3.2 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-Link 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.

5.1.3.3 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-Link 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.

5.1.3.4 User entry filter

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

Description: This option specifies an LDAP filter [RFC 2254] used to select M-Link 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.

5.1.3.5 Domain entry filter

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

Description: This option specifies an LDAP filter [RFC 2254] used to select M-Link 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.

5.1.4 **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-Link users are stored in LDAP, but the M-Link server configuration and shared folders are not, or if users and M-Link 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.

5.1.4.1 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>`

5.1.4.2 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* options is used by default. This option is ignored if SASL bind 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>`

5.1.4.3 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* options 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>

5.1.4.4 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, the provided value is ignored and the value of *ldap_bind_pwd* options 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>

5.1.4.5 Bind method

Description: Specifies the LDAP authentication method. 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, the provided value is ignored and the value of *ldap_bind_method* options 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>

5.1.4.6 STARTTLS

Description: This option tells LDAPDB to use TLS. The option may be set to "try" or "demand". When set to "try" any failure in StartTLS is ignored. When set to "demand" then any TLS failure aborts the LDAP connection. The default is None, i.e. don't use TLS.

Default value: -None-

Example: demand

XML option name: *ldapdb_starttls*

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

5.1.4.7 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>

5.1.4.8 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>

5.2 Userid canonicalization during authentication process

For user convenience M-Link 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-Link 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-Link 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-Link 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-Link is used instead.

Example:

An M-Link 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-Link 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-Link 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-Link server will canonicalize it to "test@example.net", i.e. it will not change the provided userid.

Chapter 6: Multi-User Chat

Multi-User Chat rooms allow several entities to communicate together, whether by simple text messages or more complex structured XML. They have both affiliations – long-term, persistent, relationships between users and the room – and roles – short-term relationships between room occupants and the room.

M-Link's implementation is a relatively full-featured implementation of XEP-0045, and supports persistent rooms if configured, and multiple MUC domains per IM domain.

6.1 Persistent Rooms

Rooms are persisted in the LDAP configuration directory, if configured. For any declared MUC domain, M-Link performs a subtree search below the M-Link directory entry itself, searching for a virtual domain entry corresponding to the domain name.

If none is found, persistent rooms will not be offered to clients, otherwise all rooms will be loaded and precreated. Typically this is done once, at startup.

Such an object should be created by the system administrator for all services which require persistent rooms. This layout deliberately matches the layout required for the configuration of the Multi-User Chat service itself via the directory.

Rooms made persistent will be stored by creating a directory entry of objectClass `isodeMlinkPubsubNode`, with an `isodeMlinkPubsubNode` attribute value of the room's node name, that is, the portion of the room jid to the left of the `@`-sign. Changes to the room's configuration made in XMPP will be immediately reflected in the room's directory entry – however the reverse is not true, and administrators are therefore recommended to make runtime room configuration changes via XMPP.

6.2 Creating and configuring rooms

Persistent rooms can be precreated in LDAP, but the simplest method for creating chatrooms is to simply use a standard XMPP client. Rooms are, by default, made as partial copies of a special room with the node-name “template-room”, which can be changed to provide useful defaults.

6.2.1 Room Configuration Options

This section details each room configuration option. Options are identified by the XEP-0004 form field variable name, and also include the name used by M-Link in the form, and details of how this maps to LDAP attributes of the node entry for persistent rooms.

Form names beginning with “muc#” are standardized as part of XEP-0045, and more details will be found there.

6.2.1.1 Room Title

Form Name: muc#roomconfig_roomname

LDAP Attribute: commonName

The Room Title, or formal name, of the room. Some clients present this in the UI.

6.2.1.2 Description

Form Name: muc#roomconfig_roomdesc

LDAP Attribute: description

The room's description is shown in some clients.

6.2.1.3 Publicly Listed

Form Name: muc#roomconfig_publicroom

LDAP Attribute: isodeMlinkPubsubPublic

This controls whether the room is listed in the Service Discovery “disco#items” listing of the MUC service.

6.2.1.4 Moderated

Form Name: muc#roomconfig_moderatedroom

LDAP Attribute: isodeMlinkXep45DefaultRole (see notes)

If this is set (and Members Only is not) then the Default Role will be set to Visitor. Otherwise, it will be set to Participant.

Visitors are unable to send messages to be broadcast through the chatroom to all participants, commonly known as having “voice”.

6.2.1.5 Members Only

Form Name: muc#roomconfig_memberonly

LDAP Attribute: isodeMlinkXep45DefaultRole (see notes)

If this is set, the Default Role is set to None. Non-members will therefore be unable to join.

6.2.1.6 Password

Form Name: muc#roomconfig_roomsecret

LDAP Attribute: isodeMlinkXep45Password

This stores a password (in plaintext) for the room, required to be presented (in plaintext) by users trying to join the room. This provides very limited security.

6.2.1.7 Persistent Room

Form Name: muc#roomconfig_persistentroom

LDAP Attribute: (see notes)

If checked, the room will be persisted when empty, and will therefore be stored in the directory.

6.2.1.8 SIO Label

Form Name: x-isode#sio_label

LDAP Attribute: sioLabel

This contains a security label of the room. In order to join the room, users must have a clearance which matches this label.

6.2.1.9 SIO Clearance

Form Name: x-isode#sio_clearance

LDAP Attribute: sioClearance

This contains the clearance for the room. Messages will be dropped if they do not have a label acceptable by the clearance.

6.2.1.10 Default SIO Label

Form Name: x-isode#sio_default_label

LDAP Attribute: sioDefaultLabel

Unlabelled messages to this chatroom will be implicitly labelled with the label contained here, before any clearances are checked.

6.2.1.11 Real JIDs visible to

Form Name: muc#roomconfig_whois

LDAP Attribute: isodeMlinkXep45RealjidsRole

This can be set to various roles to provide a control on what the minimum roles must be to have real jids, as well as in-room jids, presented to users.

6.2.1.12 Invitations sent by

Form Name: x-isode#roomconfig_invite

LDAP Attribute: isodeMlinkXep45InviteRole

This sets the minimum role required to send mediated invitations – ie, XEP-0045 invitations. Invitations sent through a members-only room will cause the invitee to be made a member.

6.2.1.13 Private messages sent by

Form Name: x-isode#roomconfig_privmsg

LDAP Attribute: isodeMlinkXep45PrivMsgRole

This sets the minimum role required to send private messages through the MUC system to another occupant.

6.2.1.14 Allow vCards on anonymous users?

Form Name: x-isode#roomconfig_vcards

LDAP Attribute: isodeMlinkXep45vCards

For users whom the requestor can see the real jid, any vCard (XEP-0054) request is redirected to the users' bare jids. Where the requestor cannot see the real jid, they will only be redirected if this option is set.

6.2.1.15 Allow all users to change subject?

Form Name: muc#roomconfig_changesubject

LDAP Attribute: isodeMlinkXep45SubjectRole

Internally, and in the directory, this is the minimum role required to change the subject in the room. In XMPP, this is a boolean, and when set it will automatically set the SubjectRole to None.

6.2.1.16 History Length

Form Name: x-isode#roomconfig_history_length

LDAP Attribute: isodeMlinkPubsubMaxItems

This specifies the number of messages “replayed” to new room occupants on joining.

6.2.1.17 Clear History

Form Name: x-isode#roomconfig_history_clear

LDAP Attribute: (see notes)

This pseudo-item, when set, will clear the existing history on set.

6.2.1.18 Accept any message types

Form Name: x-isode#roomconfig_pass_any

LDAP Attribute: isodeMlinkXep45PassAny

If set, then by default on simple <body/> elements in a message will be passed through – others will be stripped from the message.

6.2.1.19 Accept XHTML-IM messages

Form Name: x-isode#roomconfig_pass_html

LDAP Attribute: isodeMlinkXep45Html

If both this and the above are set, then XHTML-IM messages, containing rich formatting, will also be passed through the MUC.

6.2.2 Affiliations

This section explain the default rights and roles of affiliations. No affiliations or roles are allowed to manipulate users of a higher affiliation or role, hence a Administrator cannot kick an Owner or Super from a room.

6.2.2.1 Super

Members of the “operators” group will automatically be Owners of any chatroom they join on the server. Owners of the template-room will automatically be owners of rooms they join on the same server.

This automatic owners are, internally, treated as “Super”, with a role of “Super”, which are presented as “Owner” and “Moderator” respectively.

They do not appear on room affiliation lists, and these affiliations are temporary only.

6.2.2.2 Owner

Room owners conceptually have complete control of the room. They will be made Moderators on joining the room.

6.2.2.3 Administrator

Administrators conceptually manage the room, and can view and change affiliations equal to or lower than theirs. Like owners, they will be made Moderators on joining the room.

6.2.2.4 Members

Members have a long-term relationship with the room, and will always become Participants when they join a room, allowing them to post messages in the chatroom itself. If made Moderator, then they can manipulate users with affiliations equal to or less than their own.

6.2.2.5 Outcast

These are conceptually banned from the room, and cannot join the room.

6.2.2.6 None

Anyone without an affiliation – even after domain matching – will be assigned the default role when they join the room. This is typically indirectly set either by the “Moderated” or “Members Only” settings to be either Participant, Visitor, or None. The latter setting will not allow them to join the room.

Chapter 7: Groups for Rosters and Authorization

7.1 Overview

M-Link supports using group information from a directory, to support roster prepopulation and authorization. Groups can be based on SASL userids, on an LDAP search, or on an X.500 style group held either in the configuration or in the authentication directory.

For a group of any type, a corresponding entry will need to be created in the configuration directory, within a `cn=Groups` container beneath the main M-Link entry. To obtain enhanced functionality, including roster population, groups must be defined with the `isodeMlinkGroupInfo` auxiliary object class – this class may be added to an existing group via Sodium easily.

Note that group membership is defined as either SASL userids or Distinguished Names of user entries in the authentication directory, and to avoid complex reverse-mapping rules, the group membership (and therefore rosters defined by it) will resolve as users authenticate. This has the advantage that users who never authenticate won't “clutter” rosters.

7.1.1 Roster Groups

Roster groups may be prepopulated by setting the “Use as Roster group” attribute to true, and optionally exposing the group membership to the named groups. This is detailed below.

7.1.2 Chatroom permissions, etc

Any group may be referred to with a suitably formed jid, by using the constructed internal name and the group domain. These jids may be used as an entry in the affiliations of a chatroom (or PubSub node), such that all members of that group will have that affiliation. Note that specific affiliations set for a member will take precedence, so if all of the Sales Team is banned from a room by setting the sales.team@groups.example.net as an Outcast of the room, yet salesman@example.net

7.2 Types and ObjectClasses

7.2.1 Common

7.2.1.1 Name

LDAP attribute: `commonName`

This defines the formal name of the group. Note that internally, groups have two names, the name used for presentation (for example, in roster group names), and the simplified name used as the jid's node-name (ie, the left-hand-side of the XMPP address), which is used where a jid is required.

Typically, the simplified name will be a lower-cased version of the name, with most non-alphanumeric characters folded to a period. So a name “Sales Team” would have a jid beginning “sales.team@”, and a group entitled “Sales (UK Team)” would have the somewhat confusing jid beginning “sales..uk..team.@”.

Groups defined within the authentication directory still require a `commonName` to refer to them by, and it may be useful to use simpler names for these entries for the above reasons.

7.2.2 M-Link Group

The `isodeMlinkGroupInfo` object class, which can be added with Sodium via the M-Link Group option, contains optional configuration common to all types of groups.

In some group types, these options are presented as part of a single template tab alongside the groups' own options, whereas in others they are presented in a tab entitled “Isode M-Link Group Info”.

7.2.2.1 Use as Roster group

LDAP attribute: `isodeMlinkGroupRoster`

This boolean option controls whether the members of the group will be inserted into rosters. At minimum, all members of the group will have a two-way subscription to each other. Users may remove these if required.

7.2.2.2 Expose Members to Other Groups

LDAP attribute: `isodeMlinkGroupExpose`

This multivalued string attribute contains the `commonNames` of groups to which this group will be exposed to. In practise, this means that for a group A with members A1, A2 which is exposed to a group B with a single member B1, then A1 and A2 will be mutually subscribed, both will see B1 with a subscription of “from” – so typically won't see B1 online – and B1 will see both A1 and A2 with a subscription of “to”.

7.2.3 Isode SASL Group

The `mboxGroup` object class is used to define groups by SASL identifier. A particular advantage of this group type is that they may be defined using file-based configuration if required – contact Isode Support for details if needed.

The SASL group called “operators”, if present, is special, as it authorizes members to use the XEP-0133 and similar administrative commands, and also allows the members to join all chatrooms on the system.

7.2.3.1 Members

LDAP attribute: `mboxGroupMember`

This multivalued string attribute defines the membership of the group. To allow for future expansion, user identifiers (which contain no “@” for users within the default domain) should be prefixed with “user:”, thus a user in the default

domain would be entered as “user:fred” for example, and a user in an alternate domain would be “user:[barney@example.net](#)”.

7.2.4 Group of Names

The standard X.500 `groupOfNames` class can also be used to directly define a group as a list of distinguished names.

7.2.4.1 Group Members

LDAP attribute: `member`

This multivalued attribute contains members expressed as DNs, as per the standard attribute.

7.2.5 Isode M-Link Group Reference

The `isodeGroupReference` object class declares a group held on the authentication directory. It expects an X.500 Group of Names, but the only actual requirement is that a `member` attribute exists on the referenced entry which contains the DNs of the users of the group – this requirement includes Active Directory group objects.

7.2.5.1 Distinguished Name

LDAP attribute: `isodeAuthGroupName`

This single-valued attribute contains the DN of a group held in the authentication directory.

7.2.6 Isode M-Link Group Search

The `isodeGroupSearch` object class defines a group as the results of an LDAP search, which is executed on group load. The attributes of the group correspond to the usual LDAP search parameters. The DNs used from the search results are, by default, the DNs of the objects returned by the search, but this can be overridden.

Note that using a Search Scope of “base”, and a Use Attribute of “member”, and no Search Filter is equivalent to the Isode Group Reference above in effect.

7.2.6.1 Search Base

LDAP attribute: `isodeAuthGroupBase`

This attribute contains a DN to use as the base object of the search.

7.2.6.2 Search Scope

LDAP attribute: `isodeAuthGroupScope`

This may be either `subtree`, `base`, or `onelevel`, to search the base and anything below, just the base object, or the immediate children of the base object.

7.2.6.3 Search Filter

LDAP attribute: isodeAuthGroupFilter

This string attribute may be used to specify an LDAP search filter.

7.2.6.4 Use Attribute

LDAP attribute: isodeAuthGroupAttr

If specified, this attribute's values will be extracted from each object found by the search, and the resultant list will be treated as a list of DNs representing the group's membership.

If this is not specified, the objects themselves will be treated as the members of the group, references by their DNs.

Appendix A: Command line parameters for M-Link service applications

All M-Link 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

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

```
<ms_options>
<config_location>local</config_location>
<ms_user>mbox</ms_user>
<login_disabled>>false</login_disabled>
<domain>myisp.net</domain>
<userdir>c:\JabberStore\users</userdir>

<ldap>

<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>

<sasldbMappingSuffix>ou=users, o=myisp</sasldbMappingSuffix>
<sasldbSearchSuffix>ou=users, o=myisp</sasldbSearchSuffix>
<sasldbDefaultDomain>myisp.net</sasldbDefaultDomain>
</ldap>

</ms_options>
```

Appendix C: Example LDAP configuration

This section provides examples of M-Link server configuration entry and persistent MUC service. The two examples show a possible LDAP representation of the XML configuration described in 7.2.6.4.

```
dn: cn=M-Link, cn=Servers, o=myisp
objectClass: mboxVirtualDomain
objectClass: mboxStoreTailoringObject
objectClass: top
cn: M-Link
uid: mbox
mt-local-domain-site: myisp.net
mboxRootUserDir: c:\JabberStore\users

mboxCleartextLoginDisabled: FALSE
mboxListeners: lmtpl://mail.example.com:2003
```

```
dn: cn=Chatrooms, cn=M-Link, cn=Servers, o=myisp
objectClass: mboxVirtualDomain
objectClass: top
mt-local-domain-site: conference.myisp.net
description: Public Chatrooms
```

The next entry shows a persistent MUC room, configured as a public chatroom where private messages are forbidden, and HTML messages are filtered out of the public stream:

```
dn: isodeMlinkPubsubNode=chat, cn=Chatrooms, cn=M-Link,
cn=Servers, o=myisp
objectClass: isodeMlinkXep45Room
objectClass: isodeMlinkPubsubObject
objectClass: top
isodeMlinkPubsubOwner: usera@myisp.net
cn: General Chat
description: A room for general chatter
isodeMlinkXep45PassAny: FALSE
isodeMlinkXep45Html: FALSE
isodeMlinkXep45DefaultRole: participant
isodeMlinkXep45PrivMsgRole: moderator
isodeMlinkPubsubPublic: TRUE
```

Below you can see an example user entry:

```
dn: cn=usera, 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@myisp.net  
displayName: Jack Smith Jr.
```

Appendix D: TLS Cipher List Format

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 be 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.

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.

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.

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 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 SHA	EDH-DSS-DES-CBC3- SHA
SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA CBC-SHA	EXP-EDH-RSA-DES- CBC-SHA
SSL_DHE_RSA_WITH_DES_CBC_SHA SHA	EDH-RSA-DES-CBC- SHA
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA 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 SHA	EXP-ADH-DES-CBC- SHA
SSL_DH_anon_WITH_DES_CBC_SHA	ADH-DES-CBC-SHA
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA	ADH-DES-CBC3-SHA

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 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 SHA	EDH-DSS-DES-CBC3- SHA
TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA CBC-SHA	EXP-EDH-RSA-DES- CBC-SHA
TLS_DHE_RSA_WITH_DES_CBC_SHA SHA	EDH-RSA-DES-CBC- SHA
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA 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 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

AES ciphersuites 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

Additional Export 1024 and other cipher suites

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

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

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

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-Link redundancy to LDAP service failures

When configured to use LDAP, M-Link can cope with some temporary LDAP server outages. When an M-Link service starts up it needs to successfully read M-Link 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-Link 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-Link 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-Link 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-Link service and most M-Link utilities. If the M-Link service failed to refresh the LDAP configuration, it will keep trying to reconnect to the LDAP server and refresh the configuration.

Note that there is no caching of user authentication information.