MLINKADM-19.4

M-Link Administration Guide

Isode

Table of Contents

1 Software version

This guide is published in support of M-Link R19.4. It may also be pertinent to later releases. Please consult the release notes for further details.

2 Readership

This guide is intended for administrators who plan to configure and manage XMPP services using M-Link R19.4.

3 Typographical conventions

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

Object	Example
File and directory names	*/var/isode/log*
Program and macro names	isode.xmppd
Input to the system	`cd newdir`
Cross references	see Section 4, “File system place holders”
Additional information to note, or a warning that the system could be damaged by certain actions.	Notes are additional information; cautions are warnings.

4 File system place holders

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

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

Name	Place holder for the directory used to store...	Windows (default)	UNIX
`$(SHAREDIR)`	Configuration files that may be shared between systems.	*C:\Program Files\Isode M-Link\share*	*/opt/isode/mlink/share*
`$(SBINDIR)`	Server programs.	*C:\Program Files\Isode M-Link\bin*	*/opt/isode/mlink/sbin*
`$(VARDIR)`	Storing local data.	*C:\Isode\M-Link*	*/var/isode/mlink*
`$(LOGDIR)`	Log files.	*C:\Isode\M-Link\log*	*/var/isode/mlink/log*

Name	Place holder for the directory used to store...	Windows (default)	UNIX
$(AATETCDIR)	Configuration files for Archive Admin Tool	*C:\Isode\etc*	*/etc/isode*
$(AATSHAREDIR)	Archive Admin Tool configuration files that may be shared between systems.	*C:\Program Files\Isode \share*	*/opt/isode/share*
$(AATBINDIR)	Directory containing programs for Archive Admin Tool	*C:\Program Files\Isode\bin*	*/opt/isode/bin*
$(ETCDIR17)	Configuration files for M-Link R17.0.	*C:\Isode\etc*	*/etc/isode*

5 Support queries and bug reporting

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

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

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

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

6 Export Controls

M-Link uses TLS (Transport Layer Security) to encrypt data in transit. This means that M-Link is subject to UK Export Controls.

For some countries (at the time of shipping this release, these comprise all EU countries, United States of America, Canada, Australia, New Zealand, Switzerland, Norway, Japan), these Export Controls can be handled by administrative process as part of evaluation or purchase.

For other countries, a special Export License is required. This can be applied for only in context of a purchase order from Isode.

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

XMPP servers are generally deployed with TLS for security reasons and Isode strongly recommends that all operational deployments use the export-controlled TLS feature.

You must ensure that you comply with these Export Controls where applicable, i.e. if you are licensing or re-selling M-Link. M-Link is subject to an Isode license agreement and your attention is also called to the export terms of the license agreement.

Chapter 1 M-Link: Getting Started

This chapter discusses M-Link Server creation.

This chapter contains sections on the following topics:

1.1 About M-Link Server

M-Link Server is an *XMPP* server with a web-browser based configuration interface.

This section provides a brief introduction to terminology that will help you better understand subsequently provided materials.

A single M-Link Server instance, or two M-Link Server instances cooperatively, can provide an XMPP service for an XMPP domain or set of domains. The terms M-Link Server or *node* are to refer to a particular M-Link Server instance. The term M-Link service or *service* are used to refer to the M-Link Server instance(s) providing a particular XMPP service. When two M-Link Server instances are cooperatively providing an XMPP service, the *service* (or M-Link service) is said to be provided by a *cluster* of *node*s (or M-Link Server instances).

The distinction between *service* and *node* is important to understand in the configuration of M-Link service as some settings may apply *service* wide while others are *node* specific.

1.2 Starting and Stopping M-Link Server

Once installed, the M-Link Server will start automatically as a system service, and thereafter the service can be controlled with the system-provided mechanisms.

On first start (and subsequently), M-Link will start with its configuration interface on port `5221`, ready to be activated and configured - this is accessible at `https://localhost:5221` on the server, or at `https://hostaddress:5221` from other machines. It will initially generate a self-signed certificate - both the certificate and the default port can be changed later through the configuration interface. The configuration interface can be accessed from a web browser running on a different machine from M-Link by substituting `localhost` with the hostname or IP of the server machine. As the initial certificate is self-signed, it will need to be manually trusted before a browser will accept it - usually by clicking through a security warning.

Note: Access to the administration interface might be restricted by firewall policies. If the interface is unreachable, please check that there is a rule that allows access to port `5221`.

1.3 Creating an Initial Administrative User

M-Link Server can be configured by administrators. Administrators are users distinct from the users of the XMPP service. An initial administrator account is created as the first step through the web configuration interface - this user is stored in M-Link's configuration (the password is stored with a strong one-way hashing function, and cannot be recovered). Currently M-Link is limited to a single administrator account.

1.4 Installing an Isode Activation Key

In order to use an M-Link Server instance on a host system, an activation key file is required. This file needs to be installed through M-Link's web browser-based configuration interface. The initial configuration process will lead you through generation of an Activation Request and the subsequent installation of the Activation Key supplied by Isode in response to receipt of the Activation Request.

Questions regarding licensing should be directed to isode.support@isode.com.

1.5 M-Link Server Runtime User

When running on Unix, M-Link Server runs as an unprivileged user, created during package install, and does not need to be started as 'root'.

On Windows, the M-Link Server runs as Windows service(s) under the `LocalSystem` account.

1.6 Creating or Joining a Cluster

> **Note:** This section applies only if the activation key allows clustering. Proceed straight to Section 1.7, “Adding Initial Configuration” if not offered a choice to either “`Create a new configuration`” or “`Join an existing cluster`”.

M-Link Server can be configured to run in a cluster, where multiple M-Link Server instances, or nodes, cooperate to provide a single XMPP service. This is useful for providing high availability and load balancing.

Once logged into the M-Link Administration Interface and M-Link is activated with some clustering capability, you are offered a choice to create a stand-alone configuration or to join an existing cluster:

“`Create a new configuration`”
: This option lets you proceed with configuration as a stand-alone M-Link Server which can then be upgraded to a cluster (see Section 8.9.1, “Enabling Clustering”). It is the option to choose when there is no existing cluster or a separate cluster

is needed. After choosing this option, proceed to Section 1.7, “Adding Initial Configuration”.

Note: Configuration of the service can happen before or after enabling clustering. However, there are a number of options that, if they're going to be changed, should be changed as early as possible - particularly those to do with paths and ports.

“`Join an existing cluster`”

M-Link Server instances in a cluster share the same configuration, so joining an existing cluster will load the configuration from another M-Link and configuration steps described in following sections may have already been covered.

Choosing this option leads to a sequence of forms that will introduce the new instance to the cluster:

- The first form takes a few details, specific to this instance, such as its name. Submitting the form creates a *join request token*, displayed on the next form.
- The *join request token* will need copying into the M-Link Administration Interface of another node in the cluster, proceeding as instructed in Section 8.9.3, “Adding a Node to the Cluster” and collecting a *join response token*.
- The *join response token* will need copying back into the form at the new instance. It includes means of identification and authentication, allowing the new instance to connect with the existing cluster. On submission, the new instance will be able to synchronize with the cluster.

1.7 Adding Initial Configuration

Once you're logged into the M-Link Administration Interface and M-Link is activated you're ready to add initial configuration. Depending on the type of M-Link server that is being configured (and has been activated) you will need to either add domains for M-Link to host or add Peer Controls to allow routing between remote servers. Details of the configuration of XMPP Trunking and Peer Routing can be found in the Chapter 4, *Peer Controls and Links* chapter.

1.8 DNS Configuration

XMPP clients and servers typically use the *Domain Name System* (DNS) to determine the IP addresses and TCP ports of XMPP servers they need to connect to. For XMPP Client-to-Server (C2S) the following discovery approach is typically used.

Clients first look for SRV resource records for the XMPP domain they wish to connect to. Each SRV resource record for a domain provides a location (a domain name and a port) for the service (e.g., XMPP C2S). Multiple SRV resource records may be published for each service. In addition to the service location, each SRV resource record has a priority and a weight. Preference of two SRV resource records is given to one with the lower priority value or, in the case the equal priority, to the resource record with the higher weight. The location’s domain name is resolved to a set of IP addresses using normal A and AAAA lookup. Depending on the client’s configuration, if both IPv4 and IPv6 are supported, preference may be given to one or the other. Otherwise, IP addresses are typically tried in the order provided by the Domain Name System, which may or may not change with each lookup.

If SRV resource records are not published for the domain, clients and servers will resolve the domain name to a set of IP addresses and connect to each using the default port for the protocol. This is accomplished by looking for A and AAAA resource records for IPv4 and IPv6 addresses respectively, possibly with CNAME resolution when necessary. As above, this resolves to a set of IP addresses. Depending on the client's configuration, if both IPv4 and IPv6 are supported, preference may be given to one or the other. Otherwise, IP addresses are typically tried in the order provided by the Domain Name System, which may or may not change with each lookup.

In deployments where DNS is not available, clients will often use other host-to-IP address lookup systems, such as those provided by a hosts file. These typically behave similarly to DNS with only A and AAAA resource records. Configuration of alternative host-to-IP address lookup systems is not detailed in this guide.

To illustrate proper configuration of DNS, consider an XMPP service consisting of a two node cluster providing Instant Messaging service for the domain *example.com*, where the cluster nodes are named *node1.example.com* and *node2.example.com*. The nodes have IPv4 addresses *192.0.2.1*, *192.0.2.1* respectively. The nodes have IPv6 addresses *2001:DB8::1*, *2001:DB8::1* respectively. We assume the XMPP C2S protocol is available on its default port, 5222. Additionally we'll assume this IM service is not co-located with other example.com services such as WWW.

Note that these examples utilize the Domain Name System *zone file* format specified in *RFC 1035* (§ 5) and *RFC 1034* (§ 3.6.1). Furthermore,the examples assume placement in the zone *example.com*:

```
$ORIGIN example.com.
```

Each node should have an appropriate resource records for addressing purposes:

```
node1 IN A 192.0.2.1
node1 IN AAAA 2001:DB8::1
node2 IN A 192.0.2.2
node2 IN AAAA 2001:DB8::2
```

SRV resource records can then be provided. Here we've given equal preference to each node in our cluster.

```
_xmpp-client._tcp IN SRV 1 1 5222 node1
_xmpp-client._tcp IN SRV 1 1 5222 node2
```

While the domain naming each node allow for users to manually configure their client to connect to any one of the two nodes, it is often desirable to provide a domain which they can use to have the client connect to any of the nodes. The following resource records creates *xmpp.example.com* for this purpose as well as other purposes discussed below.

```
xmpp IN A 192.0.2.1
xmpp IN AAAA 2001:DB8::1
xmpp IN A 192.0.2.2
xmpp IN AAAA 2001:DB8::2
```

If the DNS servers providing the example.com zone support CNAME round robins, the *xmpp.example.com* A and AAAA resource records can be replaced with:

```
xmpp IN CNAME node1
xmpp IN CNAME node2
```

If a WWW service is available at the URL https://*example.com* or http://*example.com*, host-meta data for BOSH clients can be published using either XML or JSON.

XML host-meta example:

```
<?xml version='1.0' encoding=utf-9'?>
<XRD xmlns='http://docs.oasis-open.org/ns/xri/xrd-1.0'>
   ...
   <Link rel="urn:xmpp:alt-connections:xbosh"
       href="https://xmpp.example.com:5280/http-bind/" />
   ...
<XRD>
```

JSON host-meta example:

```
{
   ...
   "links": [
      ...
      {
        "rel": "urn:xmpp:alt-connections:xbosh",
        "href": "https://xmpp.example.com:5280/http-bind/"
      },
      ...
   ]
}
```

For XMPP S2S the following discovery approach is similar to XMPP C2S. For the IM domain, the following would be published assuming S2S is available on the default port, `5269`:

```
_xmpp-server._tcp IN SRV 1 1 5269 node1
_xmpp-server._tcp IN SRV 1 1 5269 node2
```

Additionally, DNS SRV records should be created for each other XMPP domain accessible by other XMPP servers. For instance, assuming a Multi-User Chat service domain of *`muc.example.com`* and Publish-Subscribe service at *`pubsub.example.com`*, the following additional resource records should be published:

```
_xmpp-server._tcp.muc IN SRV 1 1 5269 node1
_xmpp-server._tcp.muc IN SRV 1 1 5269 node2

_xmpp-server._tcp.pubsub IN SRV 1 1 5269 node1
_xmpp-server._tcp.pubsub IN SRV 1 1 5269 node2
```

1.9 Firewall Considerations

By default, M-Link will listen on a select number of TCP ports. All ports are configurable through the M-Link Administration Interface when the “Customise Ports” is enabled. Depending on the environment, firewall rules will need to be created to block or expose these ports. Isode recommends that ports are only exposed as-needed to provide the service, as in the following table.

Table 1.1. TCP ports used by M-Link

Port	Recommended exposure	Description
3999	Cluster only	Port on which the cluster nodes communicate. This port should not be exposed generally, but only between servers in a cluster.

Port	Recommended exposure	Description
5001	Users if required	The port used for HTTP File Upload (*XEP-0363*) by M-Link User Server and M-Link MU Server.
5080	Archive administration only	Port used by the M-Link Administration Interface for admin administration. This port should not be exposed generally, but only to machines that will be used to administer the service.
5221	Administration only	Port on which the M-Link Administration Interface is reachable. This port should not be exposed generally, but only to machines that will be used to administer the service and between cluster nodes.
5222	Users	The port used for XMPP Client-to-Server by M-Link User Server and M-Link MU Server.
5224	Localhost only	Port used for communication between various subservices of M-Link, such as the M-Link Archive Server or the BOSH service.
5269	Federating servers	The port for used for XMPP Server-to-Server by M-Link Edge Server, M-Link User Server and M-Link MU Gateway .
5280	Users if required	The port for used for BOSH by M-Link User Server and M-Link MU Server.

Chapter 2 M-Link Products

This chapter describes the different M-Link products.

M-Link is a single set of binaries which provides different products by enabling combinations of functionality (controlled via the Activation Key system). As a result, some chapters of this manual are not applicable to all products, as the features described will not be available.

The M-Link product set is:

2.1 M-Link User Server

This is the core Instant Messaging server, supporting 1:1 chat, Multi-User Chat (MUC), Personal Eventing and other XMPP services. See chapters:

- Getting Started
- TLS Configuration
- Security Labels, Clearances and Policies
- M-Link Configuration and Management
- Instant Messaging Domain and User Configuration
- MUC Room Creation and Administration
- M-Link Archive Server
- Monitoring
- HTTP API

2.2 M-Link Edge Server

This is used to provide an XMPP Boundary Guard service to protect organizational boundaries and provide Cross Domain services. See chapters:

- Getting Started
- Peer Controls and Links
- TLS Configuration
- Security Labels, Clearances and Policies
- M-Link Configuration and Management
- M-Link Archive Server
- Monitoring
- HTTP API

2.3 M-Link MU Server

This supports local clients using the standard XMPP Client/Server protocol, and communicates with other M-Link MU Gateway and M-Link MU Server instances using

the XEP-0361: Zero Handshake Server to Server Protocol and additionally supports "XEP-0365: Server to Server communication over STANAG 5066 ARQ", used in conjunction with XEP-0361, to support XMPP communication over HF Radio. See chapters:

- Getting Started
- M-Link MU Gateway
- Peer Controls and Links
- TLS Configuration
- Security Labels, Clearances and Policies
- M-Link Configuration and Management
- Instant Messaging Domain and User Configuration
- MUC Room Creation and Administration
- M-Link Archive Server
- Monitoring
- HTTP API

2.4 M-Link MU Gateway

Used to exchange messages between systems running on high quality links and those running over constrained networks, the M-Link MU Gateway communicates with M-Link MU Gateway and M-Link MU Server instances using XEP-0361 (and XEP-0365 for HF Radio) and with XMPP servers over normal quality links using the standard XMPP Server/Server protocol, allowing units communicating over constrained links to be integrated with existing standard XMPP installations. See chapters:

- Getting Started
- M-Link MU Gateway
- Peer Controls and Links
- TLS Configuration
- Security Labels, Clearances and Policies
- M-Link Configuration and Management
- M-Link Archive Server
- Monitoring
- HTTP API

Chapter 3 M-Link MU Gateway

This chapter introduces M-Link MU Gateway.

3.1 Introduction

Many military deployments will use network links with poor and variable quality. High latency is a particular problem for standard XMPP connectivity, especially with handshaking in session establishment. M-Link MU Gateway is specifically designed to overcome the issues of working in such environments.

Specific capabilities for operating over constrained networks include:

- Stream Compression
- Traffic Filtering, including Presence Stripping
- Optimised Server-to-Server Protocol
- Support for HF Radio (STANAG 5066)
- Connection fall-back and fall-forward

3.1.1 Stream Compression

Minimising the amount of data transferred is important, and so use of compression is desirable. compression can be enabled for both X2X and SLEP links. This compression uses the DEFLATE algorithm, which references previous data in the stream and thus becomes more effective for larger data sets or longer-lived streams.

3.1.2 Traffic Filtering

Traffic filtering removes data, and so modifies the service provided to the end user.

Filtering options available are:

- Removal of selected types of message (or other stanza)
- Removal of selected elements from messages (message folding)
- Removal of selected elements from presence stanzas (presence folding)

Traffic filtering is implemented via XSLT transformations which are applied to stanzas, and are configurable for individual Peer Controls, as described in Chapter 7, *Transformations*.

3.1.3 Optimised Server-to-Server Protocol

M-Link MU Gateway uses the *Zero Handshake Server to Server Protocol (XEP-0361)* operating over TCP to reduce data volumes and remove XMPP-level stream establishment handshaking, improving performance over constrained links. This uses three approaches:

- Use of a single (bi-directional) stream.
- Configuration of options (using peering controls) at both ends of the connection to avoid the need for negotiation during connection establishment.
- Full pipelining of stanzas, so that message stanzas are sent immediately after connection establishment.

The base transport is TCP. There will be a single TCP handshake to establish the TCP connection, and then data can flow without further handshakes. TCP optimizes use of the available link bandwidth.

While Zero Handshake Server to Server Protocol (*XEP-0361*) can be used efficiently over wired connections, it is of particular interest over satellite connectivity, where latency can significantly impair other server-to-server connections.

XEP-0361 may be operated over TLS. M-Link supports this to provide data confidentiality with an option to use peer authentication using X.509 strong authentication. With many constrained networks these services will be provided at the data link or network layer, and there is no functional requirement to provide them at the application layer. TLS adds some protocol overhead, and the handshaking may add significant latency.

3.1.4 Support for HF Radio

M-Link MU Gateway uses the SIS Layer Extension Protocol protocol with Server to Server communication over STANAG 5066 ARQ (*XEP-0365*) to enable communication over STANAG 5066 services, such as HF radio via the STANAG 5066 Subnet Interface Service. The use of SLEP provides a reliable bidirectional stream service which can support compression.

3.1.5 Connection fall-back and fall-forward

M-Link MU Gateway allows the configuration of fall-back links in priority order, e.g. falling back to use of STANAG 5066 links when satellite connectivity fails, or between different satellite systems. When a fall-back has occurred, M-Link will attempt to reestablish the primary connectivity and fall-forward to the higher priority link when that one becomes available again.

3.2 Configuring M-Link MU Gateway

M-Link MU Gateway is an M-Link server with no local users that is used to enable communication between groups of XMPP Servers over slow network links, enabling the optimization of the server-to-server protocol over the slow links.

3.2.1 Connectivity

A typical M-Link MU Gateway configuration would prefer use of faster and lower-latency SATCOM links when available, with slower but more reliable HF radio links being used when the SATCOM links are unavailable. This could be achieved by configuring Peer Controls for the appropriate domains with SATCOM (i.e. *XEP-0361*) links as the primary links, and the HF (SLEP) links as the fallback for the SATCOM links.

3.2.1.1 Link Fallback

A link may be configured with a fallback link, to be used when the parent link cannot be connected. This allows a chain of links (in order of preference) to be constructed. The appropriate Peer Control is configured to use the primary link, i.e. the link at the top of the chain.

When a connection to the domain handled by the Peer Control is required, M-Link MU Gateway will attempt to connect the Peer Control's primary link. Connection failures are handled by retrying at increasingly long intervals, until a limit is reached at which the link is considered failed. At this point M-Link MU Gateway will start connection attempts for the fallback link, applying the same retry logic. This proceeds until either

a connection is established using one of the links in the chain, or all links have been attempted and failed.

3.2.1.2 Link Fall-forward

It may be that a fallback link is in use by a Peer Control, but a more-preferred link for that Peer Control could now be established. For example, a SATCOM link is temporarily unavailable, leading to a less-preferred HF link being established instead, but then the SATCOM link becomes available again. When a less-preferred link is connected for a Peer Control, M-Link MU Gateway will regularly make connection attempts for the more-preferred link. If the more-preferred link is successfully connected, new outbound stanzas will be transferred using it. The less-preferred link will be closed after an idle period.

3.2.1.3 Timer Configuration

Connection establishment involves multiple steps. Timeouts and retry limits are configurable for many of these steps. Modification of the default settings may be required to optimize an M-Link MU Gateway installation.

The following options may be modified:

Remote Session Connection Timeout
: This configures the timeout which is applied to the initial TCP-level connect when establishing a session. This needs to be set to a value greater than the expected connection establishment latency.

Remote Session Connection Retry Base Delay
: This configures the basic delay applied before a failed connection attempt is retried. A simple algorithm is used which increases the delay by multiplying the previous delay interval by 1.5 each time a failure occurs. This is unlikely to need modifying.

Remote Session Connection Retry Delay Limit
: This configures the maximum delay interval between connection retries. Once the delay interval reaches this value (through multiplication as described above), it will not increase in length. This is unlikely to need modifying.

Remote Session Connection Retry Duration
: This configures the maximum length of time for which M-Link MU Gateway will perform connection retries before treating the connection as failed. This affects how quickly a fall-back link will be established, so should be set to be the length of time it is desired for connections to a primary link to be attempted and fail before falling back to a secondary link.

Remote Session Authentication Timeout
: Once a session has been opened to a Peer, authentication may be required. If this has not completed within this time, the session will be closed. This must be at least as long as authentication handshakes will take for all session types. If using TLS on a Link, it must allow for the latency to perform the several round-trips to establish TLS.

Remote Session Dialback Timeout
: This configures the maximum time which a remote server may take to verify its own dialback authentication request before we fail it. This is unlikely to need modification as dialback shouldn't be used over constrained bandwidth links.

STANAG S5066 Maintenance Timer
: SLEP sessions are multiplexed over a single TCP connection between the M-Link MU Gateway and a STANAG 5066 server. This timer controls how often a M-Link MU Gateway will attempt to reestablish this connection. It is unlikely that this will need modification.

Remote Session Idle Timer
: A session which has been idle (i.e. no stanza has been sent or received) for longer than this timer is closed. For sessions that are expensive to establish, but cheap

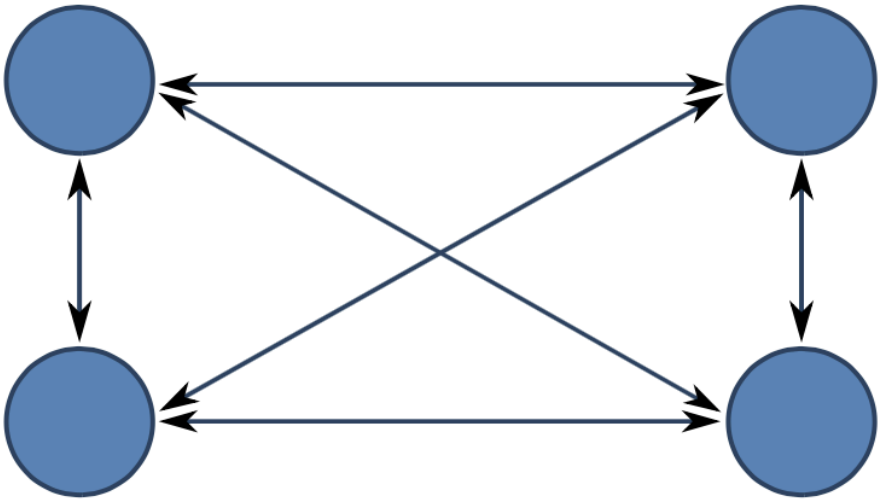
to maintain (especially those using TLS over constrained links) this should be moderately high (e.g. the default). For sessions that should be closed quickly when they are unused (such as SLEP links) this should be set lower.

Whitespace Ping Time
: This configures the length of time for which a session can be idle outbound (i.e. no stanza sent) before a single whitespace character is sent to check that the underlying TCP connection is still active. It is unlikely that this will need modifying.

Chapter 4 Peer Controls and Links

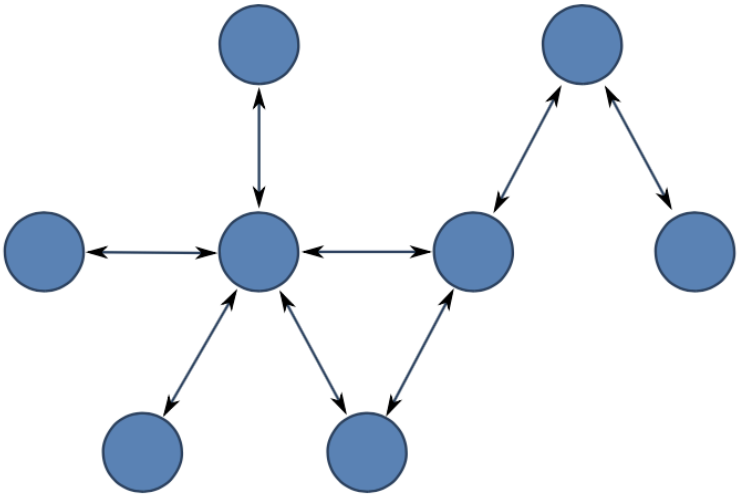
This chapter introduces the M-Link Edge Server product, explains what XMPP trunking is, and how M-Link Peer Controls can be used to support it. It also shows the use of *link*s to support operation with XML Guards.

Figure 4.1. Standard XMPP Server Configuration



In a standard XMPP configuration, XMPP servers are fully interconnected, as shown in Figure 4.1, “Standard XMPP Server Configuration”. This model works well for open XMPP deployments on the Internet. However, it does not work so well for cross domain, organizational boundary and constrained link scenarios.

Figure 4.2. XMPP Trunking



The model of XMPP Trunking, shown in Figure 4.2, “XMPP Trunking” extends the standard XMPP model to allow for configurations where servers are not fully connected. This structure enables XMPP messages to be switched through intermediate servers.

By default, when an XMPP server initiates a connection to another XMPP server (e.g., to other.organisation.example.com) it will look up the domain in the *Domain Name*

System (DNS), first with SRV records and falling back to A/AAAA records, and connect using the IP address determined from this lookup. This leads to the fully connected approach.

Peer Controls are the mechanism used by M-Link to enable configuration of an XMPP Trunking architecture. They are checked prior to the standard DNS lookup, and provide M-Link with information on how to connect to a specific domain. Each peering control relates to a specific remote domain (e.g. other.organisation.example.com), to a collection of subdomains (e.g. all subdomains of "isode.com"), or a default catch-all. A peering control can direct the connection to a list of hosts and/or IP address or specialised connection mechanism configured as part of the peering control. Peer controls can be used to reject the traffic to another site entirely, require particular security and authentication, and in some M-Link products restrict it to certain types and control the network connections used for the traffic.

Further information on XMPP Trunking and use of M-Link Peer Controls is provided in the Isode white paper Providing XMPP Trunking with M-Link Peer Controls [https://www.isode.com/whitepapers/xmpp-trunking.html].

4.1 M-Link Edge Server

M-Link Edge Server is an M-Link server with no local users that is used in boundary and cross-domain configurations to check and potentially modify traffic. M-Link Edge Server will often be used in conjunction with an XML Guard to provide cross-domain protection and may use *XEP-0361* or *GCXP* to communicate with the XML Guard.

M-Link and M-Link Edge Server are both provided by the same underlying technical product, but are sold as different products. Note that *XEP-0361* may not be used with a standard M-Link deployment that supports local users.

4.2 Peer Controls

Peer Controls define the rules for communicating with remote domains, and are checked prior to DNS being used. A Peer Control can be defined using one of three Matching Rules:

Domain
: A single specific domain. This is used to force routing for a specific domain.

Subdomains
: All subdomains of the domain, but not the domain itself. If this Matching Rule is selected for a Peer Control for `example.com`, domains that will match include `pubsub.example.com` and `chat.private.example.com`.

Domains and Subdomains
: The domain itself, and all its subdomains. If this Matching Rule is selected for a Peer Control for `example.com`, domains that will match include `example.com` and `pubsub.example.com`.

When multiple Peer Controls match a certain remote domain, the longest match is the one that will take effect. When no match is found for the domain, the *default* Peer Control will be applied.

This combination provides full flexibility to configure arbitrary XMPP trunking configurations.

When a peer control is in place, it is possible to configure specific actions for the peer:

- Chapter 6, *Security Labels, Clearances and Policies* checking against a configured security clearance for the peer.
- Section 4.3, “Peer Authentication” options.
- Section 4.4, “Relay zones”, which allows for relaying stanzas between servers.
- Traffic filtering using Chapter 7, *Transformations*, which enables removal or modification of traffic to and from the peer.
- Section 4.5, “JID Filtering”, which allows restricting whether stanzas are allowed based on their addressing.

Traffic to a remote domain will by default use S2S connections which are negotiated between the servers. If the option *Use Specialised Connection Mechanism* is enabled, communication will be sent and received using the specified link (see Section 4.6, “Links”).

SRV lookups for S2S can be overridden by enabling the *Override DNS* option and specifying values for the *Connect Host* and *Connect Port* options. *Connect Host* can either be an IP address, or a hostname. Multiple *Connect Host* and *Connect Port* pairs may be specified, with connection attempts being made in the order in which the pairs are specified.

If the option *Use Specialised Connection Mechanism* is enabled, communication will be sent and received using the specified link.

Note: Changing options on Peer Controls or Links may result in reestablishment of corresponding sessions and pending connections.

4.3 Peer Authentication

Authentication of remote servers is usually based on the addressing of the XMPP data that are being sent, not on the DNS name of the server hosting the remote service (although these are sometimes the same). There are two mechanisms for authenticating server to server (S2S) connections: Dialback (*XEP-0220*) and TLS authentication (SASL EXTERNAL).

4.3.1 TLS Authentication

TLS Authentication is based on the certificate presented by a remote server during TLS negotiation, and could take four forms.

- Normal. Absent further configuration, TLS Authentication is performed by checking the subjects of the certificate provided by the remote server, and if they match the server address (or Remote Host if a Link is in use) of the XMPP data that are being transmitted, checking to see if the certificate was issued by a trusted root certificate. For a detailed discussion, see Section 5.2, “Certificate Verification” and Section 5.3, “Trust Anchors”
- Connect Host. If `Peer Connections` is enabled through "Override DNS" and normal TLS Authentication is attempted and fails, M-Link will perform the same authentication checks:
  - Outbound. Using the hostname from the `Peer Connections` list which has resulted in the successful connection.
  - Inbound. Using each hostname from the `Peer Connections` list in turn.
- Remote Host. If and X2X or GCXP link is being used, M-Link will use the Remote Host value of the Link as the subject to check.

- Pinned Certificates. If a pinned peer or link certificate is set then TLS Authentication is performed only by comparing the certificate provided by the remote server to the one in this option.

4.3.2 Dialback

Dialback uses checks against the entries in DNS to authenticate a remote server (connections that use dialback for authentication may also use unauthenticated TLS, and if `TLS` is set to `Required`, must). This is generally considered a weaker form of authentication than TLS provides. Authentication must happen in both directions (that is: we must authenticate that the remote server is who they claim to be, and the remote server must authenticate that we are who we claim to be), and dialback is controlled independently for each direction. If `TLS` is set to any value other than `Require Authenticated TLS` we will be willing to authenticate a remote server's identity using dialback. If the `Perform Dialback When Offered` option is enabled then we will allow a remote server to authenticate our identity using dialback.

Where there is a choice of TLS Authentication or Dialback, M-Link will always prefer to use TLS Authentication.

4.4 Relay zones

Relay zones are the M-Link mechanism for allowing relaying of stanzas between servers in different logical groups (generally by bridging a network boundary of some sort, such that the servers would not be able to establish a connection otherwise). When receiving a stanza from a peer if the 'to' domain is not serviced by the local server a peer control matching the 'to' will be searched for and, if found and if the peer control of the 'to' gives a different relay zone from the peer control of the connection from which the stanza is received, it will be routed out to the target peer. This means that M-Link will not relay stanzas between servers within the same relay zone (because they should be communicating directly, or simply shouldn't be communicating).

4.5 JID Filtering

JID Filtering is a mechanism that controls which incoming stanzas from the peer should be discarded, based on the JIDs of the sender and recipient. M-Link has support for a global list of JID Filters, as well as filters on Peer Controls and IM Domains.

The `JID Filtering Mode` can be one of three values: `Disabled`, `Allow Specified Combinations`, and `Deny Specified Combinations`. If the mode is set on the Peer Control or IM Domain is set to `Disabled` (which is the default), the global JID Filtering rules will apply. Any incoming stanza that is rejected by the filter will be bounced with an error.

Error stanzas and IQ result stanzas will always be allowed through, regardless of the effective filtering mode.

Stanzas that have receipient that is in the exact same domain as the sender will not be rejected. This will generally only apply to filters that are in effect on IM domains.

4.6 Links

A Link defines a specialised connection mechanism for federation. Configuration of connection mechanisms and authentication/security live within the Link configuration for all Peers that use links, rather than in the Peer Control. M-Link currently supports these types of links:

X2X
: *Zero Handshake Server to Server Protocol (XEP-0361)* operating over TCP. This protocol is designed to support constrained bandwidth links with high latency, where standard S2S will lead to performance problems.

GCXP
: *Guard Content Exchange Protocol*, operating over TCP. This protocol is used for communications between M-Link and M-Guard.

  By default, GCXP links are configured to send and expect to receive GCXP responses, and M-Guard should be configured to let GCXP responses through. In setups where M-Guard cannot be configured to let GCXP responses through, the option *Send GCXP Responses* must be disabled as well. GCXP responses are used to provide acknowledgement of stanzas delivered between the M-Link Edge Server servers on either side of the Guard, allowing stanza delivery to be re-attempted in the case of a connection or system failure of the Guard or other M-Link Edge Server. They are also used by M-Guard to alert M-Link Edge Server when a stanza has been rejected by the Guard, allowing M-Link Edge Server to 'bounce' the stanza. GCXP responses contain no data other than the ID of the GCXP message to which they refer; connections to the Guard remain unidirectional with respect to the flow of XMPP data.

  Note: Peers using links that communicate using this protocol will generally require the *Normalize XML for M-Guard* and *Strip Legacy Delayed Delivery* Transformations.

SLEP
: Connections to other M-Link servers over HF Radio connections using *SIS Layer Extension Protocol (SLEP)* connections to Icon-5066 servers. More details on the configuration of this type of link can be found in Section 4.6.4, “SLEP Links”.

  Note: SLEP links are only available to M-Link MU Gateway and M-Link MU Server servers.

4.6.1 Disabling a Link

The configuration for each Link includes an `Enabled` item, defaulting to `true`. If set `false`, this Link will be ignored when attempting to open a new outbound connection. If a fallback link is configured, this will be used immediately instead, otherwise a connection failure will occur. The item has no effect on inbound connections or on outbound connections which are already established.

4.6.2 Connection parameters

All links can take a fallback link:

Fallback Link
: This option points to the next link that should be tried when an attempt to establish a connection fails. When this parameter is unset, no further links will be tried. This

can be used to create a list of links that will be tried in order when connecting to a peer.

Note: Peer Controls using a link with fallbacks must always point to the first link in the chain.

X2X and SLEP links support Stream Management (*XEP-0198*) acknowledgements (GCXP has its own mechanism for reliability in the form of *GCXP Responses*):

Enable XEP-0198
: When this option is enabled, the link will use Stream Management (*XEP-0198*) to ensure stanzas sent are received by the remote server. This will also enable acknowledgements of stanzas received from the remote server.

XEP-0198 Version
: This option selects which version of *XEP-0198* will be enabled on this link. By default, this will use the most recent version of the standard as supported by M-Link. When interoperating over a Stream Management enabled link with a server running a version of M-Link older than R19.0 , this option should be set to `Legacy XEP-0198`.

All X2X and GCXP links support the following parameters:

Local Port
: TCP port the link will listen on for new connections. The server will try to bind to this port on all local IP addresses.

Remote Host
: For non-Listen Only links, this indicates which host to connect to. This might either be an *Fully Qualified Domain Name* (FQDN), or an *IP address*.

Remote Port
: For non-Listen Only links, this indicates which port on Remote Host to connect to.

Use TLS
: When this option is enabled, and if a certificate is configured, the link will use and require TLS.

X2X links support the following additional parameters:

Listen Only
: When this option is set, the server will never initiate a connection over this link on its own, but will accept connections coming in. Established *Listen Only* links are always *Bidirectional*.

Bidirectional Connections
: If set, a single connection can send XML stanzas in both directions. If not set, a connection will be established for each direction of stanza flow.

When Use TLS is enabled, the following additional options are available:

TLS Identity
: Key and certificate this link presents to inbound connections, as well as sends to the peer when trying to establish a connection.

Pinned Link TLS Certificate
: If set, this link will only allow connections with peers that present this exact certificate.

Require a Valid Certificate in TLS Session
: When set, peers must use a valid certificate that isn't expired, matches the Remote Host, and when Pinned Link TLS Certificate is set, matches the pinned certificate.

  When this option is not set, any TLS certificate will be accepted, and TLS will only be used to prevent eavesdropping.

Ignore System Trust Anchors
: When set, only trust anchors explicitly configured within M-Link are trusted, and system trust anchors will be ignored.

Link Trust Anchors
: This is a list of TLS Trust Anchors that will be used for Section 4.6.3, “Authentication”.

4.6.3 Authentication

When using Links, authentication is based on the identity of the server to which the Link connects, not to the addressing of the XMPP data that are being transmitted. Unless the requireValidCertificates option is enabled on a Link (and TLS is configured) any connection associated with the Link is considered to be authenticated for transmission of XMPP data addressed to any Peer Control configured to use the Link. If the requireValidCertificates option is enabled then the certificates used in TLS must be trusted - either by the pinnedLinkCertificatePEM option matching the provided certificate, or by the presentation of a certificate whose subject (see Section 5.2, “Certificate Verification”) matches the remoteHost option and was issued by an authority configured as a trust anchor (see Section 5.3, “Trust Anchors”).

4.6.4 SLEP Links

Note: This section only applies to M-Link MU Gateway and M-Link MU Server.

To connect to other servers over HF Radio connections, M-Link MU Gateway supports the *SIS Layer Extension Protocol* (SLEP). This protocol is used to connect to remote sites over HF Radio connections using *Icon-5066* servers. Configuring links of this type takes two steps: creating a definition that will allow connecting to a *STANAG 5066 Subnet Interface Service* (SIS) connection to an Icon-5066 server, and creating a link that uses such a STANAG 5066 definition.

4.6.4.1 STANAG 5066 Server Definition

A configuration is required for every local Icon-5066 server. All fields are required, and must reflect the configuration of the Icon-5066 server. The *SAP ID* must be the same for the local and remote M-Link servers, as SLEP connections can only be established between two servers configured for the same *SAP ID*.

M-Link will try to maintain a connection to each defined STANAG 5066 server. If the connection breaks, M-Link will try to reconnect within the amount of time specified in the the *Advanced Connection Management* option *STANAG 5066 Maintenance Timer*.

4.6.4.2 Configuring SLEP Links

Once a STANAG 5066 Server Definition has been created, SLEP links can be configured. Multiple links may use the same STANAG 5066 Server Definition as long as the *Remote SIS Address* differs from other links using the same STANAG 5066 server.

Stanza acknowledgement using *XEP-0198* is available on SLEP links. Additionally, SLEP allows compression of the data sent over the radio connections. To configure this, both ends must enable the *Compress* flag in the link configuration.

Peer Controls using SLEP links will generally need Transformations to strip out (parts of) stanzas that are not strictly required in order to reduce the bandwidth required.

4.7 Reconnection Backoff and Retry

S2S and Link connections are subject to reconnection backoff and automatic retry. If a new connection is required (e.g. due to a stanza being queued for delivery to a Peer), a connection attempt will be made. If the connection attempt fails, subsequent attempts for the same connection will be initiated automatically after a delay. The delay increases by 50% on each subsequent connection failure, until the delay length reaches a limit, at which it remains constant.

When a connection succeeds, the associated delay is cleared.

When automatic reconnection attempts for an individual connection instance have been in operation for configurable length of time, further attempts at reconnection are abandoned, and connection failure is recorded.

Three system-wide configuration options allow configuration of the base retry delay, retry delay limit and duration of retry attempts:

Remote Session Connection Retry Base Delay
: This is the basic delay value, in seconds.

Remote Session Connection Retry Delay Limit
: This configures the maximum length of the retry delay, in seconds.

Remote Session Connection Retry Duration
: This configures the duration (in seconds) of automatic reconnection attempts for an individual failed connection.

For example, with a base delay of 10 seconds, a delay limit of 35 seconds and a retry duration of 90 seconds, reconnection attempts would be delayed as follows:

- First attempt: immediate
- Second attempt: delayed by 10 seconds
- Third attempt: delayed by 15 seconds
- Fourth attempt: delayed by 23 seconds
- Fifth attempt: delayed by 35 seconds
- Sixth attempt: delayed by 35 seconds, when this fails no more retries will be attempted, as the total retry duration from the first attempt is greater than 90 seconds

Chapter 5 TLS Configuration

This chapter describes how M-Link is configured to use TLS, covering configuration of certificates/keys used by the local server, and trust of certificates used by other servers.

Note: Your M-Link installation must have the TLS feature enabled before TLS can be used.

Transport Layer Security (TLS) is used by M-Link to provide privacy, data integrity and authentication for client-to-server and server-to-server connections. Use of TLS requires the configuration of three main classes of information:

Public Key Certificate Chain
: A Public Key Certificate Chain certifies the ownership of a public key (by the named subject of the certificate).

Private Key
: The Private Key corresponding to the Public Key Certificate Chain noted above.

Trust Anchors
: These are a set of certificates which are to be treated as authoritative entities for which trust is assumed.

A Public Key Certificate Chain and corresponding Private Key must be configured before M-Link can act as a TLS server (that is to say, the responding side of a TLS handshake) and provide a TLS-encrypted connection.

The same Public Key Certificate Chain and Private Key will be used by M-Link when it is acting as a TLS client (initiating a TLS exchange with a TLS server).

Whether the Public Key Certificate Chain provided by M-Link in the TLS handshake also provides authentication depends on whether the Certificate Chain has been signed by a Trust Anchor which the other party to the TLS handshake trusts.

The Trust Anchor list identifies a set of trusted signing entities which will be used when verifying the validity of a Certificate Chain. The configured list may be used in addition to Trust Anchors provided externally to M-Link (installed as part of the operating system, or placed in system-wide certificate stores), or can replace these external Trust Anchors completely.

Private Keys are normally stored in encrypted form. The passphrase which will be used when decrypting the Private Key must be specified when the Private Key is uploaded, and is then stored in encrypted form for use when needed.

A number of secondary TLS controls are available:

Minimum TLS level supported
: This configures the minimum TLS level which will be supported. The default is TLS version 1.2.

Cipher Suites
: The set of cipher suites which will be proposed or accepted during TLS handshaking can be configured, as a standard OpenSSL cipher string. This is empty by default, allowing any supported cipher suites to be used.

Ignore System Trust Anchors
: Disables use of built-in Trust Anchors; only those Trust Anchors which have been configured for M-Link will be used. This defaults to false.

Default Trust Anchors
: This allows configuration of a set of Trust Anchors to be used by the server.

Public Key Certificate Chains, Private Keys and Trust Anchors are uploaded into the appropriate Object Stores using the Configuration tab of the M-Link Administration

Interface, from PEM-encoded files. Many Certification Authorities provide certificates in this form anyway, and tools are readily available to convert certificates which have been provided in other formats.

5.1 Detailed Configuration

TLS information can be specified at various points within M-Link's configuration, as described below.

5.1.1 Default Settings

Default TLS configuration can be specified for an M-Link server as a whole. In a simple setup this may be all that is required:

Default TLS Identity

Minimum TLS Level supported

Cipher Suites

Ignore System Trust Anchors

Default Trust Anchors

Child Service TLS
: This enables TLS for Child Service connections.

Child Service TLS Identity
: This configures the TLS identity which will be used for Child Service connections.

Use TLS by Default for Links
: This configures the default for the per-Link "Use TLS" option.

Require TLS by Default for IM Domains
: This configures the default for the per-IM Domain "TLS Required" option.

Enable Certificate Revocation Checks
: This enables use of Certificate Revocation Checking. Note that this option applies to use of TLS by all configured Server-to-Server connections.

5.1.2 Certificate Revocation Configuration

If Certificate Revocation checking is enabled, certificates presented during TLS exchanges are validated using external resources to verify that they have not been revoked by their issuing Certificate Authority before the official expiry date encoded within the certificate. Two methods for revocation checking are provided:

- LDAP-based Certificate Revocation List lookup, where M-Link reads a list of revoked certificates from a Certificate Authority and verifies that the certificate being checked is not included.
- Use of the Online Certificate Status Protocol (OCSP), where the Certificate Authority is directly queried for the revocation status of a specific certificate.

The primary LDAP and OCSP servers to be used for revocation checking can be configured. Certificates may identify additional LDAP and OCSP resources via various extensions. The use of these extensions can be controlled via configuration flags.

Where various options for revocation checking are available, they are tried in the following order until one returns a valid result:

- An OCSP query using the configured OCSP server (if any), unless disabled.
- An OCSP query using information from extensions within the certificate (if any), unless disabled.

- An LDAP CRL lookup using the configured LDAP server (if any), unless disabled.
- An LDAP CRL lookup using information from extensions within the certificate (if any), unless disabled.

LDAP Directory Hostname for CRL Retrieval
: This configures the address of an LDAP Directory which will be used for CRL retrieval.

LDAP Directory Port for CRL Retrieval
: This configures the port of an LDAP Directory which will be used for CRL retrieval.

Certificate Revocation Check Type
: This configures the type of Certificate Revocation check which is performed. The options are "Do Not Perform Certificate Revocation Checks" to prevent revocation checks, while basic validity (e.g. expiry time) and Trust Anchor checking will still be performed, "Check All Certificates in Chain" and "Only Check End-Entity Certificates".

Use OCSP Nonce
: This enables use of a random number (a "nonce") which is sent in an OCSP request and included in the corresponding OCSP response, to help prevent replay attacks.

Enable OCSP URI
: This enables the configuration (and use) of an OCSP URI.

OCSP Query URI
: This configures the HTTP URI which will be used for OCSP lookups.

OCSP Responder Certificate
: This configures the certificate with which M-Link expects OCSP responses to be signed. If unspecified, no signing check will be performed on OCSP responses.

Don't use OCSP URIs from Certificate Extensions
: Certificates may contain authorityInfoAccess extensions. These extensions may in turn specify OCSP URIs which are to be used as part of the validation process. Setting this option disables the use of such URIs.

Don't get CRLs from configured LDAP server
: This option disables the use of the configured LDAP server to retrieve CRLs.

Don't get certs from configured LDAP server
: This option disables the use of the configured LDAP server to retrieve certificates. If an entry is being read to retrieve CRLs, then certificates will also be read.

Don't use URIs from extensions to look up CRLs
: This option disables the use of the CRL DP, ARL DP and freshestCRL extensions which may be present in a certificate.

Don't use URIs from extensions to look up certificates
: This disables following authorityInfoAccess extensions to find certificates.

Ignore freshestCRL extensions
: This disables the use of freshestCRL extensions in certificates and CRLs.

Always use HTTP POST for OCSP requests
: This disables use of HTTP GET for OCSP requests (by default, encoded requests smaller than 255 bytes will be made using HTTP GET).

5.1.3 IM Domain Settings

The configuration for an individual IM domain may specify TLS configuration which overrides the default settings:

TLS Required
: If this is set true, TLS will be required for all client connections to this IM domain.

TLS Identity
: Specifies the TLS identity to be used for this IM domain.

5.1.4 Pubsub Domain Settings

The configuration for an individual Pubsub domain may specify TLS configuration which overrides the default settings:

TLS Identity
: Specifies the TLS identity to be used for this Pubsub domain.

5.1.5 MUC Domain Settings

The configuration for an individual MUC domain may specify TLS configuration which overrides the default settings:

TLS Identity
: Specifies the TLS identity to be used for this MUC domain.

5.1.6 Peer Control Settings

Peer Control TLS settings have the effect of overriding the TLS settings for the IM, Pubsub or MUC domains to which they apply (and the default settings), when performing server-to-server (S2S) communication:

TLS
: This item controls how TLS is used by the S2S link, and takes the values:

  - Never use TLS
  - Use TLS Whenever Possible
  - Required TLS
  - Require Authenticated TLS

TLS Identity
: Specifies the TLS identity to be used.

Pinned Peer Certificate
: See below for a detailed description

Ignore System Trust Anchors
: Disables use of built-in Trust Anchors; only those Trust Anchors which have been configured for M-Link will be used. This defaults to false.

Peer Trust Anchors
: This allows configuration of a set of Trust Anchors to be used for this Peer Control.

A Pinned Peer Certificate allows the standard certificate verification (performed during the TLS handshake) to be overridden. Instead of the certificate presented by the Peer being verified against Trust Anchors, checked for revocation etc, a simple comparison with the Pinned Certificate is performed, and if the presented and Pinned certificates match, the authentication is successful.

5.1.7 Link Settings

Link TLS settings have the effect of overriding any default TLS settings:

Use TLS
: If this is set true, TLS will be used and required for this Link.

TLS Identity
: Specifies the TLS identity to be used.

Pinned Link TLS Certificate
: See below for a detailed description

Require a Valid Certificate in TLS Session
: If this is set true, a TLS session must present a valid certificate.

Ignore System Trust Anchors
: Disables use of built-in Trust Anchors; only those Trust Anchors which have been configured for M-Link will be used. This defaults to false.

Link Trust Anchors
: This allows configuration of a set of Trust Anchors to be used for this Link.

A Pinned Link Certificate allows the standard certificate verification (performed during the TLS handshake) to be overridden. Instead of the certificate presented by the Link being verified against Trust Anchors, checked for revocation etc, a simple comparison with the Pinned Certificate is performed, and if the presented and Pinned certificates match, the authentication is successful.

5.2 Certificate Verification

A certificate which is presented as part of a TLS handshake is verified via a multi-stage process. The first stage takes place during the handshake itself, and checks (among other things) that the certificate has been signed by one of the Trust Anchors which have been configured for the current TLS context and (optionally) has not been revoked by the Certificate Authority which generated it. Even if the certificate fails one or more of these checks, the TLS handshake may still complete successfully, with an encrypted TLS session being established.

Once the TLS handshake is complete, secondary checks are performed on the presented certificate, if the configuration of the domain or link requires a valid certificate. These are:

- If a Pinned Certificate is configured, the results of the first-stage verification are ignored and a direct comparison between this and the presented certificate is performed. If they match, the presented certificate is considered valid. If they do not match, the presented certificate is considered invalid. In either case, no further verification of the presented certificate is performed.
- A check of the result of the first-stage verification, described above. If this first-stage verification has failed, no further action is taken, and the certificate is considered invalid.
- For Server-to-Server connections which use a Peer Control, a Subject Alternative Name match is attempted for the XMPP domain to which the connection is being made. If this fails, a Subject Alternative Name match for the Peer Control's connectHost configuration setting is attempted. If both of these matches fail, the certificate is considered invalid.
- For Server-to-Server connections which use a Link, a Subject Alternative Name match against the Link's remoteHost configuration setting is attempted. If this match fails, the certificate is considered invalid.
- If the appropriate checks described above succeed, the certificate is considered valid.

A certificate can contain multiple Subject Alternative Names, of varying types. When attempting to match a domain name, DNS name or hostname against these, a number of different comparisons are performed:

- A match against one of the certificate's DNS Names. This includes wildcard matching, so that a certificate with a DNS Name of `*.isode.com` would match `mary.isode.com`.
- A match against one of the certificate's SRV Names. SRV Names are prefixed with `_xmpp-server` if the certificate belongs to an XMPP server or `_xmpp-client` if presented by an XMPP client. Thus an SRV name of `_xmpp-server.mary.isode.com` would match the domain `mary.isode.com`.

- A match against one of the certificate's XMPP Addresses.
- If the certificate has no other Subject Alternative Names, a match against one of the certificate's Common Name values.

5.3 Trust Anchors

Trust Anchors are certificates which identify trusted signing entities. These are used by M-Link to verify that a chain of certificates (up to and including an end-entity certificate) received from another XMPP server or client is valid.

Most operating systems provide a built-in set of Trust Anchors which identify commercial Certification Authorities. The location and format of these is system-specific. By default, M-Link will make use of these Trust Anchors. Use of system Trust Anchors can be overridden via configuration either for the whole M-Link installation or per-Peer Control or per-Link.

A set of private Trust Anchors may be specified as part of M-Link's configuration, either across the whole installation, per-Peer Control or per-Link. Use of private Trust Anchors is required when the end-entity certificates being presented have been signed by Certification Authorities whose CA certificates are not configured as part of the operating system.

Chapter 6 Security Labels, Clearances and Policies

This chapter describes how Security Labels, Clearances and Policies are configured and used within M-Link.

A Security Label is a structured representation of the sensitivity of a piece of information (e.g. a message). This is used in conjunction with a Security Clearance (a structured representation of what information sensitivities an entity is authorized to access) and a Security Policy to control access to each piece of information. For instance, a message stanza could be labelled as "SECRET", hence requiring the sender and the receiver to each have a clearance granting access to "SECRET" information. Labels include a Display Marking which provides a human-readable form of the machine-readable label, as well as information about which colours XMPP clients should use when displaying the label.

The use of Security Labels in XMPP is described in detail in XEP-0258: Security Labels in XMPP. General information on Isode's Security Policy Infrastructure can be found at Isode Security Policy Infrastructure and its use by Isode within XMPP is described in Using Security Labels to Control Message Flow in XMPP Services.

Determining whether a labelled message stanza is permitted to be transferred to an entity (i.e. a domain or person) is a two stage process:

- The label associated with the destination entity is tested against the clearance associated with the message stanza's sender entity. This is testing "Is the sender of this message cleared to send a message to this entity?".
- The label on the message stanza is tested against the clearance associated with the destination entity. This is testing "Is the recipient of this message cleared to receive a message with this security label?".

Each of the tests described above takes place within the context of a Security Policy. The Security Policy controls how the Label/Clearance comparison is performed, and may provide other facilities such as equivalences to other Security Policies. When Security Label conversion is taking place, two Security Policies will be used: the Policy under which the Label was issued and the Policy to which the Label will be converted.

Each domain which originates or accepts message stanzas may be configured with its own Security Label and Clearance. A system of defaulting within the M-Link configuration means that, in the simplest case, a single server-wide Security Label and Clearance are all that need to be configured. For more complex systems, individual domains can override defaults with their own settings.

A final configuration item which may be associated with each domain entity is a Default Stanza Label. This is the Security Label which will be applied to any unlabelled message stanza originating at that entity.

IM Domain users may be configured with their own Clearance (see Client Security Clearance Configuration); if not set, this falls back to the IM Domain User Default Security Clearance.

Security Labels and Clearances are maintained within catalogs (a Label Catalog is a collection of associated labels; a Clearance Catalog is a collection of associated clearances). Multiple label and clearance catalogs may be loaded into an M-Link configuration. Specific label and clearance catalogs may then be associated with individual components of the server's configuration, and individual Security Labels and Clearances selected for use from within these catalogs.

6.1 Object Stores

Security Policies, Security Label catalogs and Clearance catalogs are maintained in type-specific Object Stores within M-Link. All three types of entity are usually maintained and distributed publicly as XML documents. The M-Link Administration Interface allows these XML documents to be loaded into the appropriate Object Stores. Catalogs, Labels and Clearances within catalogs, and Security Policies can then be selected for use in configuration options.

Security Policies are represented within M-Link as SDN.801c Security Policy Information Files in Open XML SPIF format. Isode provides sample Security Policies and associated Security Label and Clearance catalogs in subdirectories of $(SHAREDIR)/security_label/example_data.

6.2 Server Configuration

The global "Use Security Labelling" option within the M-Link Administration Interface controls whether any Security Labelling configuration is exposed and enabled.

Note: If Security Labelling has been enabled, but no Clearance has been configured for the Server, all messages which contain Security Labels will be blocked.

Within the Security Labelling Configuration section of the M-Link Administration Interface, the following items may be configured:

Server Default Security Policy
: The default Security Policy to be used by the server. This is a selection from the set of policies which have been loaded into the Security Policy Object Store. A Security Policy will normally define a default Security Label and Clearance; if this is the case, the defaults will be used as described in Security Policy Default Label and Clearance.

Server Default Security Label
: This is the default Security Label to be used by the server. It is selected in a two-stage process; first a Security Label Catalog is chosen from the Security Label Catalog Object Store, and then an individual Security Label is chosen from within that catalog. This allows the transit of messages from originators with unsuitable clearances to be blocked.

Stanza Default Security Label
: This is the default Security Label which is applied to unlabelled stanzas. It is selected in a two-stage process; first a Security Label Catalog is chosen from the Security Label Catalog Object Store, and then an individual Security Label is chosen from within that catalog. The Security Label Catalog defaults to that chosen for the Server Default Security Label.

Server Default Security Clearance
: This is the default Security Clearance to be used by the server. It is selected in a two-stage process; first a Security Clearance Catalog is chosen from the Security Clearance Catalog Object Store, and then an individual Security Clearance is chosen from within that catalog. The Clearance blocks the transit of messages without a suitable label.

Default XEP-0258 Label Format
: This controls the encoding in which Security Labels will be emitted.

Require Security Label on Inbound Message
: This mandates that a Security Label is present on all inbound message stanzas.

Reject Inbound Messages With Unrecognized Security Labels
: This option controls how messages which include valid but unrecognized labels are handled. The default behavior is to reject such messages. Disabling this option causes such messages to have the label discarded and to then be treated as unlabelled.

User Default Security Clearance
: This is the default clearance which will be applied to Users. It may be overridden by per-IM Domain and per-User configuration.

Update Display Markings
: Security Labels include a display marking phrase (e.g. "Secret") and optionally foreground and background color specifications. The default behavior of M-Link is to normalize these fields when serializing a label, replacing values present in the label with those defined for the label by the Security Policy. This control allows this functionality to be disabled, so that labels are emitted with the display markings that they arrived with.

6.3 Security Policy Default Label and Clearance

A security policy may designate one of the classifications within the policy as the default, and derive a clearance and label from this classification. These are used by M-Link as the defaults of last resort when performing access control decisions.

For example, if a message is being delivered to an IM domain, the message's clearance will be tested against a label associated with the IM domain. A security label may be explicitly configured for the IM domain, or it may inherit the server's explicitly configured security label. If neither of these are set, the default label (if any) defined by the security policy will be used. In the same way, if the message does not have an associated clearance, the default clearance defined by the security policy will be used in the access control decision, and if the message is not labelled, the default label configured for the domain will be used, falling back to the security policy default label.

The same defaulting is used when checking that a user has sufficient clearance for an IM domain's label during association establishment to provide defaults for the user clearance and IM domain label.

6.4 Peer Control Configuration

Individual Peer Controls may have their own Security Labelling configuration. This provides a mechanism for controlling the classification of messages which can flow to and from an external domain. For example, a simple configuration might allow messages labelled as Unclassified, Sensitive or Confidential to be sent between users within the same local IM domain, but only permit Unclassified messages to be passed to external domains.

Individual Peer Controls may also have their own Security Policy configured. When this is the case, messages flowing to the Peer Domain will have their labels converted so that they are consistent with Security Policy which has been configured for the Peer.

If an inbound message includes a label which has not been issued under either the Server Security Policy or the Peer's own Security Policy (if configured), then the label will be discarded and the message will be treated as if unlabelled. An Audit record will be generated to record the label discard. See Security Label Handling for more information.

The following items can be configured:

Peer Security Label
: This is the Security Label applied to the Peer: only stanzas with a source clearance which permits access to this Security Label will be transferred to the Peer. The label is selected in a two-stage process; first a Security Label Catalog is chosen from the Security Label Catalog Object Store, and then an individual Security Label is chosen from within that catalog. When setting this option, the catalog choice will default to that selected for the Server as a whole, but the Peer Security Label does not have a default and must be explicitly selected, since it would normally be different from that of the Server.

Peer Default Stanza Security Label
: This is the default Security Label which is applied to unlabelled stanzas arriving from this Peer. It is selected in a two-stage process; first a Security Label Catalog is chosen from the Security Label Catalog Object Store, and then an individual Security Label is chosen from within that catalog. The Security Label Catalog choice defaults to that selected for the Server as a whole, and the Peer Default Stanza Security Label choice defaults to the Server's Stanza Default Security Label.

Peer Outbound Default Stanza Security Label
: This allows the configuration of a Security Label which will be applied to all unlabelled outbound messages for this Peer.

Peer Security Clearance
: This is the Security Clearance applied to the Peer: only stanzas which include a label which the Peer's clearance allows will be transferred to the Peer. The clearance is selected in a two-stage process; first a Security Clearance Catalog is chosen from the Security Clearance Catalog Object Store, and then an individual Security Clearance is chosen from within that catalog. The Clearance Catalog defaults to that selected for the Server as a whole, while the Peer Security Clearance itself must be explicitly selected, since it would normally be different from that of the Server.

Peer Relabelling Security Policy
: This is the Security Policy which applies to the Peer. If this is set (and different from the Server Default Security Policy), the label on an outbound message (issued under the Server Default Security Policy) will be replaced with an equivalent label issued under the Peer Relabelling Security Policy. For this to be possible, the Server Default Security Policy must have equivalences for the Peer Relabelling Security Policy configured. See Security Label Handling for more information.

Peer Outbound Security Label Format
: This controls the format in which outbound Security Labels are emitted. It defaults to the server-wide Default XEP-0258 Label Format setting.

Peer Update Display Markings
: Security Labels include a display marking phrase (e.g. "Secret") and optionally foreground and background color specifications. The default behavior of M-Link is to normalize these fields when serializing a label, replacing values present in the label with those defined for the label by the Security Policy. This control allows this functionality to be disabled, so that labels are emitted with the display markings that they arrived with.

Require Security Label on Inbound Messages
: This mandates that a Security Label is present on all inbound message stanzas from a Peer.

Reject Inbound Messages With Unrecognized Security Labels
: This controls whether an unrecognized Security Label causes an inbound message to be rejected, or whether the Security Label is stripped off and the unlabelled message is allowed to proceed.

Reject Inbound Messages With Unrecognized Security Labels
: This option controls how messages which include valid but unrecognized labels are handled. Disabling this option causes such messages to have the label discarded and to then be treated as unlabelled.

Peer Mandatory Label
: This allows configuration of an IC-ISM format label which will be required by default to be present on all inbound message stanzas from the Peer. Messages which do not contain a matching label will be rejected. A label match will cause the label to be removed from the message stanza and the Peer Default Stanza Security Label (if any) to be applied to it. The label may also be applied to outbound messages to the Peer. This is a specialized option and should only be used when suggested by Isode Support. See Security Label Handling for more information.

Require Peer Mandatory Label on Inbound Messages
: The default value of this option requires the label configured in the Peer Mandatory Label option to be present in all inbound messages. See Security Label Handling for more information.

Apply Peer Mandatory Label to Outbound Messages
: This causes the label configured in the Peer Mandatory Label option to be applied to all outbound messages, replacing any other label present.

Add FLOT to Outbound Messages
: This configures the conditions under which a First Line Of Text marking is added to the body of messages, and is intended for use when sending messages to Peers which do not support XEP-0258 labeling. The FLOT marking added is derived from the message's Security Label. If the message also includes an HTML payload, this will be stripped. The "Add FLOT if message has non-default Label" option will cause the Security Label on the message to be compared against the Peer Outbound Stanza Default Security Label, if set, otherwise against the label marked as Default in the Security Policy, if any. If a match is detected, no FLOT will be inserted.

Peer Fixed Security Label
: This configures a fixed Security Label (chosen from a Label Catalog) which will be applied to all outbound messages, replacing any Security Label present on the message.

Peer Disable Label Out
: If set to true, outbound messages will not include a security label. Access control checks will still be performed if configured.

6.5 IM Domain Security Configuration

Individual IM Domains may have their own Security Labelling configuration. This provides a mechanism for controlling the classification of messages which can flow to and from an internal domain.

If a message includes a label which has not been issued under the Server Security Policy, then the label will be discarded and the message will be treated as if unlabelled. An Audit record will be generated to record the label discard.

The following items can be configured:

IM Domain Security Label
: This is the Security Label applied to the IM domain: only stanzas with a source clearance which permits access to this Security Label will be permitted to be sent to or from the domain. The label is selected in a two-stage process; first a Security Label Catalog is chosen from the Security Label Catalog Object Store, and then an individual Security Label is chosen from within that catalog; this defaults to the Server Default Security Label.

IM Domain Default Stanza Security Label
: This allows the configuration of a Security Label which will be applied to all unlabelled messages sent from this IM domain.

IM Domain Security Clearance
: This is the Security Clearance applied to the IM domain: only stanzas which include a label which the domain's clearance allows can be sent to or from the domain. The clearance is selected in a two-stage process; first a Security Clearance Catalog is chosen from the Security Clearance Catalog Object Store, and then an individual Security Clearance is chosen from within that catalog; this defaults to the Server Default Clearance.

IM Domain User Default Security Clearance
: This is the default Security Clearance applied to users within this IM domain. The clearance is selected in a two-stage process; first a Security Clearance Catalog is chosen from the Security Clearance Catalog Object Store, and then an individual Security Clearance is chosen from within that catalog; this defaults to the server's User Default Security Clearance.

IM Domain Update Display Markings
: Security Labels include a display marking phrase (e.g. "Secret") and optionally foreground and background color specifications. The default behavior of M-Link is to normalize these fields when serializing a label, replacing values present in the label with those defined for the label by the Security Policy. This control allows this functionality to be disabled, so that labels are emitted with the display markings that they arrived with.

Add FLOT to Messages Before Delivery
: This configures the conditions under which a First Line Of Text marking is added to the body of messages before delivery to IM Domain users. The FLOT marking added is derived from the message's Security Label. If the message also includes an HTML payload, this will be stripped.

Strip Security Label from Messages Before Delivery
: If set to true, messages will not include a security label when delivered. Access control checks will still be performed if configured.

IM Domain Outbound Security Label Format
: This controls the encoding in which Security Labels will be emitted for users within the IM domain.

IM Domain XEP-0258 Label Catalog
: This is the Security Label Catalog which will be used to generate the XEP-0258 label catalog which is served to the IM domain's clients. It is chosen from the Security Label Catalog Object Store. It defaults to the catalog from which the Server Default Security Label has been chosen.

6.6 MUC Domain Configuration

Individual MUC Domains may have their own Security Labelling configuration. This provides a mechanism for controlling the classification of messages which can flow into MUC rooms within the domain.

The following items can be configured:

MUC Domain Security Label
: This is the Security Label applied to the MUC domain: only stanzas with a source clearance which permits access to this Security Label will be permitted to be sent to MUC rooms within the domain. The label is selected in a two-stage process; first a Security Label Catalog is chosen from the Security Label Catalog Object Store, and then an individual Security Label is chosen from within that catalog; this defaults to the Server Default Security Label.

MUC Domain Default Stanza Security Label
: This allows the configuration of a Security Label which will be applied to all unlabelled messages sent to this MUC domain.

MUC Domain Security Clearance
: This is the Security Clearance applied to the MUC domain: only stanzas which include a label which the domain's clearance allows can be sent to MUC rooms within domain. The clearance is selected in a two-stage process; first a Security Clearance Catalog is chosen from the Security Clearance Catalog Object Store, and then an individual Security Clearance is chosen from within that catalog; this defaults to the Server Default Clearance.

6.7 Audit Logging

Access Control Decision Function (ACDF) failures are audited by M-Link. Three different audit types are generated:

acdfSourceFailure
: This indicates that either the source of the stanza did not have a Clearance, or that the processing entity's Label is not permitted by the Clearance.

acdfLabelFailure
: This indicates that the stanza's Label is not permitted by the processing entity's Clearance.

clientAcdfFail
: This audit is logged when a client is binding to an IM Domain, and indicates that the IM Domain's Label is not permitted by the client's Clearance.

In the first two cases, the stanza's To and From address fields are included in the audit record, together with the name of the domain for which the ACDF is being called. In the case of an unexpected ACDF failure, this should enable the identification of configuration point responsible for the failure. A free-form error string indicating the reason for the ACDF failure is also included. For a ClientAcdfFail, the JID of the client is included.

6.8 Security Label Handling

This section summarizes how an inbound message, which may be labelled or unlabelled, is handled when Security Labelling is enabled. The actions listed below are taken, in order.

- If the message has arrived via a Peer Control, and a Peer Mandatory Label is configured for the domain, and the domain's Require Peer Mandatory Label on Inbound Messages option is set True, messages which are unlabelled or whose label does not match the configured Peer Mandatory Label will be rejected. A message with a matching label will have the label stripped and the unlabelled message will be allowed to continue.
- If the message includes a label, this is extracted. Failure to extract the label (e.g. because it is badly formatted) will cause the message to be rejected.
- If the message has entered via a Peer Control which has a Peer Relabelling Security Policy configured, the label is validated against this alternative policy, and if it passes validation, an Equivalent Label is generated under the Server Security Policy, replacing the message's original label. If the label does not pass validation against the alternative policy, the label is discarded and the unlabelled message is allowed to

continue. If an Equivalent Label cannot be generated, the original label is discarded, and the unlabelled message is allowed to continue.

- The label (or Equivalent Label) is validated against the Server Security Policy. If it passes validation, the message is allowed to continue, with the label attached. If validation fails, the label is discarded and the unlabelled message is allowed to continue.

Following label acquisition and validation, checks for label presence are performed:

- If the message has arrived via a Peer Control or IM Domain which has the Require Security Label on Inbound Messages option set True, and does not now contain a label, it will be rejected.

The final stage of label handling on ingress is to set a default label. If the message does not contain a label, a default label may be applied, using the following rules:

- If the destination domain is a MUC domain, apply the domain's MUC Domain Default Stanza Security Label.
- If the source domain is non-local, apply the appropriate peer control's Peer Default Stanza Security Label.
- If the source domain is a local IM domain, apply the domain's IM Domain Default Stanza Security Label.

6.9 XEP-0258 Label Catalogs

M-Link supports XEP-0258 label catalog discovery. The label catalog which is provided to a client in response to a catalog discovery request is generated as follows:

The IM Domain XEP-0258 Label Catalog configured for the client's IM domain is used as the basis for what will be returned to the client. This is then subject to further filtering which removes any label categories which would not be permitted by the following clearances:

- The clearance configured for the destination IM, MUC or Peer Domain. If the destination does not have a clearance configured, an empty label catalog is returned to the client.
- The clearance configured for the source IM domain: IM Domain Security Clearance.
- The clearance configured for the source client: Client Security Clearance Configuration. If no source client clearance is set, the IM Domain User Default Security Clearance is used, if set.
- If the target is a client within a local IM domain, the clearance configured for the target client: Client Security Clearance Configuration. If no client clearance is set for the target, the IM Domain User Default Security Clearance for the target domain is used, if set.

6.10 Clearances

A message is assigned a Clearance on ingress to the M-Link server. If the message arrives via a Peer Control, it is assigned the Peer Control's Peer Security Clearance. If ingress is via an IM Domain (i.e. from a client), the message is either assigned the client's Security Clearance (see Client Security Clearance Configuration) or the IM Domain's IM Domain User Default Security Clearance.

6.10.1 Client Security Clearance Configuration

For IM domains which are configured with an Authentication Type of Static Users, a user can be assigned a clearance by setting their User Security Clearance option.

When LDAP authentication is used, the user's clearance will be held as an attribute within the user's Directory entry. M-Link looks for various attribute types, trying the values returned for each attribute type until one which can be successfully interpreted as a valid Clearance issued under the Server Default Security Policy is found. Three attribute types are attempted, in the order shown below:

- An X.509 certificate, held in a userCertificate attribute. The clearance itself should be held within the X.509 certificate in a Subject Directory Attributes Extension.
- An XML clearance, held in an sioClearance attribute.
- An X.509 clearance, held in a clearance attribute.

6.11 Client IM Domain Access Control

If an IM Domain has been configured with an IM Domain Security Label, only clients which have been configured with a clearance allowing access to the label will be able to authenticate. See Client Security Clearance Configuration. for more information.

Chapter 7 Transformations

This chapter describes transformations that may be configured for Peer Controls and locally served domains.

Transformations can be applied to stanzas coming in from or going out to peers and clients, as well as stanzas going to local *Multi-User Chat* (MUC) and *PubSub* services. The configurations of each of these can have a number of these Transformations, which are executed in the order they are listed. One typical case where this may be useful is when network capacity is constrained: for example, on a slow link it may make sense to set up a transformation to strip all Chat State Notifications.

M-Link also supports a global list of transformations. Global transformations are applied by any entity performing transformations if the `applyGlobalTransformations` option is enabled, which it is by default. Global inbound transformations are performed after the transformations defined on the domain or peer control, global outbound transformations are applied before the domain or peer control specific transformations. Global transformations that are configured for both directions will be applied to stanzas twice, once at the source, and once at the destination.

Transformations are selected from the Transformations object store. Each transformation is an *XSLT 1.0* document operating on a single stanza in the `jabber:server` namespace, wrapped in a minimal XMPP Server-to-Server stream element. The result of a transformation should also be a single stanza wrapped in a stream element.

M-Link ships with transformations for common operations, such as stripping XHTML-IM (*XEP-0071*) elements from messages, or normalizing XML for communications to M-Guard. These transformations are always present, and cannot be removed or renamed, however uploading variants of the built-in rules with modified behaviour is possible. A complete list of all transformations shipped with M-Link can be found in Appendix C, *Pre-defined Transformations*.

The wizard which allows user-defined *XSLT 1.0* documents to be uploaded to the Transformation object store also allows simple transformations which strip namespace / element combinations to be created and uploaded.

Example 7.1. A stanza wrapped for transformations

```
<stream:stream id='1' version='1.0' xmlns='jabber:server'
    xmlns:stream='http://etherx.jabber.org/streams'>
<message from='anne@example.com' to='bob@example.net' type='chat'>
  <body>Hello!</body>
  <active xmlns='http://jabber.org/protocol/chatstates'/>
</message>
</stream:stream>
```

Example 7.2. An XSLT that will strip Chat State Notifications from message stanzas

```
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:stream='http://etherx.jabber.org/streams'
    xmlns:xmpp='jabber:server'
    xmlns:xhtml-im='http://jabber.org/protocol/xhtml-im'
    xmlns:csn='http://jabber.org/protocol/chatstates'>

 <xsl:output method="xml" omit-xml-declaration="yes"/>

 <xsl:template match="@*|node()">
  <xsl:copy><xsl:apply-templates select="@*|node()"/>
 </xsl:copy></xsl:template>
```

```
 <xsl:template match="/stream:stream/xmpp:message/csn:active"/>
 <xsl:template match="/stream:stream/xmpp:message/csn:composing"/>
 <xsl:template match="/stream:stream/xmpp:message/csn:paused"/>
 <xsl:template match="/stream:stream/xmpp:message/csn:inactive"/>
 <xsl:template match="/stream:stream/xmpp:message/csn:gone"/>
</xsl:stylesheet>
```

If the original stanza had children, but the resulting stanza no longer has any, the option *Allow Childless Message and Presence Stanzas Through* determines whether the stanza is let through. For example, if the stanza in example Example 7.3, “A stanza with only Chat State Notifications contents” is sent to a peer that has a Transformation rule that removes CSN (as shown in Example 7.2, “An XSLT that will strip Chat State Notifications from message stanzas” and the option *Allow Childless Message and Presence Stanzas Through* set to *Don't let stanzas through* or *Only allow presence stanzas through*, this stanza will not be sent to the recipient.

Example 7.3. A stanza with only Chat State Notifications contents

```
<stream:stream id='1' version='1.0' xmlns='jabber:server'
    xmlns:stream='http://etherx.jabber.org/streams'>
<message from='anne@example.com' to='bob@example.net' type='chat'>
  <composing xmlns='http://jabber.org/protocol/chatstates'/>
</message>
</stream:stream>
```

If the transformation returns an empty or invalid document, or a document that does not parse into a syntactically valid stanza, or the test described in the previous paragraph blocks the stanza, what happens is determined by the *Action on Blocked Stanza* option associated with the transformation. *Bounce* will send back an `error` stanza to the source of the stanza if appropriate for the stanza. *Drop* will cause the stanza to be discarded without notifications.

Chapter 8 M-Link Configuration and Management

This chapter introduces the M-Link Administration Interface, used for configuration and management of the M-Link server.

M-Link supports configuration and management through the M-Link Administration Interface, including management of product activation keys. M-Link has a built-in web server accessible to a browser by giving the server host as URL and specifying the "Administration Interface Port". By default, that is 5221, for example:

```
https://localhost:5221/
```

This example assumes that HTTPS is used for the Admin interface, which is the default unless the TLS feature is not available. Alternatively:

```
http://localhost:5221/
```

It is also possible to configure M-Link programmatically using the interface defined in Chapter 17, *HTTP API*.

M-Link cannot be used without an activation key. Upon first use, the M-Link Administration Interface takes the user through initial administrator user setup and product activation.

- Initial administrator user credentials are included within the M-Link Administration Interface configuration, to allow set up independently of any external directories. Their input is a one-off set up.
- Product activation key management is addressed later in this chapter (see Section 8.12, "Product Key Management").

If the operations listed above have been performed, the M-Link Administration Interface starts with the login page.

Once successfully logged in, the "home page" of the M-Link Administration Interface offers a choice between configuration, monitoring and administration activities. These are briefly described in following sections.

8.1 UI Navigation

After successful login, the top bar becomes visible and features the following:

- On the left, the application name which varies according to which M-Link features are activated.

  This is also a button for returning to this initial location from anywhere.
- A choice of main areas *(dependent on the activated M-Link product)*:
  - "Status"
  - "Configuration"
  - "Monitoring"
  - "Multi-User Chat"
  - "Archive"

- “Publish-Subscribe”
- “Form Discovery and Publishing”
- “Clustering”

- On the right, a drop-down menu with classic options providing:
  - Access to documentation.
  - Information about M-Link and the activation status.
  - Means of logging out or editing credentials.

A sidebar menu appears when entering each of the main areas which adjusts to the current location, providing a choice of sub sections and means of stepping back towards the top level. The “Back” browser action may also be used, but that always returns to the previous page, not necessarily to a page closer to the root.

Every navigation device (button, card or link which, upon mouse click, takes the M-Link Administration Interface to a different page) also supports “Open Link in New Tab” and “Open Link in New Window” normally provided by browsers in their context menus.

When working with multiple servers, it may be useful to identify tabs/windows connected to the same server: see option “Application Title Override” at the bottom of the “Global Options”, the starting point of the “Configuration” area.

8.2 Status

The Status area provides the status of various components configured on the server. If any component is not fully operational, then this will be indicated in the overall status displayed at the top of the page.

Warning: This page is a snapshot. Use the refresh button on the page to obtain an updated status.

8.3 Configuration

The Configuration area consists of a main or root page, displaying “Global Options”, and a number of sections or sub pages, accessible by clicking in a menu bar on the left hand side. These are organized in a tree-like structure, each page having a path from the root.

8.3.1 Common Features of Configuration

8.3.1.1 Viewing and Editing Settings

Each option is displayed with a name and followed by a description. Please refer to other chapters of this manual for the full significance of each option. Sometimes, long descriptions appear shortened. These can be viewed in full by clicking on the `More...` link.

Most options are defined by a simple value which can be edited in various ways as appropriate: mouse clicks, text input or drop-down menu option selection. Changes only take effect, that is, are conveyed to the server, when the Submit button, at the bottom of the page, is clicked. There is also a Cancel button which restores the page to the values currently set in the server.

Some options are only visible and in use when enabled by other settings.

- This serves to remove clutter, such as option “Customise Ports”, which needs to be selected before port number options appear.
- It also helps to manage dependencies: some option combinations would not always make sense. When options are thus controlled from another page, it may be necessary to submit the page with the enabling option before the dependent features appear. For example, the “Security Labelling Configuration” section only appears after submitting the “Global Options” page with “Use Security Labelling” ticked.

A few options are more complex and have a specialized editing mechanism. These are described later.

8.3.1.2 Default Values

A large number of options have defaults. These are denoted by a value displayed in italics, unless it is a boolean option (i.e. selected or not). They will also have a “Use default” tick box.

- This can be ticked to restore the option to its default value. If the tick is showing, then the option is definitely set to its default.
- Changing the value will automatically untick the box. It is, of course, possible to set values which happen to equal the defaults, in which case the server remembers this state explicitly, and the configured values will persist even if the defaults are changed.

Default values are often immutable, possibly defined by a standard. In some cases the default is itself an option which can be modified. Any option used as a default elsewhere is displayed with a message including paths to all the options currently defaulting to that option.

8.3.1.3 Validation

Option values are generally constrained by their types. For example, numerical values may be range limited, and domain names must adhere to standard specifications, etc.

- In most cases, validation is applied as you type and errors get highlighted.
- In a few cases, where consistency is required with other options, validation only happens upon submission to the server.

In any case, the M-Link server rejects invalid configuration updates. When that occurs, the page remains unsubmitted and an explanation pops up in a red, persistent box, which can be used to located the changes required to make a subsequent submission succeed.

An option may be mandatory if no default would be sensible. This is typical of list items that can be added, but not without filling in some essential properties. The mandatory options are marked by a label reading “Required”. Also, the Submit button will remain disabled until all mandatory options are filled.

Due to dependencies between settings, it is possible for a change to enable mandatory settings on other pages. This somewhat rare condition is detected upon submission and produces a pop-up, named “Missing fields”, listing options that will be made visible by the update and which are currently not set, yet mandatory. These are editable in the pop-up and must be configured to complete the submission. It is recommended to make a note of the option paths listed on the “Missing fields” pop-up and visit the containing pages after the current submission completes successfully, as other setting changes may be desirable.

Other unrelated errors may occur, such as caused by a connection failure. They generate red, persistent pop-ups containing an explanatory message.

All error pop-ups can be dismissed by a simple click.

8.3.2 Special Options

8.3.2.1 Options Referring to Objects

Configuring an M-Link server generally requires importing external data objects which cannot be predetermined. For example, TLS features require keys and certificates which do not ship with the server. This is a two step process:

- Upload of a number of objects into M-Link managed stores.
- Selection of stored objects where needed in the configuration.

For convenience, it is in fact possible to upload an object from the places where an object selection is needed. But logically, the two operations are disconnected, as the same object may be selected in several places of the configuration. An object may also be uploaded but not selected anywhere (i.e. not in use in the configuration) or cease to be selected.

Object stores group objects of the same kind and act like internal catalogs. Setting a configuration option to an object from a store amounts to making a selection from a list of object names. Such options appear with an “Edit” button which offers a drop-down menu. The options in the drop-down depend on what is currently uploaded in the associated store.

Since some of the uploaded objects are themselves catalogs of items (such as security label catalogs), the internal stores can have two levels. The editing pop-up will then show two drop-down menus. The option is then rendered as a combination of the two choices.

The M-Link Administration Interface has a section for object store management. See Section 8.3.3.9, “Stores”.

8.3.2.2 List Options

There are basically two kinds of lists:

- Collections of configurable items.
- Options that happen to be multi valued.

The first case corresponds to the M-Link Administration Interface sections (such as “Peer Controls”, see Section 8.3.3.4, “Peer Controls”) where the side bar menu leads to a page of cards representing the items in existence.

- Such pages may include an “Add item” button, unless displaying a fixed set of items.
- The cards may be clicked to access the features of the corresponding items.
- Items may also be reflected in the side bar menu, where a click will have the same effect.

There are variations with the second case of lists. Depending on complexity, the list items are either

- rendered in full or
- only partially but viewable one at a time in a pop-up.

In all cases, obvious controls are available to add, edit or delete items.

8.3.3 Configuration Sub Sections

8.3.3.1 Global Options

This page features all the options that do not belong in any of the sub sections. That is, options that apply server wide, excluding the following topics, which are substantial enough to require dedicated sections:

- TLS Configuration, with the exceptions mentioned below.
- Logging Configuration.
- Security Labelling Configuration.

The “Global Options” do include a couple of options that are TLS related:

Use HTTPS for Administration Interface
: This controls whether the administration interface (i.e. how the M-Link Administration Interface connects to the server) uses the HTTPS or HTTP protocols. Always use HTTPS, unless encryption is not available.

  If switching protocols is required, please note that the current connection will close, so the M-Link Administration Interface will appear to freeze and the browser must be told to visit the new location.

Use Secure Cookie Flag
: This controls whether use of the login cookie is restricted to secure connections. When enabled, browsers will be instructed to never use the login cookie over HTTP connections.

  The cookie is required for most interactions with the server and the default (which depends on the option above, `Use HTTPS for Administration Interface`) is generally the correct setting. However, this may need overriding in some circumstances, such as when connecting through a proxy that handles TLS for the administration interface.

Warning: Improper use of this option can block the M-Link Administration Interface updating the configuration.

TLS Key Pair for Administration Interface
: This option is obviously required when HTTPS for the Administration Interface is enabled.

If the TLS Key Pair is misconfigured somehow, M-Link will automatically revert to expecting connections through HTTP.

Please note that the “Security Labelling Configuration” section is only available when option “Use Security Labelling” featured in the “Global Options” is set.

8.3.3.2 MUC Domains

This page collects all the MUC domains currently configured, represented as cards. Please refer to Chapter 12, *MUC Room Creation and Administration* for an introduction to Multi-User Chat concepts.

- Domains can be added by clicking the “Add item” button.
- Domain settings can be viewed or edited by clicking on the corresponding card.
- Deleting a Domain requires viewing it and using the “Delete...” button on the viewing page.

8.3.3.3 Publish-Subscribe Domains

This page lists all the Publish-Subscribe domains currently configured, represented as cards. It functions very much like the “MUC Domains” page described above.

A PubSub domain serves as a namespace for Publish-Subscribe nodes. Its configuration includes global settings of the associated PubSub service. One option in particular, “Use domain for FDP”, decides whether it will be used for a *Form Discovery and Publishing* (FDP) service, a specialization of the generic Publish-Subscribe service (see Section 15.2, “FDP Administration”).

For general purpose Publish-Subscribe domain administration, please refer to Section 14.2, “Publish-Subscribe Node Administration”.

8.3.3.4 Peer Controls

This page collects all the Peer Controls currently configured, represented as cards. The card named “default” is special (it does not represent any real Peer), but is present here so it can be edited in the same way as proper Peer Controls.

- Peer Controls can be added by clicking the “Add item” button.
- Peer Control settings can be viewed or edited by clicking on the corresponding card.
- Deleting a Peer Control requires viewing it and using the “Delete...” button on the viewing page.

The “default” item will be used by M-Link when no Peer Control explicitly matches a domain. Please refer to Chapter 4, *Peer Controls and Links* for an explanation of how Peer Controls are used by the M-Link server.

8.3.3.5 Links

This page collects all the Links currently configured, represented as cards. Please refer to Chapter 4, *Peer Controls and Links* for an explanation of the concept.

- Links can be added by clicking the “Add item” button.
- Link settings can be viewed or edited by clicking on the corresponding card.
- Deleting a Link requires viewing it and using the “Delete...” button on the viewing page.

8.3.3.6 TLS Configuration

This page displays defaults and server wide options relating to TLS Configuration. See Chapter 5, *TLS Configuration* for details on this topic.

8.3.3.7 Logging Configuration

The side bar menu presents a “Logging Streams” entry, which leads to a page of cards.

- Logging Streams can be added by clicking the “Add item” button.
- Stream settings can be viewed or edited by clicking on the corresponding card.
- Deleting a Stream requires viewing it and using the “Delete...” button on the viewing page.

If the “Advanced Logging Level Configuration” is selected in a Logging Stream (and submitted), a “Facility Levels” entry appears in the side bar menu, which leads to a page of cards.

8.3.3.8 Security Labelling Configuration

The side bar menu entry for this page only exists if the global option “Use Security Labelling” is selected.

This page displays defaults and server wide options relating to Security Labelling. See Chapter 6, *Security Labels, Clearances and Policies* for details on this topic.

8.3.3.9 Stores

M-Link comes with a fixed set of object stores, each aggregating objects of a specific type and making them available as potential values for appropriate items in the configuration.

- The stores are represented here as cards.
- The contents of each store can be managed by clicking on the associated card.

Stores may be initially empty. They can be populated by clicking on the “Add item” button.

- In most cases, the operation involves choosing one, or a few, local files which contain the external embodiment of the object, for example, an XML file or a certificate in PEM format. The forms that pop up should be self-explanatory.
- In some cases, additional data needs to be input, such as a password.
- In all cases, a unique name must be given to the object. This defaults to the (main) file name without its extension, if any.

Store contents are presented as a list showing the names and upload dates of each object it contains.

- Each entry has a “Details” button leading to a read-only page showing significant details and all the properties of the object. The properties are specific to each type of object.
- This page also allows object deletion via the “Delete...” button, but only if the object is not in use, that is, has not been selected as the value of an item somewhere in the configuration. Otherwise, the header includes links to the places where it is used.

When objects are themselves a type of catalog, their sub structure is also presented in the form of a sub store. The item list of the store includes “Subitems” buttons, which lead to pages resembling the display of store contents. The differences derive logically from the fact that the sub items are parts of an immutable object:

- The list of sub items has no “Add item” button.
- The sub items cannot be deleted individually. Only the whole containing catalog object can be deleted.
- The sub items have neither upload dates nor source file names: these are attributes of the containing catalog object. They may have individual properties, however, since the containing catalog object may provide these.

8.3.3.10 Access Tokens

M-Link allows access to its configuration via application access tokens. These provide an alternative way to authenticate HTTP requests and are carried in `Authorization` headers (see section Authentication by Access Token).

The side bar includes option “Access Tokens” which leads to a page listing all application access tokens currently known to M-Link. The page includes token management facilities:

- The “Add token” button allows the creation of a new token and requires the input of a unique name.

  Please note: the token string for subsequent use in `Authorization` headers is only displayed once: a button is provided to copy the token string to the clipboard.
- “Delete” buttons can be used to permanently remove tokens from M-Link.

8.3.3.11 Node Specific Options

'Node Specific' options are a feature of clustering - see Section 11.3, “Node Specific Options” for further details

8.4 Monitoring

Monitoring activities have their own chapter in this manual. Refer to Chapter 16, *Monitoring* for further details.

8.5 Multi-User Chat Administration

MUC administration has its own interface, included in the M-Link Administration Interface under tab “Multi-User Chat"”.

> **Note:** Please note that MUC domain configuration and administration is only available for M-Link products that support *Multi-User Chat* (MUC). Refer to Section 1.4, “Installing an Isode Activation Key”.

See Chapter 12, *MUC Room Creation and Administration*.

8.6 Archive Administration

Archive administration has its own interface, included in the M-Link Administration Interface under tab “Archive”.

> **Note:** Please note that the “Archive” tab will only appear if archiving has been enabled in the “Configuration” area, setting option “Use Message Archiving”. You will need to refresh the browser after changing that option.

See Section 13.5, “M-Link Archive Server Administration”.

8.7 Publish-Subscribe Administration

General purpose Publish-Subscribe administration has its own interface, included in the M-Link Administration Interface under tab “Publish-Subscribe”.

> **Note:** Please note that Publish-Subscribe administration is only available for M-Link products that support *Instant Messaging* (IM). Refer to Section 1.4, “Installing an Isode Activation Key”.

See Section 14.2, “Publish-Subscribe Node Administration”.

8.8 Form Discovery and Publishing Administration

The FDP application of Publish-Subscribe has its own interface, included in the M-Link Administration Interface under tab “FDP”.

> **Note:** Please note that Form Discovery and Publishing administration is only available for M-Link products that support *Instant Messaging* (IM). Refer to Section 1.4, “Installing an Isode Activation Key”.

See Section 15.2, “FDP Administration”.

8.9 Clustering

Note: Please note that clustering is only available for M-Link products activated accordingly. Refer to Section 1.4, “Installing an Isode Activation Key”.

This area provides the facilities available in the M-Link Administration Interface for cluster management.

8.9.1 Enabling Clustering

This feature appears only if clustering has not been enabled, which would be the case for an initial M-Link configuration. Enabling clustering on this page will turn the current M-Link Server into the initial node of a new cluster.

The page displays a form with a few details to fill in. Submitting it will initialize clustering and the M-Link Server will become the initial node of the cluster. The form will then be replaced by the cluster status.

8.9.2 Cluster Status

When the M-Link Server belongs to a cluster, the “Clustering” page displays a card per node in the cluster, including one for itself. Disconnected nodes are highlighted as such.

8.9.3 Adding a Node to the Cluster

When the M-Link Server belongs to a cluster, the “Clustering” page includes a button to add a new node to the cluster.

Joining a cluster is an operation initiated at the joining node (see Section 1.6, “Creating or Joining a Cluster”), which must supply a *join request token*. The “Add node” button opens a form to submit the *join request token*. The *join response token* is then generated and displayed. This token should be provided to the joining node.

Warning: Joining only happens when that *join response token* is successfully submitted at the joining node (see Section 1.6, “Creating or Joining a Cluster”).

Note: Once the joining node has been added to a cluster, the set of services which make up the joining node should be restarted.

8.10 Removing a Node from the Cluster

Appart from the card representing the “home” node (that is, the M-Link Server this M-Link Administration Interface is connected to), cards in the “Clustering” page have a button to remove the node from the cluster.

Warning: The remove button acts immediately without warning.

This is typically useful when a node no longer exists. If however the removed node does exist, its own “Clustering” page, possibly after a refresh, will show all other nodes as disconnected. There is no way back into the cluster other than by re-installing a blank configuration and re-joining.

8.11 Miscellaneous Features

8.11.1 Server Version

The bottom of the side bar displays the current version of the server connected to the M-Link Administration Interface. Please include that version number in any support request.

The current version number can also be found with the “Product Activation” status in the “About” side bar (see below).

8.11.2 Documentation Side Bar

The “About Side Bar” is brought up by selecting the “Documentation” option in the drop-down menu of the top bar. It provides buttons for:

- Displaying the online manual.
- Displaying the release notes.

8.11.3 About Side Bar

The “About Side Bar” is brought up by selecting the “About” option in the drop-down menu of the top bar. It provides the following features:

- Display of product activation details and
- Management of the activation key (see Section 8.12, “Product Key Management”).

8.11.4 Signing Out or Editing Credentials

Normally the M-Link server keeps the M-Link Administration Interface logged in unless inactive for the duration of the “Administrators inactivity logout period” (configurable in the “Global Options”). When logged out at the server's initiative, the M-Link Administration Interface may not react immediately depending on the current activity. Any action requiring valid credentials will automatically redirect the current page to a login page.

It is possible however for the user to log out deliberately by choosing the “Sign Out” option in the top bar drop-down menu. This takes the M-Link Administration Interface straight back to a login page.

Credentials may be changed by clicking on the username in the drop-down menu. This opens the “Edit Credentials” form.

The main purpose of this form is to set a new password for the administrator currently logged in. It is also possible to edit the user name at the same time.

- Current (old) credentials are required input.
- Leaving field “New Username” blank preserves the current user name.

The changes take effect immediately (unless they are rejected). In other words, the user is not logged out, and opening a new connection to the server will require entering the new credentials.

8.12 Product Key Management

Product Key Management is accessible from the “About” side bar, which slides out when the “About” button is clicked in the drop-down menu. The side bar displays the details of the activation key currently loaded.

A new product key is required for the following purposes:

- To change the details of the activation.
- To extend the activation to a later release.
- To add or remove features.
- To widen or narrow the scope of the activation.

The “Update Key” button initiates the same operation required during initial activation.

- The pop-up may be closed at any time to continue working with the current key.
- The final step (the actual key update) overwrites the existing, but only if the new key is valid.

If an M-Link server is no longer needed, the product key can be removed using the “Deactivate” button. This operation cannot be undone, unless the key was saved elsewhere. As this returns the M-Link product to a non-activated state, product activation is then the only operation available.

Chapter 9 Instant Messaging Domain and User Configuration

This chapter describes how Instant Messaging domains and users are configured within M-Link.

9.1 Instant Messaging Domain Configuration

A single M-Link Server may host multiple Instant Messaging domains. Within the IM Domains section of the M-Link Administration Interface, the following items may be configured:

Domain
: The name of the Instant Messaging domain. This must be a valid JID domain component.

Authentication Type
: Specifies whether users within the domain are configured statically within M-Link Server or are held within an external LDAP Directory.

> **Note:** The use of Static Users is only appropriate for a demonstration system. Production systems should use an external LDAP Directory to hold user configuration.

LDAP Configuration
: If the Instant Messaging domain is configured to use LDAP for authentication, this item specifies the LDAP configuration which is to be used. See LDAP Configurations for more information.

TLS Required
: When this option is enabled, and if a TLS identity is configured, only TLS-encrypted connections from clients will be accepted

TLS Identity
: The key and certificate this Instant Messaging domain presents to inbound connections. If this is configured but the "TLS Required" item is set false, TLS will be offered but not required.

Enable HTTP Upload
: Support for HTTP File Upload (*XEP-0363*) can be enabled or disabled per domain. See Chapter 10, *HTTP File Upload* for more information.

An Instant Messaging domain may also have associated Security Labelling configuration: this is described in Section 6.5, “IM Domain Security Configuration”

9.1.1 Instant Messaging Domain Static User Configuration

If an Instant Messaging Domain is configured with an Authentication Type of "Static Users", a list of Static Users can be configured subordinate to the Domain itself. Each Static User may be configured with the following items:

JID
: The JID of the user. This must have the same domain component as the Instant Messaging Domain.

Password
: The password for the user.

A User may also be assigned a Security Clearance. This is described in Section 6.10.1, “Client Security Clearance Configuration”.

9.1.2 Static Groups

One or more Static Groups can be configured within the Domain. Each Static Group contains a list of JIDs, which should all be members of the Instant Messaging Domain.

Note: Group names including right-to-left characters are not supported.

9.1.3 Roster Groups

An Instant Messaging Domain may be configured with one or more Roster Groups. A Roster Group provides a list of JIDs to all users who themselves are members of the Roster Group. Each Roster Group may be configured with:

Name
: A name assigned to the Roster Group. This name will be exposed to Roster Group members.

Group Type
: This configures the type of the Roster Group; either "Static", referencing an existing Static Group configured for the Instant Messaging Domain, or "LDAP".

Static Group
: If a Group Type of "Static" is selected, this option specifies the Static Group to be used.

LDAP Group DN
: If a Group Type of "LDAP" is selected, this option specifies the Distinguished Name of the LDAP entry which holds the Roster Group.

9.2 LDAP Configurations

An LDAP Configuration defines how M-Link will use an LDAP directory for User and Group lookup. Multiple LDAP Configurations may be configured; the same Configuration may be used by multiple Instant Messaging Domains. An LDAP Configuration may have the following settings:

Configuration Name
: A user-friendly name assigned to the LDAP Configuration.

LDAP URI
: This encodes the address and port of the LDAP Directory server, for example "ldap://ldap.example.com:19389".

Use TLS
: Configures how TLS is used when connecting to the LDAP server.

LDAP Authentication Identity
: The identity used for authentication to the LDAP directory. This may be either a SASL identity or a string-encoded DN as appropriate.

LDAP Authorization Identity
: The identity used for authorization by the LDAP directory. This defaults to the LDAP Authentication Identity value.

LDAP Password
: The password associated with the LDAP Authentication Identity.

Perform authentication by binding to the DSA
: Select how user password validation takes place. If this option is selected, the server will bind to the LDAP server as the user; if using an Active Directory LDAP server

you should always select this option. If the option is not selected, the server will read userPassword attributes from the LDAP server and perform validation locally; the SCRAM-SHA-1 and PLAIN SASL mechanisms will be offered. Note that for this second option to work, the LDAP Authorization Identity must have read permission for userPassword attributes.

LDAP User Base
: This configures the base for User searches. It is configured as a string-encoded Distinguished Name.

LDAP User Attribute
: This configures the name of the LDAP attribute which holds a User's JID. If unset, the default used is "mail".

LDAP User Search Scope
: This configures the scope of the search operation used to locate users beneath the LDAP User Base entry.

LDAP User Filter
: This enables the configuration of an additional LDAP filter term to be used when searching for users beneath the LDAP User Base entry.

LDAP Group Base
: This configures the Distinguished Name of the entry in the Directory beneath which LDAP Groups are located. This must be set for LDAP Group lookup to occur. An LDAP Group is any entry which contains "member", "uniqueMember" or "roleOccupant" attribute values.

LDAP Group Search Scope
: This configures the scope of the search operation used to locate group entries beneath the LDAP Group Base entry.

LDAP Group Filter
: This enables the configuration of an additional LDAP filter term to be used when searching for groups beneath the LDAP Group Base entry.

Enable Advanced Configuration
: This checkbox allows configuration of advanced options.

Excluded Attributes
: This is a list of attribute types which should be excluded from the set which is requested from the LDAP server for a user. An example is `jpegPhoto`, which is used when constructing an initial VCard for a user.

LDAP UID Rules
: This allows the configuration of multiple regex/replacement pairs which operate on user account name values. For example, this could be used to normalize a "mail" attribute value of "alice@development.example.com" into a user account name of "alice@example.com".

Note: This item is located on the sidebar menu rather than on the page for the LDAP configuration itself.

9.3 User Provisioning Using Cobalt

Refer to the Cobalt User Manual for information on how to provision users.

Chapter 10 HTTP File Upload

This chapter describes the HTTP File Upload feature.

10.1 HTTP File Upload

M-Link MU Gateway and M-Link MU Server support HTTP File Upload (*XEP-0363*), which provides a standardised way for users to share files using an *HTTP* service that is advertised by XMPP servers. Using this protocol, clients supporting the protocol can request a location to upload a file to, and share a (public) link to the uploaded file.

The HTTP File Upload service shares configuration settings with the M-Link Server running on the host system, and will automatically detect and apply changes relevant to its operation.

10.2 Configuration

By default, HTTP File Upload is disabled. It can be enabled through the M-Link Administration Interface on the main HTTP File Upload configuration page. A hostname must be provided, this is the *DNS* name that will be used for both uploading and downloading shared files. If *TLS* is enabled, a `TLS identity` must be selected as well. This identity should cover the configured `Default HTTP Hostname` to ensure clients will be able to use the service. Once enabled, this page also allows configuration of file size limits.

Once support for HTTP File Upload has been enabled, every configured IM Domain will support the feature, although this can be disabled per domain on the corresponding IM Domain section. Uploading files is by default limited to local users of the domain only, but can be exposed to users of any local or remote domain by disabling the `Limit to local users` option.

Chapter 11 Clustering

This chapter discusses using several M-Link Server instances to provide a single M-Link service.

11.1 Background

An M-Link service that consists of two M-Link Server instances co-operating to provide a single XMPP service is called a *cluster*. Each of the M-Link Server instances is a *cluster node* or *node*. If cluster configurations are not relevant to your deployment, you can skip this chapter.

Some of the reasons for using a cluster, rather than a standalone M-Link service, include:

- To provide redundancy: if one system fails, then the XMPP service will continue to be provided by other systems that are part of the same cluster.
- To spread load: as workload increases, it may be more effective to add extra servers that act as cluster members, rather than upgrade the hardware of a single system.
- To reduce network latency: if clients of the service are distributed over a wide geographical area, then having all of them depending on a server in a single location may be less efficient than providing servers at locations that are close to the users.

Reasons that you might *not* want to use a clustered configuration would include:

- Cost: for every member of the cluster that is added, extra hardware and product licenses are required.
- Management complexity: while clustering is relatively straightforward, it does require extra configuration.
- Clustering overhead: nodes of an M-Link cluster communicate with one another to share state, and so enabling clustering entails some level of extra load on the network and systems.

11.2 How nodes are added to a cluster

A single-node M-Link service becomes a cluster when you create a new M-Link Server that is part of the existing service. The two nodes co-operate to ensure that the service configuration is consistent between them (see Section 1.1, “About M-Link Server”).

In a two-node cluster the initial node retains no special status; both members of a cluster are equal peers.

Clusters can be configured using the M-Link Administration Interface, see Section 8.9.1, “Enabling Clustering”.

Note: Once a new node has been added to a cluster, the set of services which make up the new node should be restarted.

11.3 Node Specific Options

Most configuration options are synchronized across the nodes in the cluster; Node specific options refer to settings that are configured independently such that the nodes in the cluster may have different values (e.g. for IP listen addresses).

Chapter 12 MUC Room Creation and Administration

This chapter introduces the *MUC* room concept and covers how to create and administer *MUC* rooms.

If the activated M-Link product supports MUC, the M-Link Administration Interface will feature a "Multi-User Chat" tab. Under that, an administrator may create new *MUC* rooms and manage all existing *MUC* rooms.

12.1 MUC Room Concepts

Note: The Multi-User Chat protocol is defined in XEP-0045.

A *Multi-User Chat* (MUC) room is an addressable chat for multiple users (those users currently “in” the room are known as “occupants”). Rooms belong to a domain hosted by an XMPP server and have a unique address combining a room name with the domain name: <room-name>@<domain-name>.

Once a hosted *MUC* domain has been created (see Section 8.3.3.2, “MUC Domains”), XMPP users may “join” rooms in that domain from their XMPP clients, possibly creating the rooms on the fly if they didn't exist depending on configuration. Rooms can be persistent or transient, with transient rooms being destroyed when the last “occupant” leaves them. Rooms created through the M-Link Administration Interface are persistent.

Rooms have “affiliation” lists, defining a relationship between the listed users and the room, used by the room to determine their initial roles and privileges upon joining. Please refer to XEP-0045 for an exact definition of roles and privileges. In decreasing privilege order, the affiliation types are:

Owner
: Room owners have full control over the room configuration and existence. A user who creates a room in their XMPP client is automatically an owner of the room.

Administrator
: Users who are administrators can moderate other occupants, but not configure the room.

Member
: For “members only” rooms, a user may only enter if they have a member (or higher) affiliation.

Outcast
: Users affiliated as outcasts are banned from the room. This means they can neither see nor send any message exchanged in the room, and they get expelled from the room should they become outcasts while in it.

Affiliations are determined by two levels of configuration, domain and room levels. The lists refer to JIDs and can contain bare user JIDs or domain names (domain affiliations are equivalent to an affiliation for every user on that domain).

Domain-Wide
: A MUC domain administrator can configure affiliations that automatically apply to all rooms in the domain. For example, to ban a specific user from all rooms in the domain the user's JID can be added to the outcast affiliation list of the MUC domain.

Per Room
Room specific affiliations will apply unless overridden by domain-wide affiliations in the following cases:

- A domain-wide affiliation exists, either for the same JID or for a domain or group it belongs to, and confers higher privileges.
- An outcast affiliation at the domain-wide applies to the user JID, also explicitly or by inclusion.

12.2 Room Administration

Note: The facilities for creation, configuration and removal of *MUC* domains can be found in the Configuration area under Section 8.3.3.2, “MUC Domains”.

An administrator may use the M-Link Administration Interface to perform the following operations:

- View the configuration of all existing rooms in the MUC domains hosted by the server.
- Create rooms.
- Change default settings for new rooms.
- Change current settings of any room (room settings are not affected by changes in the defaults after their creation).
- Manage the list of affiliations associated with a room.
- Manage domain-wide affiliations that apply to all rooms within it, potentially overriding their own room specific affiliations.
- Monitor the status of rooms (e.g. number of occupants, *FMUC* status).
- Destroy rooms.
- Make temporary rooms persistent.

12.2.1 Available MUC Domains

The landing page under tab "Multi-User Chat" displays the MUC domains hosted by this M-Link server. The creation and removal of domains is a matter for server configuration. Refer to Section 8.3.3.2, “MUC Domains”.

Each domain is represented by a card showing the domain name and the number of rooms it currently has. Clicking on a card leads to the Domain Management page for that domain. See the next section.

12.2.2 Domain Management

Listing Rooms in a Domain

A MUC domain management page displays the name of the domain and represents all its rooms as cards. A click on a card opens the associated room management pop-up. See Section 12.2.4, “Change Room Settings”.

Note: This is a dynamic page and rooms can be transient. Please note the auto-refresh button and setting.

The page also provides access to domain level settings with two buttons beside the domain name:

Manage Domain...
: This button opens a pop-up where domain-wide settings can be changed. Tabs in its header allow switching between the features:

  Room Defaults
  : For details on domain-wide room defaults, refer to Section 12.2.3, “Room Creation”.

  Domain Affiliations
  : For details, refer to Section 12.2.5, “Affiliation Management”.

  Changes to the defaults or affiliations take affect upon clicking the “Update” button in the footer of the pop-up.

Create New Room...
: This button opens the room creation pop-up.

  For details on room creation, refer to Section 12.2.3, “Room Creation”.

  Clicking the “Add” button creates the room and dismisses the pop-up.

Rooms can also be destroyed from the room configuration pop-up, Section 12.2.6, “Room Destruction”.

12.2.3 Room Creation

12.2.3.1 Domain-Wide Room Defaults

Room defaults are settings that will be given to new rooms upon their creation. They are defaults in the sense of an initial room configuration: Subsequent erasure of a room setting does not amount to specifying a value from the defaults and subsequent changes to the defaults do not affect existing rooms.

12.2.3.2 Room Creation by a MUC Administrator

The “Create New Room...” on a Domain Management page opens the room creation pop-up.

The first field is the mandatory name for the room, which must be unique within the domain. It must be valid as the node part of a JID.

Other fields will have been filled according to the room defaults described above. They may be modified before clicking the “Add” button.

12.2.3.3 Room Creation by an XMPP User

Users may attempt to join rooms that do not exist. A successful join (the facility depends on the XMPP client being used) means the room was created on the fly using the room defaults set up by a MUC administrator for the domain. It will be owned by the creating user.

XMPP clients may also provide means for users with owner affiliation (and administrator affiliation to a lesser extent) to change room settings as well.

12.2.4 Change Room Settings

From a domain management page, clicking on the card for a room opens a pop-up offering a number of tabs:

Room Configuration
: This tab displays all the configurable settings of the room. Changes take effect when clicking the “Update” button.

Room Affiliations
: This form works in exactly the same way as for domain-wide affiliations. Please refer to Section 12.2.5, “Affiliation Management”.

FMUC Status
: This tab is for monitoring purposes and only appears if FMUC is enabled in the "Global Options". Please refer to Section 12.2.8, "Room Status".

12.2.5 Affiliation Management

Domain-wide and room-specific affiliations are maintained in the same way, so this section covers both. As summarized in Section 12.1, "MUC Room Concepts", domain-wide affiliations may take precedence over room-specific affiliations.

- All affiliation types are displayed by default (if any), but the list can be filtered according to the check-boxes in the "Filter by" box.
- The "Add Affiliation" button can be used to add new entries. This opens a pop-up giving the choice to set a user JID, domain or group affiliation.

  Setting group affiliations requires some groups to have been defined. This can be done in the "Configuration" area under the "IM Domains" page, selecting (or adding) a domain (see Chapter 9, *Instant Messaging Domain and User Configuration*). There are then two options:

  Static groups
  : Each IM domain can define static groups. Click on the "Static Groups" option in the sidebar menu and use the "Add..." button to create a new static group. Then use the "Members" option in the sidebar menu to add members.

  Groups defined in a remote *DSA*
  : If the selected domain uses LDAP based authentication (see option "Authentication Type"), then groups defined in the remote *DSA* can be used. This *DSA* will have its own administration interface to manage the groups.

  Note: The "IM Domains" page is only available if the activated M-Link product supports IM.

- Entries have a "Remove" button and an affiliation selector that allows subsequent affiliation changes without having to remove and re-insert the entry.
- Entries are sorted by JID and the sort order can be changed by clicking on the column header.

12.2.6 Room Destruction

The footer of the Room Configuration pop-up includes a "Destroy Room..." button.

Note: Possible path to get there:

- Click on the "Main Menu" link in the left side-bar (while in the "Multi-User Chat" area).
- Click on the domain card.
- Click on the room card.

Destruction is effective when confirming with a click on the "Yes I'm sure" button.

12.2.7 Persisting Temporary Rooms

The M-Link Administration Interface only creates persistent rooms. However, transient rooms may be created by users as they join a non-existing room. A transient room is automatically destroyed when the last "occupant" leaves.

The MUC domain management page will display cards for transient rooms while they exist and these may be configured just like persistent rooms. An administrator may

make a transient room persistent by ticking the “Persistent Room” box in the Room Configuration tab and updating it.

> **Note:** Persistent rooms cannot be made transient: the “Persistent Room” box will be visible and ticked, but disabled. Immediate room removal is available however: refer to Section 12.2.6, “Room Destruction”.

12.2.8 Room Status

The cards display some status information about the rooms:

- Number of occupants.
- Number of joined *FMUC* rooms, or “N/A” if the room is not federated.
- If the room is federated, any remote nodes failing to join.

Clicking on the card for a room opens a pop-up which includes an “FMUC Status” tab. If applicable, this tab provides federation details which cannot be shown on the card.

12.3 MUC Federation

> **Note:** The Federated Multi-User Chat protocol is defined in XEP-0289.

Although XMPP provides a service federated over multiple servers, standard MUC rooms operate on a single server. This centralized MUC model works well in many situations, but is not ideal in all environments. In particular:

- Where there is a constrained link between a pair of servers, operating a MUC room on one server with clients on both servers leads to inefficient operation. It also prevents operation of remote clients when the link is not working.
- In a cross-domain environment, it is undesirable to have clients directly accessing a MUC room in the remote domain

Federated MUC (FMUC) addresses these issues. FMUC enables MUC rooms on multiple servers to federate and provide a single MUC room. This provides optimized performance for constrained links. More information on FMUC is provided in the Isode whitepaper "Federated Multi-User Chat: Efficient and Resilient Operation over Slow and Unreliable Networks", available at https://www.isode.com/whitepapers/federated-muc.html.

FMUC is enabled server-wide (see Section 12.3.1, “Enabling FMUC for the server”); individual MUC rooms can be configured to participate in a federated FMUC room, simply by including the names of the federated rooms to which messages are sent. FMUC is simply configured using a set of standard MUC rooms that will work together.

FMUC configuration is per-room, but first requires FMUC to be enabled for the server.

12.3.1 Enabling FMUC for the server

A control in the M-Link Administration Interface “Global Options” allows FMUC to be enabled for the M-Link server. Doing this makes a number of additional controls visible:

FMUC Connection Retry Timer
: The interval (in seconds) at which the M-Link server will attempt to establish federation between MUC rooms which are configured to be federated but have

become detached. Configuring a value of zero will disable background reconnection attempts.

FMUC Ping Timer
: The interval (in seconds) at which XMPP "ping" stanzas will be sent to each MUC room's federated nodes.

FMUC Stanza Timeout
: The period (in seconds) after which an FMUC stanza (including a "ping" stanza) will time out. Stanza timeout results in federated room detachment.

FMUC Join Timeout
: The period (in seconds) after which an FMUC join attempt times out.

12.3.2 Configuring FMUC for a room

Once FMUC has been enabled for the server, a set of FMUC options are made available on each MUC room configuration form. These are:

- "**Allow FMUC from arbitrary sources**" will, if enabled, accept requests to start federating from any other MUC, irrespective of whether it is in the configured node list or not.
- "**Federated MUC nodes**" is a list of the MUCs that this MUC is configured to federate with - note that while it is possible to federate with many nodes, there must be no 'loops' in the config (that is: it must be an acyclic graph). So to have the rooms *demo@conference.example.com* and *sample@rooms.server.local* federate you would add *sample@rooms.server.local* to the **Federated MUC nodes** list of *demo@conference.example.com*, and add *demo@conference.example.com* to the **Federated MUC nodes** list of *sample@rooms.server.local*.
- "**Add remote FMUC domain name to nicks**" controls whether the remote FMUC domain name should be appended to the nickname for those MUC occupants who are present via federation.
- "**History requested from FMUC**" determines how many historic messages should be requested when MUC federation occurs.

12.3.3 Things to note

- When a federated MUC becomes unavailable it may take some time before the link is considered 'dead' and remote occupants in the MUC are marked as having left the room. It is possible for occupants of different nodes of an FMUC room to see messages arrive in a slightly different order when either messages are sent at the same time or the network link between servers is slow.
- You can only federate between local and remote MUC rooms, you cannot federate between two MUC rooms hosted on the same M-Link instance.
- If FMUC is enabled for the server, anyone able to create a MUC room can then configure it to federate with a remote host - this provides a mechanism for potential abuse if your users are not trusted.

Chapter 13 Archiving

This chapter describes the M-Link archiving capabilities.

13.1 M-Link Archive Server

The M-Link Archive Server is used to archive the XMPP traffic on behalf of M-Link Server. Communication between the M-Link Archive Server and the M-Link Server uses a high performance protocol.

Most stanzas that pass through the M-Link Server using XMPP, covering all types of traffic, are sent to the configured M-Link Archive Server. Key state changes are also sent.

The archiving mechanism is asynchronous, and so normal processing of XMPP traffic is not affected by database writes (or reads). The M-Link Server writes queued archive data to disk until they have been transferred to the M-Link Archive Server and acknowledged. This makes archiving resilient to outages and restarts of both M-Link Archive Server and M-Link Server processes.

The M-Link Archive Server normally shares configuration settings with the M-Link Server running on the host system. When M-Link Server configuration is modified, the M-Link Archive Server reloads its configuration.

For information about command-line administration of the archive, see Appendix A, *Archive Admin Tool*

13.2 User Access to Archives

End user access to the archives is provided using the XMPP extension protocol called Message Archive Management (MAM). XMPP clients that implement MAM issue queries to the M-Link Server which in turn queries the M-Link Archive Server. Standard XMPP authentication and access control applies, so users can only see traffic that they were party to or could have been party to.

Isode provides the Message Archive Browser, an XMPP over BOSH client that uses MAM to provide user access to the archives.

13.3 Archive Data and Configuration

The M-Link Administration Interface enables configuration of archiving for IM, MUC and Pubsub domains within a server. The following configuration options are available:

Use Message Archiving
: This option turns on message archiving and enables more detailed configuration.

Archive Scope
: This option configures a global setting for archive scope. The options are "Messages and Events", "Events Only" or "No Archiving". The same option is available for individual IM, MUC and Pubsub Domains; these default to the global option setting,

but can be configured individually to allow or permit archiving for individual domains.

Allow Message Archive Management
: This option enables the Message Archive Management (*XEP-0313*) feature at a global level. The same option is available for individual IM, MUC and Pubsub Domains; these default to the global option setting, but can be configured individually to allow or permit MAM for individual domains.

Archive Queue Path
: This is the location in which messages which have not yet been acknowledged as archived are stored.

Note: This option is hidden unless the global "Customise Storage Paths" option is selected.

Note: Changing this path will not automatically move any existing unarchived items.

Queue Timeout
: The minimum amount of time (in milliseconds) the server waits for acknowledgement from the archiving service until it persists messages to an on-disk queue.

Maximum Page Size
: This option configures the maximum number of items returned per MAM query.

Default Page Size
: This option configures the default number of items returned per MAM query.

Archiving Server Listening Address
: This option configures the IP address on which the archiving server serves its administrative API.

Archiving API Port
: This option configures the port on which the archiving server serves its administrative API.

Note: This option is hidden unless the global "Customise Ports" option is selected.

Use HTTPS for the API
: This option enables the use of HTTPS for the administrative API.

Archiving API TLS Key Pair
: This option configures the key and certificate pair used to serve the administrative API.

13.4 Message Archive Browser Configuration

The configuration file for Message Archive Browser is located at *$(VARDIR)/webapps/config.js*. A sample config file is provided at *$(SHAREDIR)/webapps/config_sample.js*, copying this file and renaming should be sufficient for most deployments.

The configuration is stored in a JavaScript file in an Object variable named `cfg`; it is important that you do not change the name of this variable.

boshURL
: This will be used to prefill the BOSH URL on the login form.

modules
: This is a JavaScript Object using the module name as the key and a Boolean value to indicate whether the module should be shown or not. There is currently one module available: `mam`.

13.5 M-Link Archive Server Administration

The M-Link Archive Server provides an administrative interface that can be used to manage the Archiving Service.

If option “Use Message Archiving” is set in the “Archiving Configuration”, the M-Link Administration Interface will offer an **Archive** tab from which the Archiving Service can be managed.

Note: You will need to refresh the browser after changing that option.

The **Archive** tab provides access to the functions described in the following sections.

Warning: As mentioned in Chapter 1, *M-Link: Getting Started*, until the initial, auto-generated self-signed certificate is replaced, browsers will raise a security warning and require an override acknowledgement. This creates an exception, recorded by the browser, telling it to trust that certificate, despite it being self-signed, but only for the purpose of connecting with the M-Link Administration Interface. When using the Archiving Service, another server is involved (using port `5080` by default) and because this access does not result directly from user interaction, browsers are not likely to prompt the user for another manual override and will simply block the transaction. The user experience is then a terse network failure whilst trying to use M-Link Archiving administration functions.

Tip: The solution is to manually take the browser to the port served by the M-Link Archive Server and override the warning about the certificate. In practice, that means changing the port number in the browser address bar from `5221` to `5080`, assuming the defaults are still in place. Please note that the path following the port will not mean anything to the M-Link Archive Server and will result in a blank page. After accepting the certificate exception, revert back to the previous path using, for example, the browser back facility. It is recommended, however, to configure the M-Link Server with a properly signed certificate as soon as possible anyway.

13.5.1 Searching Data

The M-Link Administration Interface provides an interface to perform search queries on the archives. This enables a server administrator to search for MUC and user chat history, with the option to perform a zoom operation on a message and to redact selected messages.

To perform a search, select the **Search** option from the sidebar menu under the **Archive** tab. This page displays a dialog, allowing you to specify the search parameters. By selecting option “`1:1 chats between`” from the drop-down menu in the search box, search results will include only messages that have been sent to or from a specific person (JID). A search of 1:1 chats will match all messages that are sent to or from the specified JID. If you want to limit these results, you can specify a second JID in the Search box, in this case, only messages exchanged between those two addresses will be returned.

Select the `Chatroom` option from the drop-down menu to search for messages that appeared in a specific chatroom (MUC), then enter a specific chatroom name.

In addition, it is also possible to specify a date-time range, limiting the results to messages sent between the specified times. These are optional and one or both date-time parameters may be omitted from a search query, in which case the maximum available time range is returned.

The final parameter is an optional text filter, allowing you to specify a string of text that must be present in the results. Any results that do not contain the specified string will not be displayed.

Once you have entered the search parameters, click the **Search** button to perform the search. The results will be displayed in the main window, with the option to redact selected messages or zoom in on a specific message.

13.5.1.1 Message Redaction

Redaction is a process whereby content is adapted into a form suitable for publication. Within the context of the M-Link Administration Interface, redaction of messages refers to the removal of selected message content, preventing the message content from appearing in subsequent search results. This is useful in cases where it may not be possible or desirable to disclose the content of specific search results in search history or exported data. By redacting a message, the content of the message is removed and the message is marked as redacted. Any subsequent searches returning redacted messages will only display the details of the message, excluding the message body. When displaying a redacted message in the M-Link Administration Interface, text will be displayed in the body of the message indicating that the message has been redacted, specifying the time at which the redaction occurred.

Note: It is not possible to undo a redact operation. Applying the redaction will remove the content of the selected messages. While the contents of a redacted message are permanently removed from the archive, the message header containing the time stamp and sender/recipient address is retained. If a time-based search returns redacted messages, they will be shown in the results view, with a message indicating that they have been redacted. It is also worth noting that some IM clients maintain local chat history, and any such history will be unaffected by a redaction operation.

13.5.1.2 Zoom Function

The zoom function in the M-Link Administration Interface is provided to display a specific message from an Archive search query in context, with the immediately chronologically preceding and proceeding messages. The zoom button appears as the mouse hovers over an entry in the search results, on the right hand side. It is represented by a magnifying glass and applies to the message in the same row.

Note: This option is relevant when search results have been filtered according to a specified search term using the `Match these words` field in the search parameters, or when a search query has been made within a specified time period, and a result is selected.

Where a search term has been provided in a search query, only results containing that term will be displayed. This means that the chronologically preceding and proceeding messages are not necessarily visible, depending on whether they contain the search term or not. By clicking the Zoom button on a message from the search results, chronologically preceding and proceeding results will be displayed regardless of whether they contain the search term.

Where a time period has been specified, results will only be displayed if the time-stamp for each message falls within this period. This may exclude immediately surrounding

messages, depending on the time-stamp of each message. In this context, the zoom button will display chronologically preceding and proceeding messages regardless of their time-stamp.

13.5.2 Expiring Data

Expiring data is a way to remove data from the archive database. This can be useful in situations where data can only be retained for a specified time period. M-Link Administration Interface provides means to expire data before a specified date and time.

When performing an expire operation, an expire request is sent to the M-Link Archive Server, which removes all of the data before a specified date and time. Depending on the size of the database, this may take some time to complete, and the operation can not be undone.

Warning: Note that once you have expired any data from the Archive database, you must not subsequently import data to that database if that data contains information which precedes the expiry date.

In order to expire data using the M-Link Administration Interface, select the **Expire** option from the **Archive** sidebar menu. It is possible to expire data on a single domain, or for all domains. The **Expire Archive** page allows you to specify the date-time before which all messages will be expired: either by selecting a specific date, or by specifying a time period relative to the current date-time. A confirmation dialog will be shown before the expiration is carried out.

Note: Expired data cannot be recovered from the archive, so it may be appropriate to back up the archive before performing an expire operation.

13.5.3 Archive Files

Selecting the **Files** option from the **Archive** sidebar menu takes the browser to the **Files** page. This page lists XML files held on the Archive server in a staging area as opposed to data stored in the archive database. From this page, files can be...

exported;
: an operation which extracts data from the archive database and stores it as an XML file in the staging area.

downloaded;
: an operation which transfers (copies) a file from the staging area to the local machine.

uploaded;
: an operation which transfers (copies) a file from the local machine to the staging area.

imported;
: an operation which loads data from an XML file into the archive database.

deleted;
: an operation which removes a file from the staging area.

Note: All operations are asynchronous and the **Files** page is not necessarily updated upon their completion; so it may be necessary to refresh the page after a while to see the effect of an operation.

The files are listed in a table, associating each file with a number of status flags and buttons or links. The flags have a value of `Yes` or `No` and indicate whether the file is valid and which operations have been successfully performed on it: uploaded, downloaded, imported or exported.

13.5.3.1 Exporting Data

Data stored in M-Link Archive Server may be exported to an XML file. It is possible to export data for a specific domain, or for all domains, and to specify what time period the export should include.

The export functionality of the M-Link Administration Interface can be used to export data from the Archive database to an XML file. XML export is useful in situations where a backup of archive data is required. This is because it can be imported into an Archive server in the event of a loss of data, loss of hardware or corruption.

When performing an export, the M-Link Administration Interface sends a request to the server for the archive data. This is then stored on the server as an XML file, showing in the table on the **Files** page, from which it can be downloaded using the “Download” link.

The **Files** page of **Archive** tab has an **Export...** button for exporting data from the Archive database. Clicking this button will present a dialog where you will be able to specify a name for the file, the domain to export (if applicable), and the time period to export. It is possible to exclude one of the time fields, in which case data before/after the specified time will be exported. If no times are specified, the complete archive (for the selected domains) will be exported. Depending on the size of the database, this may take some time. Submit the dialog by clicking the **Export** button in its footer.

After a while, refresh the page until the file appears in the table. It should have a `Yes` in the **Exported** column and a `No` in the **Downloaded** column. You may then use its “Download” link to download the file to your local machine.

The XML format of the export is given in Appendix G, *Message Archive Format*.

13.5.3.2 Importing Data

The M-Link Archive Server supports importing data from XML files.

When performing an import operation, the M-Link Archive Server will attempt to validate the selected file. If the file is a valid archive XML file, the data it encodes will be integrated into the archive database.

Warning: Note that once you have expired any data from the Archive database, you must not subsequently import data to that database if that data contains information which precedes the expiry date. For more information on data expiry, see Section 13.5.2, “Expiring Data”.

The XML format of the import is given in Appendix G, *Message Archive Format*.

Archive data can be imported using the **Files** option in the sidebar menu of the **Archive** administration tab. The **Upload...** button allows you to add files to the list. An uploaded file will be initially marked as not valid, because the validation only occurs while attempting to import it. Use a file's **Import** button to import it into the archive. After a successful import (and a browser refresh), the file will be listed in the table with the `Imported` and `Valid` flags set to `Yes`.

Chapter 14 Publish-Subscribe Administration

This chapter introduces the *Publish-Subscribe* (PubSub) concept and covers how to create and administer PubSub nodes.

If the M-Link server is activated as either an M-Link User Server or an M-Link MU Server, the M-Link Administration Interface will feature a “Publish-Subscribe” tab. Under that, an administrator may create new Publish-Subscribe nodes and manage them to some extent.

14.1 Publish-Subscribe Concepts

M-Link User Server and M-Link MU Server products support PubSub nodes defined under dedicated domains (see Section 8.3.3.3, “Publish-Subscribe Domains”). Nodes are targets to which publishers send data and from which subscribers receive event notifications. Depending on configuration, events may deliver the data or merely identifiers for data items which subscribers then need to fetch.

Access to nodes is controlled by affiliations. An affiliation associates a user JID with a privilege level when dealing with that node.

owner
: An owner has full control over the node and may, in particular, configure the node properties, delete items published to the node, or delete the node itself (which removes all items).

publisher
: Publishers can neither configure nor delete the node, but may delete any published item.

publish-only
: Users affiliated with the publish-only privilege can only delete items they published themselves.

member
: Members can subscribe to the node and then receive publish event notifications, but cannot publish items.

outcast
: Outcasts are users who are explicitly banned from accessing the node.

For full details on *Publish-Subscribe* (PubSub) refer to XMPP extension Publish-Subscribe (*XEP-0060*).

The Form Discovery and Publishing service (FDP) is implemented on top of Publish-Subscribe with additional rules. See Chapter 15, *Form Discovery and Publishing Administration*.

14.2 Publish-Subscribe Node Administration

Note: The facilities for creation, configuration and removal of Publish-Subscribe domains can be found in the Configuration area under Section 8.3.3.3, “Publish-Subscribe Domains”.

An administrator may use the M-Link Administration Interface to perform the following operations:

- View PubSub hosting domains (including FDP domains).
- List nodes under a PubSub domain.
- Create new PubSub nodes.
- Configure PubSub nodes, including affiliations.
- Destroy PubSub nodes.
- View items published under a node.
- Delete node items.

Warning: Nodes involved in FDP will be visible in the Publish-Subscribe administration area. But it is not recommended to create or delete such nodes from there.

14.2.1 Available PubSub Domains

The landing page under tab “Publish-Subscribe” displays the PubSub domains hosted by this M-Link server. The creation and removal of domains is a matter for server configuration. Refer to Section 8.3.3.3, “Publish-Subscribe Domains”.

Each domain is represented by a card showing the domain name. FDP domains are included. Clicking on a card leads to the Domain Management page for that domain. See the next section.

14.2.2 Domain Management

Listing Nodes in a Domain

A Publish-Subscribe domain management page displays the name of the domain and represents all its nodes as cards. A click on a card opens the associated node management pop-up. See Section 14.2.4, “PubSub Node Configuration and Destruction”.

The page also includes a button “Create New Node...” allowing the administrator to create a new node under this domain. See Section 14.2.3, “PubSub Node Creation”.

14.2.3 PubSub Node Creation

The “Create New Node...” button on a Domain Management page opens the node creation pop-up, which requires the input of a node name. Node names may be anything, except for some character restrictions due to RFC-3454, but form validation will alert the administrator when appropriate.

Warning: Please note that, in theory, a node name may be slightly adjusted by the server upon creation (i.e. the name on the card might be a variation of the name input).

Note: Nodes have other configuration settings than their names, but they cannot be set at this stage. See the next section.

14.2.4 PubSub Node Configuration and Destruction

Clicking on a node card brings up the node management pop-up, which includes a “Configuration” tab. M-Link supports a small number of settings which can be changed here. New values take effect as soon as the “Update” button is clicked.

Worthy of note is the *Max items* setting. The number of data items stored in the node cannot exceed this value: older items are eliminated when items are published beyond this limit. “max” is a special value implying a large, server defined maximum.

Warning: Items may also disappear irretrievably when the *Max items* setting is reduced.

The “Configuration” tab also presents a “Destroy Node...” button in the footer. Clicking that, and confirming the operation, removes the node and all its items.

14.2.5 PubSub Node Affiliations

Clicking on a node card brings up the node management pop-up, which includes an “Affiliations” tab.

- All affiliation types are displayed by default (if any), but the list can be filtered according to the check-boxes in the “Filter by” box.
- The “Add Affiliation” button can be used to add new entries. This opens a pop-up giving the choice to set a user JID, domain or group affiliation.

  Setting group affiliations requires some groups to have been defined. This can be done in the “Configuration” area under the “IM Domains” page, selecting (or adding) a domain (see Chapter 9, *Instant Messaging Domain and User Configuration*). There are then two options:

  Static groups
  : Each IM domain can define static groups. Click on the “Static Groups” option in the sidebar menu and use the “Add...” button to create a new static group. Then use the “Members” option in the sidebar menu to add members.

  Groups defined in a remote *DSA*
  : If the selected domain uses LDAP based authentication (see option “Authentication Type”), then groups defined in the remote *DSA* can be used. This *DSA* will have its own administration interface to manage the groups.
- Entries have a “Remove” button and an affiliation selector that allows subsequent affiliation changes without having to remove and re-insert the entry.
- Entries are sorted by JID and the sort order can be changed by clicking on the column header.

14.2.6 Viewing and Deleting PubSub Node Items

Clicking on a node card brings up the node management pop-up, which includes an “Items” tab. This tab displays a list of all currently published items by ID (not items that have been deleted or purged). Rows also include a fragment of the item content and a “Delete” button.

- Clicking on a content fragment opens a pop-up displaying the full item content.
- Clicking the “Delete” button, and confirming the operation, removes the item from the node.

Chapter 15 Form Discovery and Publishing Administration

This chapter introduces the *Form Discovery and Publishing* (FDP) concept and covers how to create and administer FDP topics.

If the M-Link server is activated as either an M-Link User Server or an M-Link MU Server, the M-Link Administration Interface will feature an “FDP” tab providing administrators with Form Discovery and Publishing management functions.

15.1 Form Discovery and Publishing Concepts

M-Link User Server and M-Link MU Server products support *Form Discovery and Publishing* (FDP). An FDP domain may be thought of as a repository of forms. It holds blank forms, which users can select, fill out and submit, and collects the submitted forms. These entities are grouped into topics. Each topic is made up of

- a single form template, which defines a blank form that users can fetch.
- a collection of submitted forms, also known as published forms.
- an optional XSLT stylesheet; this may be used by some clients to transform submitted FDP forms into another format.

Topics are created by administrators who can also decide who may submit forms (publish) to which topics.

Form Discovery and Publishing is implemented as a special case of Publish-Subscribe with additional rules, but this is transparent to the users. For full details on *Form Discovery and Publishing* (FDP) refer to XMPP extension Form Discovery and Publishing (*XEP-0346*).

15.2 FDP Administration

Note: The facilities for creation, configuration and removal of FDP domains can be found in the Configuration area under Section 8.3.3.3, “Publish-Subscribe Domains”.

An FDP administrator may use the M-Link Administration Interface to perform the following operations:

- View FDP hosting domains.
- List topics under an FDP domain.
- Create a new topic by uploading a form template.
- Update a topic template.
- Control who may submit forms per topic.
- Remove a topic.
- Update XSLT Stylesheet for a topic

- Remove XSLT Stylesheet for a topic

15.2.1 Available FDP Domains

The landing page under tab “FDP” displays the PubSub domains hosted by this M-Link server which support Form Discovery and Publishing. The creation and removal of such domains is a matter for server configuration. Refer to Section 8.3.3.3, “Publish-Subscribe Domains”.

Each domain is represented by a card showing the domain name. Clicking on a card leads to a domain specific page. See the next section.

15.2.2 Listing FDP Topics

An FDP domain management page displays the name of the domain and represents all its topics as cards. A click on a card opens the associated topic management pop-up. See Section 15.2.4, “Updating or Deleting FDP Topics”.

The page also includes a button “Add Topic...” allowing the administrator to create a new topic under this domain. See Section 15.2.3, “FDP Topic Creation”.

15.2.3 FDP Topic Creation

Clicking on the “Add Topic...” button on an FDP domain management page opens the “Add Topic” pop-up. This requires the input of a topic name and the upload of a form template file. An optional XSLT stylesheet can also be provided.

- There are very few limitations on topic names, similarly to PubSub node names. RFC-3454 applies as topic names may appear in the resource part of a JID, but form validation will alert the administrator when appropriate.

  Warning: Please note that, in theory, a node name may be slightly adjusted by the server upon creation (i.e. the name on the card might be a variation of the name input).

- The template file should be a valid *XEP-0004* XML file that defines the form structure and the fields that users can fill out.

  Warning: At the moment, invalid forms are not rejected on upload!

- A number of example form templates are provided in the *$(SHAREDIR)/examples/fdp-templates* directory.

15.2.4 Updating or Deleting FDP Topics

Any form in use (topic) may need alteration. This can be done by uploading a new form template. This operation replaces the current template without invalidating already submitted forms.

Clicking on a topic card brings up the topic management pop-up, which includes a “Topic Configuration” tab. This tab displays the current form template and a “Choose a file...” button.

- The template viewer updates immediately when a new template file is uploaded via the file chooser.
- Apply the changes by clicking the “Update” button or use the “Cancel” button.

The “Topic Configuration” tab also presents a “Delete Topic...” button in the footer. Clicking that, and confirming the operation, removes the topic and all its submitted forms.

15.2.5 Controling FDP Publishers

Clicking on a topic card brings up the topic management pop-up, which includes a “Topic Publishers” tab. There are two options on this tab:

Anyone
: When this option is selected, any user can submit forms to this topic, and the tab displays nothing else.

Specified Users
: When this option is selected, the tab displays a list of users who may submit forms to this topic.

- The “Add Publisher” button can be used to add new entries. This opens a pop-up giving the choice to set a user JID, domain or group affiliation.

  Setting group publishers requires some groups to have been defined. This can be done in the “Configuration” area under the “IM Domains” page, selecting (or adding) a domain (see Chapter 9, *Instant Messaging Domain and User Configuration*). There are then two options:

  Static groups
  : Each IM domain can define static groups. Click on the “Static Groups” option in the sidebar menu and use the “Add...” button to create a new static group. Then use the “Members” option in the sidebar menu to add members.

  Groups defined in a remote *DSA*
  : If the selected domain uses LDAP based authentication (see option “Authentication Type”), then groups defined in the remote *DSA* can be used. This *DSA* will have its own administration interface to manage the groups.

- Entries have a “Remove” button.
- Apply the changes by clicking the “Update” button or use the “Cancel” button.
- Entries are sorted by JID and the sort order can be changed by clicking on the “Jabber ID” column header.

15.2.6 Updating or Removing XSLT Stylesheets

The topic management pop-up also includes an “XSLT Stylesheet” tab. This tab allows the administrator to upload a new or remove an existing XSLT stylesheet for the topic.

The tab displays the current XSLT stylesheet and a “Choose a file...” button. The stylesheet viewer updates immediately when a new stylesheet file is uploaded via the file chooser. Apply the changes by clicking the “Update” button.

The tab also includes a “Delete XSLT” button, which removes the current XSLT stylesheet.

Chapter 16 Monitoring

This chapter introduces the features available for monitoring the state of the server.

16.1 Sessions Listing and Monitoring

Sessions connecting the local M-Link server to remote servers or to clients have different properties and are monitored in a similar way but on separate pages. Hence the following two sub options or cards in the `Monitoring` area for displaying a list of current sessions of the corresponding kind:

- `Server Sessions`
- `Client Sessions` (only available for M-Link products that support *Instant Messaging* (IM))

The lists are snapshots: use the refresh button provided on the page to get an updated list, or enable `Auto Refresh` and choose a refresh rate from the drop-down menu beside it.

The second half of the page displays traffic data from the sessions that have been selected using the check boxes in the left hand side column. It displays stream fragments (XML) carried by the sessions since they were selected. This is transient data which is lost when leaving the page. Deselecting a session stops data collection for that session and also removes any already collected data from the display. All traffic data can also be reset by clicking its `Clear` button (with a trash icon).

To monitor traffic data from new sessions as they are created, use the checkbox at the top. Checking it has the effect of automatically checking the monitoring checkbox for any new session created while this top checkbox has a check mark.

Warning: Traffic data may accumulate while automatically monitoring new sessions.

Unchecking the top checkbox will only disable the automatic selection facility; it will not suspend monitoring for sessions being monitored.

The session list may be empty if there are no sessions of the selected kind.

16.1.1 Server Session Properties

The list of server sessions presents the following session properties:

A monitoring checkbox
: This checkbox controls whether or not traffic from the session is included in the traffic data monitor window in the second half of the page.

`SID`
: A unique identifier for the session.

`Type`
: A code for the session type. There are three types of sessions supported at the moment: `S2S`, `X2X`, `GCXP`. For details about these types see Section 4.2, “Peer Controls” and Section 4.6, “Links”.

`Dialback`
: When an incoming session authenticates using the dialback mechanism, a separate session is needed to interrogate the calling server according to DNS. If no session to

the relevant server already exists, a new outgoing session is initiated which will be indicated in this column.

`TLS`
: This column indicates whether the session is protected with TLS.

`Authentications`
: This lists the XMPP domain endpoint pairs (local and remote) for which this session has been authenticated (allowing stanza routing). Specialised connection mechanisms authenticate based on Peer Control configuration and for those endpoint pairs will appear here once stanzas for that pairing have been exchanged. When Relaying is in use the 'local' domain of the pair will be the domain on to which M-Link will be forwarding the received stanzas, or on behalf of which it is forwarding.

`Initiated`
: This column indicates whether the local server (`Outbound`) or the remote server (`Inbound`) initiated the session.

`Direction`
: This column indicates whether the session is able to send (`Outgoing`) or receive (`Incoming`) stanzas, or both (`Bidirectional`).

`Local Port`
: Sessions are underpinned by a TCP/IP socket uniquely identified by IP address and port number. This column uses the standard IP notation (IPv4 or IPv6 as applicable) with port separated by a colon.

`Remote Port`
: As the `Local Port`, but indicating the IP/port pair used by the remote server.

`Created On`
: The moment the session was created in the web browser's locale.

`End`
: The last column includes a button that allows termination of the session. Clicking on the button only displays a confirmation dialogue from which it is still possible to step back.

16.1.2 Client Session Properties

The list of client sessions presents the following session properties:

A monitoring checkbox
: This checkbox controls whether or not traffic from the session is included in the traffic data monitor window in the second half of the page.

`SID`
: A unique identifier for the session.

`Type`
: Indicates whether the session uses the BOSH protocol or is transported by TCP only.

`JID`
: The Jabber identifier used by the connected client. This will include a resource part.

`TLS`
: This column indicates whether the session is protected with TLS.

`Local Port`
: Sessions are underpinned by a TCP/IP socket uniquely identified by IP address and port number. This column uses the standard IP notation (IPv4 or IPv6 as applicable) with port separated by a colon. This field is blank for BOSH sessions.

`Remote Port`
: As the `Local Port`, but indicating the IP/port pair used by the client.

`Created On`
: The moment the session was created in the web browser's locale.

End
: The last column includes a button that allows termination of the session. Clicking on the button only displays a confirmation dialogue from which it is still possible to step back.

Chapter 17 HTTP API

This chapter covers the APIs available for programmatic configuration of M-Link.

17.1 Authentication HTTP API

M-Link supports two authentication methods:

- Access by login, which is used by the M-Link Administration Interface, and
- access by application access tokens, which are meant for external application programmatic access.

17.1.1 Logging in as an Administrator

URL	/api/login
Method	POST
Request Content-Type	application/json

17.1.1.1 Description

The method allows user authentication based on a cleartext password.

17.1.1.2 Request

The request body contains values for username and password in the following JSON format:

```
{
    "username": "admin",
    "password": "adminpass"
}
```

17.1.1.3 Response

The response has no content. If successful, it includes a Set-Cookie header setting the "`M-LinkLoginToken<admin port>`" cookie.

This cookie must be included in the Cookie header of all subsequent requests that require it, as specified in their related sections. Failure to include the loginToken cookie will fail these requests with a status code of 403 (FORBIDDEN) and no content in the response body.

17.1.1.4 Status Codes

204 (NO CONTENT)	Correct credentials were provided and cookie "`M-LinkLoginToken<admin port>`" is set as HTTP only (so the browser environment does not have access to it) and possibly secure as well (cookie forbidden on HTTP) depending on option `secureCookie`.
403 (FORBIDDEN)	The credentials provided were incorrect.

17.1.2 Acquiring the Security Token

URL	/api/token
Method	GET
Response Content-Type	application/json

17.1.2.1 Description

Provides a CSRF token to authenticate users.

17.1.2.2 Request

The request has no content.

17.1.2.3 Response

If a valid login token is set, responds with 200 (OK) and a body containing the token as in the example below.

```
{
    "token": "..."
}
```

The client should store this token for later use in the X-Auth-Token header of requests that require it (please refer to their own sections). Failure to include this header with a valid token will fail these requests with a status code of 403 (FORBIDDEN) and no content in the response body.

Clients SHOULD fetch a new CSRF token after an hour, as they expire after two hours.

17.1.2.4 Authentication

The request must have a cookie obtained from the /api/login end point. Failure to provide a valid loginToken cookie will cause a return with an error status of 403 (FORBIDDEN) and an empty response body.

17.1.2.5 Status Codes

200 (OK)	A valid login token was set. The body contains: { "token": "..." }.
403 (FORBIDDEN)	No login token was present in the request or it was invalid.

17.1.3 Who Is Logged in?

URL	/api/me
Method	GET
Response Content-Type	application/json

17.1.3.1 Description

This end point returns the user name of the currently logged in administrator. The request must have a cookie obtained from /api/login.

17.1.3.2 Request

The request has no content.

17.1.3.3 Response

The response body will be a JSON object. If no initial administrator has been set up, the "username" will not be present. The response object may additionally hold values for internal use by the M-Link Administration Interface, the format of which may vary between releases, and should therefore not be relied upon.

```
{
    "username": "$username"
}
```

17.1.3.4 Authentication

The request must have a cookie obtained from the /api/login end point, if the initial admin user has been set up. Failure to provide a valid loginToken cookie then causes a return with an error status of 403 (FORBIDDEN) and an empty response body.

17.1.3.5 Status Codes

200 (OK)	If the current session is fully authenticated, the response will contain a "username". Servers that do not have any administrative users configured yet will also respond with a 200 (OK), but the response object will not contain a "username".
403 (FORBIDDEN)	The initial admin user is set up, but the login token cookie was missing or invalid. The body contains no "username".

17.1.4 Changing Administrator username and/or password

URL	/api/change_credentials
Method	POST
Request Content-Type	application/json

17.1.4.1 Description

The method allows the administrative user's name or cleartext password to be changed. The current administrative username and password must be provided in addition to the desired new password and optionally new username.

17.1.4.2 Request

The request body contains values for existing and new username and password in the following JSON format:

```
{
    "username": "admin",
    "password": "adminpass",
    "newUsername": "administrator",
    "newPassword": "secret"
}
```

17.1.4.3 Response

The response has no content.

17.1.4.4 Authentication

The request must have a cookie obtained from the /api/login end point and a token obtained from the /api/token end point. Alternatively, the request may include a valid access token delivered in an Authorization header. Failure to comply will cause a return with an error status (such as 403 (FORBIDDEN)); please refer to the Authentication HTTP API for information on how to get the tokens.

17.1.4.5 Status Codes

204 (NO CONTENT)	Correct current credentials were provided and the new username and password have been stored.
422 (UNPROCESSABLE ENTITY)	The request body was invalid, or is missing required information, or the current credentials provided were incorrect.

17.1.5 Authentication by Access Token

As an alternative to the login cookie and, where applicable, the security token, external applications may access M-Link end points programmatically by including an `Authorization` header in their HTTP requests. See RFC-7235.

Tip: Syntax reminder:

```
Authorization: <scheme-name> <credentials>
```

Please note: The scheme name and credentials string are separated by a single space.

M-Link uses a proprietary authentication scheme named `Isode-API-Token` (scheme names are case insensitive), which carries a token string as credentials. For example:

```
Authorization: Isode-API-Token xLSNPnA+eJIImp2TKxt7qLHzO791SNaT
```

Application access tokens can only be obtained manually using the M-Link Administration Interface (see section Access Tokens).

17.2 Configuration HTTP API

This section describes the HTTP REST interface which allows third party clients to configure options in the option database.

The options can be thought of as held in a JSON document, described by a JSON Schema, which can be obtained from the /api/config/schema end point. The standard Schema has been extended with the following elements (please note: these are not always valid in combination):

```
{
    "properties": {
        "choice": {
            "isode_label": "someName",
            "isode_required": true,
            "isode_validator_type": "aValidatorName",
```

```
            "isode_multi_line": true,
            "isode_underlying_type": "NotNegativeExample",
            "isode_enum_titles": [
                "Display This Instead of the First Enum Value",
                "Display This Instead of the Second Enum Value"
            ],
            "isode_default_comes_from": "JSON Pointer",
            "isode_only_if": [
                {
                    "check": "is",
                    "ref": "1/enabled",
                    "value": true
                }
            ],
            "isode_store_uri": "/api/store/someObjectStore",
            "isode_substore_item": true,
            "isode_icon": "address-book"
        }
    }
}
```

isode_label
: This property is appropriate for lists of complex options (i.e. with nested sub options). It refers by name to one of the sub options, singling it out as distinctive of the list item. For example, the label might designate a sub option meant to be unique to each item and thus suitable as an item identifier. Lists of simple options do not require any label as they only carry one value per item.

isode_multi_line
: Indicates new-line characters in the option value are to be preserved.

isode_validator_type
: Denotes a validator which constrains string values to certain formatting rules. When taking input for an option with a validator, the GUI should validate it using end point /api/config/validator.

isode_underlying_type
: An informative property naming the underlying type of the option. It might be used to identify options that the end point consumer can validate by itself. However, it should only appear along side the "isode_validator_type" property.

isode_enum_titles
: Short definitions of the choices identified by an enum option. The option should also have the standard "enum" property, listing the enum values in the same order.

isode_only_if
: In the absence of any isode_only_if, the values of options fetched from the /api/config end point are either their default if unset, or the value they were last set to. An isode_only_if property is a boolean value computed as the logical AND of all the conditions it lists. The clauses in the isode_only_if connect the value of an option, referred to by a JSON Pointer (the "ref" element) to a constant value (the "value" element) with an operator (the "check" element) which is either "is" or "isNot". If the isode_only_if property is true, the behaviour is the same as when there is no isode_only_if property. If it is false, fetching the option value returns its default, whether or not the option was explicitly set. Please note that there are several ways to generate a default. If the option is complex (that is, it has nested sub options), a false isode_only_if property is logically applied to all the sub options, overriding any explicit "isode_only_if" on sub options. If the option is a list, a false "isode_only_if" makes it appear empty. While the "isode_only_if" is false, set values are not lost and options may still be set, modified or unset.

isode_default_comes_from
: Extends schema property default: instead of a value, refers to another option, via a JSON Pointer, from which to collect a default value. In cases where a default value is returned (e.g. fetching an unset option or fetching an option with false "isode_only_if"), "isode_default_comes_from" takes precedence over "default".

Multiple options may be chained with "isode_default_comes_from". In other words, the target of an "isode_default_comes_from" may be unset or have a false "isode_only_if", in which case its own default is retrieved, wherever that comes from. Please note that some types have a natural default, meaning that even without any "isode_default_comes_from" or "default" a fetch will succeed. Other types must have some form of explicit default.

isode_required
: Indicates that a value must be provided when patching the configuration. Defaults to false if missing on an option. This is similar to the standard "required" property. However, this extension accommodates the "isode_only_if" extension: false "isode_only_if" conditions effectively disable the requirement to be set, as indeed only the default can then be fetched. In other words, unset options with "isode_required" but a false "isode_only_if" do conform to the schema (which would not be the case with a standard "required" property).

isode_store_uri
: Indicates that the option value is the unique identifier of an object held in an object store. Object stores are persistent containers of objects of a type specific to the store. These can be manipulated through a separate API (see section Object Stores HTTP API). The value of the isode_store_uri property is the path to the object store according to that API, relative to the server URI.

isode_substore_item
: If true, indicates that the option value is the unique identifier of an object within a substore. Substores are object stores which are themselves items of an object store. Please refer to section Object Stores HTTP API. The identifier only makes sense when knowing which substore it belongs to: Options with this property are expected to exist within a complex option also including a sub option with the same name plus "Substore" and providing the required information (see property isode_store_uri). Property isode_substore_item is incompatible with isode_underlying_type: Types with the latter are expected to support a validator (implicitly or explicitly), however substore items require validators which cannot be triggered via the /api/config/validator endpoint. They may still have validators applied by the server after calling PATCH /api/config, but those are not declared in the JSON Schema.

isode_icon
: The value is only a hint to the UI client, drawing meaning by convention: If it is appropriate for the option, the UI may interpret the value as the class name of an icon/glyph within some version of FontAwesome (tm) and use it where appropriate in relation to the rendering of that option.

Available operations are described below. Some end points allow multiple methods. They allow configuration retrieval, option update through JSON patches and validation. A status code of 200 (OK) implies that the request was successful.

17.2.1 Obtaining the JSON Schema

URL	/api/config/schema
Method	GET
Response Content-Type	application/json

17.2.1.1 Description

As mentioned in the overview of Section 17.2, “Configuration HTTP API”, the set of configuration options presents as a JSON document. In order to set values properly and derive defaults, it is necessary to retrieve a JSON Schema describing the whole document.

17.2.1.2 Request

Fetches the JSON Schema for the application options. The request body is empty.

17.2.1.3 Response

Returns the JSON Schema as a JSON document in the body.

Example:

```
{
    "$schema": "http://json - schema.org/draft - 04/schema#",
    "description": "Server configuration",
    "id": "http://isode.com/api/config/schema.json",
    "properties": {
        "useTls": {
            "default": true,
            "description": "Boolean value",
            "type": "boolean"
        },
        "max_stanza_size": {
            "default": 10024,
            "description": "Maximum size of a stanza",
            "maximum": 9223372036854775807,
            "minimum": -9223372036854775808,
            "type": "integer"
        }
    },
    "propertyOrder": [
        "max_stanza_size",
        "useTls"
    ],
    "title": "Config",
    "type": "object"
}
```

17.2.1.4 Authentication

The request must have a cookie obtained from the /api/login end point or include a valid access token delivered in an Authorization header. Failure to comply will cause a return with an error status (such as 403 (FORBIDDEN)); please refer to the Authentication HTTP API for information on how to get the tokens.

17.2.1.5 Status Codes

Status codes relating specifically to this method:

200 (OK)	Schema successfully retrieved.
	Failure codes are possible, but only due to higher layer processing (such as authentication issues).

17.2.2 Retrieving the Current Configuration

URL	/api/config
Method	GET
Response Content-Type	application/json

17.2.2.1 Description

Retrieves the current configuration in JSON format. This configuration format is described in the JSON Schema provided by the /api/config/schema end point. Please note that the document may be incomplete: where options have not been explicitly set they are omitted, as their values can be derived either from the default included in the schema, or from an isode_default_comes_from, or the type has a natural default, such as strings which can be empty.

17.2.2.2 Request

Fetches the current configuration in JSON format. The request body is empty.

17.2.2.3 Response

The JSON document returned in the body conforms to the JSON Schema. It only includes explicitly set options.

Example:

```
{
    "max_stanza_size": 10000
}
```

17.2.2.4 Authentication

The request must have a cookie obtained from the /api/login end point or include a valid access token delivered in an Authorization header. Failure to comply will cause a return with an error status (such as 403 (FORBIDDEN)); please refer to the Authentication HTTP API for information on how to get the tokens.

17.2.2.5 Status Codes

Status codes relating specifically to this method:

200 (OK)	Successfully retrieved the configuration.
	Failure codes are possible, but only due to higher layer processing (such as authentication issues).

17.2.3 Applying Changes to the Configuration

URL	/api/config
Method	PATCH
Request Content-Type	application/json-patch+json
Response Content-Type	application/json

17.2.3.1 Description

Applies changes, described in a JSON Patch document, to the current configuration.

17.2.3.2 Request

The request body should contain a valid JSON Patch document describing the changes to be applied to the configuration, seen as a JSON document conforming to the JSON Schema provided by the /api/config/schema end point. A JSON Patch is an array of objects, each describing a single operation to be performed on a specific option referred to by its path (a JSON Pointer).

Only a sub set of JSON Patch operations are supported: replace, add, remove and move.

replace
: This operation sets an existing single option to a value. It fails if the path leads to a complex or list option.

  Example:

```
[
```

```
    {
        "op": "replace",
        "path": "/max_stanza_size",
        "value": 99000
    }
]
```

add
: This operation adds an item to a list. The path provided must point to where the item will be inserted in the list. All existing items from that point onwards will move to the right. As a special case, if path is a list option with '/-' appended to it (suggesting the index number beyond the current last), the item will be appended at the end. If the item is complex, the value is a compound JSON object matching the sub tree. Only list items can be added as the option tree is fixed.

  Example: N.B. the list assumed below does not feature in the simple schema example given out above:

```
[
    {
        "op": "add",
        "path": "/imDomains/0/users/-",
        "value": {
            "jid": "alice@wonderland.lit",
            "password": "secret"
        }
    }
]
```

remove
: This operation removes an item from a list. The path must lead to an existing list item. Other options are not removable.

  Example: N.B. the list assumed below does not feature in the simple schema example given out above:

```
[
    {
        "op": "remove",
        "path": "/imDomains/0/users/5"
    }
]
```

move
: This operation moves an item within a list from the `from` position and inserts it at the `path` position. Items can only be moved within the same list.

  The `from` path must lead to an existing list item. The index provided in `path` must be between 0 and the number of items on the list. This operation is functionally equivalent to removing an item and adding it back at the specified position.

  Example: N.B. the list assumed below does not feature in the simple schema example given out above:

```
[
    {
        "op": "move",
        "path": "/imDomains/0/users/5",
        "from": "/imDomains/0/users/2"
    }
]
```

17.2.3.3 Response

If the patch was successfully applied, the new configuration is returned in the response body in the same way as in response to a GET /api/config.

Example:

```
{
    "max_stanza_size": 99000,
    "useTls": true
}
```

If the patch failed, the body is empty.

17.2.3.4 Authentication

The request must have a cookie obtained from the /api/login end point and a token obtained from the /api/token end point. Alternatively, the request may include a valid access token delivered in an Authorization header. Failure to comply will cause a return with an error status (such as 403 (FORBIDDEN)); please refer to the Authentication HTTP API for information on how to get the tokens.

17.2.3.5 Status Codes

Status codes relating specifically to this method:

200 (OK)	The patch was successfully applied.
415 (UNSUPPORTED MEDIA TYPE)	The content type or the JSON syntax was invalid.
422 (UNPROCESSABLE ENTITY)	The JSON Patch provided was invalid or could not be applied.
	Other failure codes are possible, but only due to higher layer processing (such as authentication issues).

17.2.4 Validating Option Values

URL	/api/config/validator
Method	POST
Request Content-Type	application/json
Response Content-Type	application/json

17.2.4.1 Description

This end point provides a facility to validate a particular option value before submitting it to end point /api/config with PATCH. Most value types are fully defined in the JSON Schema. For example, integers might have a maximum and minimum. Other types are defined as strings but are actually more elaborate and/or have specific formatting restrictions that cannot be easily expressed in the schema. These will have a "validator_type" property with which to query the option database through this end point.

17.2.4.2 Request

The body is a JSON document defining a type and value to validate. The type is the value of a "validator_type" property and should be recognized by the option database.

Example:

```
{
    "type": "typeName",
    "value": "valueToTest"
}
```

17.2.4.3 Response

If the validator ("type" in the request) is supported and the value valid, the body contains:

```
{
    "valid": true
}
```

If the validator is supported but the value invalid, the body content resembles the following, the value of userMessage being supplied by the validator.

```
{
    "valid": false,
    "userMessage": "a hint"
}
```

17.2.4.4 Authentication

The request must have a cookie obtained from the /api/login end point or include a valid access token delivered in an Authorization header. Failure to comply will cause a return with an error status (such as 403 (FORBIDDEN)); please refer to the Authentication HTTP API for information on how to get the tokens.

17.2.4.5 Status Codes

Status codes relating specifically to this method:

200 (OK)	The validator is supported. If the value is valid, the body contains: { "valid": true }, otherwise: { "valid": false, "userMessage": "a hint" }
400 (BAD REQUEST)	Unsupported content type or unsupported validator. In the latter case, the body contains: { "error": "message" }
	Other failure codes are possible, but only due to higher layer processing (such as authentication issues).

17.3 Object Stores HTTP API

This section describes the HTTP end points for the management of Object Stores. All Object Store end points require a valid login token. The calls that modify the store or its contents will additionally require a CSRF token.

The API provides access to two kinds of stores: Object Stores and Object Substores. Object Substores provide another hierarchy layer to Object Stores where each item in the Object Store is another Object Store containing items - e.g. every Security Label Catalog in the Security Label Catalog Store can be addressed as a substore, providing access to all the labels in the catalog. Substores are immutable, so any operation attempting to modify a substore will fail.

Each available Store has a base path associated with it. Store paths will be represented by */`storepath`* in this section. The base paths for most stores can be found from the *GET /api/stores* endpoint.

The base paths for substores are generated by taking the path of the base store, */`storepath`*, appending the id of the item in the base store that you want to use as a substore, and appending */sub_store*. For example, if a Security Label Catalog store was located at */api/stores/label_catalogs*, and contained a catalog with id 'a64ef', */api/stores/label_catalogs/a64ef/sub_store* would be the base path for a substore containing the labels provided by that catalog.

17.3.1 List available Object Stores

URL	/api/stores
Method	GET
Response Content-Type	application/json

17.3.1.1 Description

This method can be used to discover the object stores that are available.

17.3.1.2 Request

The request has no content.

17.3.1.3 Response

Returns an array of objects. Each object has at least the members `name`, `path`, `description`, and `upload_fields` as shown. The `path` member represents the base path of each store. If the `allowDownloads` member is present and set to true, then items in this store may be downloaded. If the `has_substores` member is present and set to true, this store has substores. The `upload_fields` member is an array of specifications of the fields which the store requires when creating a new entry. In the simple case, this is a single file containing the data of the entry, but more complex stores may require multiple components to be supplied (e.g. a certificate chain, private key and passphrase for a TLS identity).

```
[
    {
        "name": "Example Store",
        "path": "/api/stores/example",
        "allowDownloads": true,
        "description": "All available examples are stored here.",
        "upload_fields": [
            {
                "name": "object",
                "type": "file",
                "title": "Filename",
                "description": "Choose a file to upload",
                "accept": ".pem,.crt",
                "required": true
            }
        ]
    },
    {
        "name": "Example Store Containing Substores",
        "path": "/api/stores/example_catalogs",
        "description": "Example catalogs are stored here.",
        "has_substores": true
    }
]
```

17.3.1.4 Authentication

The request must have a cookie obtained from the /api/login end point or include a valid access token delivered in an Authorization header. Failure to comply will cause a return with an error status (such as 403 (FORBIDDEN)); please refer to the Authentication HTTP API for information on how to get the tokens.

17.3.1.5 Status Codes

200 (OK)	The request was successful.
403 (FORBIDDEN)	The login token cookie was missing or invalid.

17.3.2 List all objects in a Store or Substore

URL	/*storepath*
Method	GET
Response Content-Type	application/json

17.3.2.1 Description

This call returns a list of all items in the specified Object Store or Object Substore.

17.3.2.2 Request

The request has no content.

17.3.2.3 Response

Returns an array of objects. Each object consists of the members `id`, `name` and an optional `creationDate`, in seconds since 1970-01-01T00:00:00Z. `name` can be used for display purposes. The `id` of a given object will not change, and is used in methods that operate on a specific stored item.

```
[
    {
        "id": "7b5237b4-c3ea-4f9f-8d16-20eb96486b0e",
        "name": "An example object",
        "creationDate": 1582798994
    },
    {
        "id": "174edbfa-e2b3-4e5c-b5bb-d3783a364e3d",
        "name": "Another object",
        "creationDate": 1585582903
    }
]
```

17.3.2.4 Authentication

The request must have a cookie obtained from the /api/login end point or include a valid access token delivered in an Authorization header. Failure to comply will cause a return with an error status (such as 403 (FORBIDDEN)); please refer to the Authentication HTTP API for information on how to get the tokens.

17.3.2.5 Status Codes

200 (OK)	The request was successful.
403 (FORBIDDEN)	The login token cookie was missing or invalid.
404 (NOT FOUND)	The requested store does not exist.

17.3.3 Add an object to the Store

URL	/*storepath*
Method	POST
Request Content-Type	multipart/form-data
Response Content-Type	javascript/json

17.3.3.1 Description

This call can be used to add an object to the specified store.

The object content is wrapped within a multipart/form-data, as explained in the request section below. Its *filename* parameter is meant to record the name of the file the object is uploaded from. The name of the new object in the store may also be set using a `name` field, but defaults to the filename. It is returned in the response as the value of property "name", along with property "id" which is generated by the POST operation as a unique identifier. Other attributes may also exist, but those are object type specific.

17.3.3.2 Request

The call expects a valid multipart/form-data payload.

You can provide form values for each field mentioned in the `upload_fields` entry from the HTTP GET which discovered the Object Store. Fields marked as required must always be sent.

Fields of type `file` must have their *filename* attribute set. When multiple fields of type `file` are submitted, the object will be stored using the first filename encountered.

This is compatible with regular HTTP file upload mechanisms found in most web clients. Multiple objects may thus be stored in a single request. Objects should a Content-Type values of application/octet-stream, or something more specific.

The multipart/form-data payload may also include a field named `name`. Its value is an arbitrary name to be associated with the object. It must have a Content-Type of text/plain with charset of UTF-8, but these are default values. If omitted, the *filename* parameter of the first field which contains one will be used.

17.3.3.3 Response

If the call succeeds, the same object information as returned by 'Get object information'. In case of an error, the body will be an object with a `message` member with a user-presentable error.

17.3.3.4 Authentication

The request must have a cookie obtained from the /api/login end point and a token obtained from the /api/token end point. Alternatively, the request may include a valid access token delivered in an Authorization header. Failure to comply will cause a return with an error status (such as 403 (FORBIDDEN)); please refer to the Authentication HTTP API for information on how to get the tokens.

17.3.3.5 Status Codes

200 (OK)	The request was successful.
403 (FORBIDDEN)	The login token cookie was missing or invalid.
404 (NOT FOUND)	The requested store does not exist.
422 (UNPROCESSABLE ENTITY)	The request body was invalid, or is missing required information.

17.3.4 Retrieve an object from the Store

URL	/*storepath*/*id*
Method	GET
Response Content-Type	*Content specific*

17.3.4.1 Description

This call returns the object data for an item in an Object Store. Unless the store holds composite objects which are not convertible to a single file, in which case this feature is disabled and a retrieval attempt yields an error.

17.3.4.2 Request

The request has no content.

17.3.4.3 Response

Returns the data associated with object *id* in the specified store. The `Content-Type` will depend on the object stored. In case of an error, the Content-Type will be `application/json`, and the body will be an object with a `message` member with a user-presentable error.

17.3.4.4 Authentication

The request must have a cookie obtained from the /api/login end point or include a valid access token delivered in an Authorization header. Failure to comply will cause a return with an error status (such as 403 (FORBIDDEN)); please refer to the Authentication HTTP API for information on how to get the tokens.

17.3.4.5 Status Codes

200 (OK)	The request was successful.
403 (FORBIDDEN)	The login token cookie was missing or invalid.
404 (NOT FOUND)	The requested store or item does not exist or the store does not support content retrieval.

17.3.5 Download an object from the Store

URL	/api/stores/*storepath*/*id*
Method	GET
Response Content-Type	application/octet-stream

17.3.5.1 Description

Returns the data associated with the object *id* in the specified store. For stores which hold composite objects or which do not have `allowDownloads` set to `true`, this feature is disabled and will return an error.

17.3.5.2 Request

The request has no content.

17.3.5.3 Response

Returns the data associated with object *id* in the specified store.

17.3.5.4 Authentication

The request must have a cookie obtained from the /api/login end point or include a valid access token delivered in an Authorization header. Failure to comply will cause a return with an error status (such as 403 (FORBIDDEN)); please refer to the Authentication HTTP API for information on how to get the tokens.

17.3.5.5 Status Codes

200 (OK)	The request was successful.
404 (NOT FOUND)	The requested store or item does not exist or the store does not support downloading of items.

17.3.6 Rename an object

URL	/*storepath*/*id*
Method	POST
Request Content-Type	javascript/json
Response Content-Type	javascript/json

17.3.6.1 Description

This call updates the name of the specified object in the Object Store.

17.3.6.2 Request

The request body must be an object with a single *name* member that holds the desired new name of the object.

```
{
    "name": "New object name"
}
```

17.3.6.3 Response

If the call succeeds, the object identified by *id* will have its *name* updated. In case of an error, will return an object with a `message` member with a user-presentable error.

Please note that the call will fail if the object is read-only (see section Get object information).

17.3.6.4 Authentication

The request must have a cookie obtained from the /api/login end point and a token obtained from the /api/token end point. Alternatively, the request may include a valid access token delivered in an Authorization header. Failure to comply will cause a return with an error status (such as 403 (FORBIDDEN)); please refer to the Authentication HTTP API for information on how to get the tokens.

17.3.6.5 Status Codes

204 (NO CONTENT)	The request was successful.
403 (FORBIDDEN)	The login token cookie was missing or invalid.
404 (NOT FOUND)	The requested store or item does not exist.
422 (UNPROCESSABLE ENTITY)	The request body was invalid, is missing the required information, or refers to a read-only object.

17.3.7 Remove an object from the Store

URL	/*storepath*/*id*
Method	DELETE
Response Content-Type	javascript/json

17.3.7.1 Description

This call removes the specified object from the Object Store.

17.3.7.2 Request

The request has no content.

17.3.7.3 Response

If the call succeeds, the object identified by *id* will no longer be present in the Object Store. In case of an error, will return an object with a `message` member with a user-presentable error.

Please note that the call will fail if the object is read-only (see section Get object information).

17.3.7.4 Authentication

The request must have a cookie obtained from the /api/login end point and a token obtained from the /api/token end point. Alternatively, the request may include a valid access token delivered in an Authorization header. Failure to comply will cause a return with an error status (such as 403 (FORBIDDEN)); please refer to the Authentication HTTP API for information on how to get the tokens.

17.3.7.5 Status Codes

204 (NO CONTENT)	The request was successful.
403 (FORBIDDEN)	The login token cookie was missing or invalid.
404 (NOT FOUND)	The requested store or item does not exist.
422 (UNPROCESSABLE ENTITY)	The requested item is still in use in an active configuration, or is read-only.

17.3.8 Get object information

URL	/*storepath*/*id*/info
Method	GET
Response Content-Type	application/json

17.3.8.1 Description

This call returns all available additional information on an object in the specified store.

17.3.8.2 Request

The request has no content.

17.3.8.3 Response

Returns the information as an object with at least the following members: `id`, `name`.

Further there may be the following members: `creationDate`, `filename`, `contentType`, `readOnly`.

`id`
: The object's unique identifier generated when it was added to the store.

`name`
: The name chosen for the object. This may have defaulted to the filename of origin, or may have been set when adding the object or using the rename procedure.

`filename`
: The filename given when adding the object.

`contentType`
: The MIME content type given when adding the object.

`creationDate`
: The date and time the object was added. This is conveyed as a number of seconds since the Epoch (1970-01-01T00:00:00Z).

`readOnly`
: If this member is present and set to `true`, the object is read-only and cannot be removed, renamed or overwritten.

Additionally, there may be an `attributes` member with an object as value. This object will hold additional information; which attributes are available depends on the store. Attributes which have empty values will not be returned, and this may mean that the `attributes` member is empty.

```
{
    "id": "7b5237b4-c3ea-4f9f-8d16-20eb96486b0e",
    "name": "An example object",
    "filename": "example.xml",
    "contentType": "text/xml",
    "creationDate": 1574091301,
    "attributes": {
        "valid_from": "2020-01-01Z00:00:00"
    }
}
```

17.3.8.4 Authentication

The request must have a cookie obtained from the /api/login end point or include a valid access token delivered in an Authorization header. Failure to comply will cause a return with an error status (such as 403 (FORBIDDEN)); please refer to the Authentication HTTP API for information on how to get the tokens.

17.3.8.5 Status Codes

200 (OK)	The request was successful.
403 (FORBIDDEN)	The login token cookie was missing or invalid.
404 (NOT FOUND)	The requested store or item does not exist.

Appendix A Archive Admin Tool

This chapter describes the Archive Admin Tool operations to manage the M-Link archive.

A.1 Installing Archive Admin Tool

Archive Admin Tool is installed separately from the M-Link Server and is installed from different packages. It it supported on the same platforms as the M-Link Server.

The Archive Admin Tool uses the same activation file as the M-Link Server so must be installed on the same host as the M-Link Server.

Archive Admin Tool is a Java application and requires at least Java 11 to be installed on the system and the JAVA_HOME environment variable to be set.

A.2 Archive Admin Tool Archive commands

Archive Admin Tool provides a command-line interface with which you can perform certain operations on the Archive Server for an M-Link Service. The four available commands are:

export
: Exports the data from the Archive server to a file in XML or PDF/A format.

expire
: Removes data from the Archive server database prior to a specified date.

backup
: Creates a backup of the Archive SQLite3 database in database format.

import
: Imports an XML file that was previously exported from an Archive server.

Commands for Archive operations are structured as follows:

Linux:

```
% /opt/isode/bin/archiveadmintool (COMMAND)
 (MANDATORY_PARAMETERS)
 (COMMAND_PARAMETERS)
```

Windows (note: a default path for the Java 17 JRE is assumed in this example):

```
C:\> cd C:\Program Files\Isode\bin
C:\Program Files\Isode\bin> java.exe
 -jar java\classes\isode-archiveadmintool.jar (COMMAND)
 (MANDATORY_PARAMETERS)
 (COMMAND_PARAMETERS)
```

Every Archive command requires an authentication token. This token can be generated using the M-Link Administration Interface, see Section 8.3.3.10, “Access Tokens”. The

token must be typed after the `--token` option (possibly abreviated to `-t`), as shown below.

-t / --token *token*
: Where *token* is a valid access token obtained from the M-Link Administration Interface using option **Access Tokens** from the **Configuration** sidebar menu. This option is mandatory! It is advised to surround the token with double quotes on Windows, as it may contain forward slash characters.

In addition to operation specific optional parameters, there are several optional parameters that are applicable to all Archive commands:

-s / --silent
: Run the command in silent mode. This will disable any output logged to the command line. This is not recommended for a first run.

-host / --host *hostname*
: Where *hostname* is the HTTP hostname to use for the command. This shouldn't be needed in most cases, as the default value is `localhost`.

-port / --port *number*
: Where *number* is the HTTP port number to use for the command. This shouldn't be needed in most cases, as the default value is the default port number for the M-Link Archive Server (`5080`).

-tls / --tls
: Specifies TLS encryption, using the HTTPS protocol.

-cf / --certfile *filename*
: Specifies a certificate file that contains one or more pinned certificates in PEM format to use for verifying the M-Link Archive Server certificate. If omitted, the Archive Admin Tool will interactively suggest trusting the certificate and/or saving it for future use with this option.

-tf / --tafile *filename*
: Specifies one or more trust anchors (root CA certificates) in PEM format to use for verifying server certificate.

A.2.1 Import parameters

The Archive import command can be used to import data into the Archive database from an Archive XML file-based format.

-if / --impfn *xmlfile*
: [mandatory] Where *xmlfile* is the file to import. This should be an XML file containing valid archive data.

An example of an import command is included below:

Linux:

```
% /opt/isode/bin/archiveadmintool import -t secret-token-value
 -tls -cf servercert.pem -if archive.xml
```

Windows:

```
C:\>cd C:\Program Files\Isode\bin
C:\Program Files\Isode\bin> java.exe
 -jar java\classes\isode-archiveadmintool.jar import
 -t "secret-token-value"
 -tls -cf servercert.pem
 -if archive.xml
```

A.2.2 Export parameters

The Archive export command can be used to export archive data to XML or PDF/A file format.

-dom / --domain *domain*
Where *domain* is the domain name to which this operation applies.

-sdate / --startdate *date-time*
Where *date-time* is the date-time after which all archive data will be exported. This should be in the form dd/MM/yyyy'T'HH:mm:ss, e.g. 03/07/2024T12:35:00.

-edate / --enddate *date-time*
Where *date-time* is the date-time before which all archive data will be exported. This should be in the form dd/MM/yyyy'T'HH:mm:ss, e.g. 04/07/2024T12:35:00.

-xml / --isXML
Specifies XML as the type of the exported file. If omitted, the type is PDF/A.

-ef / --expfn *filename*
[mandatory] Where *filename* is the path and name for the output file, e.g. /home/user/output.xml. The path prefix should be a valid directory on the file system. The Archive Admin Tool will not automatically append any type denoting extension.

An example of an export command is included below:

Linux:

```
% /opt/isode/bin/archiveadmintool export
 -t secret-token-value -tls
 -cf servercert.pem -ef archive.pdf
```

Windows:

```
C:\>cd C:\Program Files\Isode\bin
C:\Program Files\Isode\bin> java.exe
 -jar java\classes\isode-archiveadmintool.jar export
 -t "secret-token-value"
 -tls -cf servercert.pem
 -ef archive.pdf
```

A.2.3 Expire parameters

The expire command is used to remove data before a specified date-time. Note that this operation can not be reversed.

-dom / --domain *domain*
Where *domain* is the domain to which this operation applies.

-exdate / --expdate *date-time*
[mandatory] Where *date-time* is the expiry date before which all messages will be removed. This should be in the form dd/MM/yyyy'T'HH:mm:ss, e.g. 03/09/2023T12:35:00.

An example of the expire command is included below:

Linux:

```
% /opt/isode/bin/archiveadmintool expire -t secret-token-value
 -tls -cf servercert.pem -exdate 03/09/2023T12:35:00
```

Windows:

```
C:\>cd C:\Program Files\Isode\bin
C:\Program Files\Isode\bin> java.exe
 -jar java\classes\isode-archiveadmintool.jar expire
 -t "secret-token-value"
 -tls -cf servercert.pem
 -exdate 03/09/2023T12:35:00
```

A.2.4 Backup parameters

The Archive backup command can be used to create a backup of the Archive SQLite3 database, as a database file (.db). There are no optional parameters for the backup command, except the standard optional parameters defined in Section A.2, “Archive Admin Tool Archive commands”.

Running an Archive backup operation will construct a copy of the database with the current time-stamp as the file name, and place this into the *backups* directory

- Linux: */var/isode/mlink/wabac/backups*.
- Windows: *C:\Isode\M-Link\wabac\backups*.

An example of a backup command is included below:

Linux:

```
% /opt/isode/bin/archiveadmintool backup -t secret-token-value
 -tls -cf servercert.pem
```

Windows:

```
C:\>cd C:\Program Files\Isode\bin
C:\Program Files\Isode\bin> java.exe
 -jar java\classes\isode-archiveadmintool.jar backup
 -t "secret-token-value"
 -tls -cf servercert.pem
```

A.3 Archive Admin Tool GUI

Archive Admin Tool also provides a graphical user interface (GUI) that can be used to perform export operations. It takes the same parameters as the command line interface but instead the user enters them into the GUI.

On Windows an application shortcut is created in the start menu, under `Isode 19.0`, and on Linux it can be started by running:

```
% /opt/isode/bin/archiveadmintool
```

Appendix B Integration with Isode Icon-Topo

This appendix describes how to integrate M-Link MU Gateway with Isode Icon-Topo.

B.1 Enabling support for Icon-Topo

Support for integration with Icon-Topo is enabled using the *Enable Icon-Topo Support* option in *Global Options*.

Isode Icon-Topo will need an *Access Token* to interface with M-Link MU Gateway. This token can be generated using the part of the Management Interface described in Section 8.3.3.10, “Access Tokens”. Details on how to configure M-Link integration on Icon-Topo servers can be found in its manual.

B.2 Configuration managed by Icon-Topo

Icon-Topo will create and maintain the routing information of an M-Link MU Gateway server. More specifically, it will maintain the configuration of Links, STANAG 5066 Servers and Peer Controls. As long as the *Enable Icon-Topo Support* option is set, routing information configured through Icon-Topo will always override locally made changes.

B.2.1 Links

When Icon-Topo defines a link, M-Link will first check whether there's already an existing link that was previously configured by Icon-Topo with the same name.

If it cannot find an existing Icon-Topo-managed link, it will try to find an existing link that matches. For *XEP-0361* links, an existing link with the same *Local Port* will match. For STANAG 5066 links, the STANAG 5066 Server *Host* and *Port* options, as well as the *Remote SIS Address* will be used. If it finds a matching link, it will automatically become associated with the Icon-Topo-provided name. Taking over existing links can be useful to pre-configure certain aspects (such as TLS details) of links before they are actually used for routing.

Once a suitable link is found, or a new one has been created if there was no existing matching link, M-Link will ensure that the details provided by Icon-Topo will remain configured.

The following aspects of Links are managed, maintained according to the latest configuration provided by Icon-Topo, and should not be changed manually:

- Link Name
- Link Type
- Local Port (*XEP-0361* only)
- Remote Host (*XEP-0361* only)
- Remote Port (*XEP-0361* only)
- STANAG 5066 Server (STANAG 5066 only)
- Remote SIS Address (STANAG 5066 only)

B.2.2 STANAG 5066 Server Definitions

For each STANAG 5066 link defined by Icon-Topo, M-Link will ensure there is a corresponding STANAG 5066 Server Definition. The *Local SIS Address* will be updated if it changes in a new Icon-Topo configuration.

B.2.3 Peer Controls

Icon-Topo has the ability to maintain both the *default* Peer Control, as well as any additionally configured Peer Controls (also known as *routes*.)

Peer Controls that were previously configured by Icon-Topo and are no longer present in the newest configuration will have the domain extended with `.disabled.invalid`. When a Peer Control is added for which there is an existing control with a domain ending in `.disabled.invalid`, the existing route will be updated and made active again by having the suffix stripped. These two mechanisms will allow for persistence of local changes made to routes even when a Peer is temporarily unavailable.

The following aspects of Peer Controls are managed, maintained according to the latest configuration provided by Icon-Topo, and should not be changed manually:

- Domain
- Matching Rule
- Use Link
- Link
- Inbound-Only Links
- Relay Zone

Note: Peer Controls are matched on both their *Domain* and *Matching Rule*s. Icon-Topo supports the *Domain* or *Domain and Subdomains* matching rules. Use of manually configured Peer Controls using the *Subdomains* matching rule on a system managed by Icon-Topo is not advised.

Appendix C Pre-defined Transformations

This appendix describes the Transformations that are included with M-Link.

C.1 Block common File Transfer and VOIP mechanisms

This transformation will ensure that common mechanisms for file transfer and VOIP will be removed.

The protocols covered are In-Band Bytestreams (*XEP-0047*), Stream Initiation (*XEP-0095*), and Jingle (*XEP-0166*) and their associated standards.

When using this transformation, it is recommended to set the *Action on Blocked Stanza* option on the rule to *Bounce* to ensure the blocked IQ requests are replied to with an error.

C.2 Normalize XML for M-Guard

This transformation should be used to normalize the XML sent to M-Guard servers. It should generally be the last one configured on a given peer.

This will ensure `xml:lang` attributes are all lowercase, and that whitespace is normalized.

C.3 Strip Chat State Notifications

This transformation will remove Chat State Notifications (*XEP-0085*) elements from message stanzas. Chat State Notifications are used to inform users of the state of their conversation partner in a chat session ("inactive", "composing", "paused", etc.)

Disabling these notifications may be desirable in cases where bandwidth is restricted.

C.4 Strip Legacy Delayed Delivery

This transformation removes Legacy Delayed Delivery (*XEP-0091*) elements from message or presence stanzas. The Legacy Delayed Delivery specification has been deprecated in favour of Delayed Delivery (*XEP-0203*).

Disabling Legacy Delayed Delivery and allowing Delayed Delivery ensures that only one representation of timestamps is emitted, e.g. when crossing a security guard appliance.

C.5 Strip Message Delivery Receipts

This transformation removes Message Delivery Receipts (*XEP-0184*) elements from message stanzas. Message Delivery Receipts are a mechanism that will allow a sender to ask the recipient to acknowledge receipt of a content message by returning an ack message.

Disabling Message Delivery Receipts may be desirable in cases where bandwidth is restricted.

C.6 Strip XHTML-IM parts from messages

This transformation will remove XHTML-IM (*XEP-0071*) markup from message stanzas. XHTML-IM is a method to include lightweight text markup.

Disabling XHTML-IM parts may be desirable in cases where bandwidth is restricted.

C.7 Strip vCards

This transformation removes requests and notifications concerning vCards.

Removing vCards may be desirable in cases where bandwidth is restricted.

Appendix D Standards Supported by M-Link

An overview of Open Standards M-Link products conform to.

D.1 Supported by M-Link Edge

RFC 0793 Transmission Control Protocol

RFC 1034 Domain names - concepts and facilities

RFC 1035 Domain names - implementation and specification

RFC 3454 Preparation of Internationalized Strings ("stringprep")

RFC 6120 Extensible Messaging and Presence Protocol (XMPP): Core

RFC 6121 Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence

RFC 6122 Extensible Messaging and Presence Protocol (XMPP): Address Format

XEP-0004 Data Forms

XEP-0030 Service Discovery

XEP-0082 XMPP Date and Time Profiles

XEP-0198 Stream Management
M-Link only supports Stanza Acknowledgements. Stream Resumption is not supported.

XEP-0199 XMPP Ping

XEP-0220 Server Dialback

XEP-0258 Security Labels in XMPP

XEP-0361 Zero Handshake Server to Server Protocol

D.2 Supported by M-Link MU Gateway

RFC 0793 Transmission Control Protocol

RFC 1034 Domain names - concepts and facilities

RFC 1035 Domain names - implementation and specification

RFC 3454 Preparation of Internationalized Strings ("stringprep")

RFC 6120 Extensible Messaging and Presence Protocol (XMPP): Core

RFC 6121 Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence

RFC 6122 Extensible Messaging and Presence Protocol (XMPP): Address Format

XEP-0004 Data Forms

XEP-0030 Service Discovery

XEP-0082 XMPP Date and Time Profiles

XEP-0198 Stream Management
M-Link only supports Stanza Acknowledgements. Stream Resumption is not supported.

XEP-0199 XMPP Ping

XEP-0220 Server Dialback

XEP-0258 Security Labels in XMPP

XEP-0361 Zero Handshake Server to Server Protocol

XEP-0365 Server to Server communication over STANAG 5066 ARQ

D.3 Supported by M-Link MU Server

RFC 0793 Transmission Control Protocol

RFC 1034 Domain names - concepts and facilities

RFC 1035 Domain names - implementation and specification

RFC 3454 Preparation of Internationalized Strings ("stringprep")

RFC 6120 Extensible Messaging and Presence Protocol (XMPP): Core

RFC 6121 Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence

RFC 6122 Extensible Messaging and Presence Protocol (XMPP): Address Format

XEP-0004 Data Forms

XEP-0030 Service Discovery

XEP-0045 Multi-User Chat

XEP-0049 Private XML Storage

XEP-0054 vcard-temp

XEP-0060 Publish-Subscribe

XEP-0082 XMPP Date and Time Profiles

XEP-0092 Software Version

XEP-0124 Bidirectional-streams Over Synchronous HTTP (BOSH)

XEP-0191 Blocking Command

XEP-0198 Stream Management
M-Link only supports Stanza Acknowledgements. Stream Resumption is not supported.

XEP-0199 XMPP Ping

XEP-0203 Delayed Delivery

XEP-0206 XMPP Over BOSH

XEP-0220 Server Dialback

XEP-0227 Portable Import/Export Format for XMPP-IM Servers
M-Link only supports importing data from XEP-0227 sources.

XEP-0258 Security Labels in XMPP

XEP-0280 Message Carbons

XEP-0289 Federated MUC for Constrained Environments

XEP-0313 Message Archive Management
M-Link currently supports versions 0.3 (MAMv0) and 0.5.1 (MAMv1) of the specification.

XEP-0334 Message Processing Hints
M-Link currently only supports the <store/> hint defined by this specification.

XEP-0346 Form Discovery and Publishing

XEP-0361 Zero Handshake Server to Server Protocol

XEP-0363 HTTP File Upload

XEP-0365 Server to Server communication over STANAG 5066 ARQ

D.4 Supported by M-Link User Server

RFC 0793 Transmission Control Protocol

RFC 1034 Domain names - concepts and facilities

RFC 1035 Domain names - implementation and specification

RFC 3454 Preparation of Internationalized Strings ("stringprep")

RFC 6120 Extensible Messaging and Presence Protocol (XMPP): Core

RFC 6121 Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence

RFC 6122 Extensible Messaging and Presence Protocol (XMPP): Address Format

XEP-0004 Data Forms

XEP-0030 Service Discovery

XEP-0045 Multi-User Chat

XEP-0049 Private XML Storage

XEP-0054 vcard-temp

XEP-0060 Publish-Subscribe

XEP-0082 XMPP Date and Time Profiles

XEP-0092 Software Version

XEP-0124 Bidirectional-streams Over Synchronous HTTP (BOSH)

XEP-0191 Blocking Command

XEP-0198 Stream Management
M-Link only supports Stanza Acknowledgements. Stream Resumption is not supported.

XEP-0199 XMPP Ping

XEP-0203 Delayed Delivery

XEP-0206 XMPP Over BOSH

XEP-0220 Server Dialback

XEP-0227 Portable Import/Export Format for XMPP-IM Servers
M-Link only supports importing data from XEP-0227 sources.

XEP-0258 Security Labels in XMPP

XEP-0280 Message Carbons

XEP-0289 Federated MUC for Constrained Environments

XEP-0313 Message Archive Management
M-Link currently supports versions 0.3 (MAMv0) and 0.5.1 (MAMv1) of the specification.

XEP-0334 Message Processing Hints
M-Link currently only supports the <store/> hint defined by this specification.

XEP-0346 Form Discovery and Publishing

XEP-0363 HTTP File Upload

Appendix E Glossary

This appendix provides a glossary of terms.

Technical Terms

Domain
See Domain Name.

Domain Name
A name within the *Domain Name System*. See [RFC1035].

Domain Name System (DNS)
A service for providing a mapping between *domain names* (for example, `example.com`) and *IP addresses*. See *RFC 1035*.

Directory Service Agent (DSA)
A server process which maintains and provides access to the Directory, a distributed database built to X.500 standards. In the context of M-Link, an LDAP Server.

Extensible Messaging and Presence Protocol (XMPP)
A collection of open standards for real-time communication, including those for *instant messaging*, presence, and *multi-user chat*. See [RFC6120].

Extensible Markup Language (XML)
A markup language used to represent structured information which is designed to be readable by both humans and machines.

Federated Multi-User Chat (FMUC)
An *instant messaging* service allowing multiple multi-user chat rooms to be connected together. See [XEP-0289].

Form Discovery and Publishing (FDP)
A service providing forms for users to look up, fill out and submit. See [XEP-0346].

Fully Qualified Domain Name (FQDN)
A complete *domain name* identifying a host system or other entity on the Internet. See Also Domain Name System (DNS).

Guard Content Exchange Protocol (GCXP)
Protocol used for communications between M-Link and M-Guard.

Hypertext Transfer Protocol (HTTP)
An application layer network protocol for distributed, collaborative, hypermedia information systems.

Icon-5066
A modem-independent STANAG 5066 server. It enables applications to work efficiently over HF Modems/Radios and allows multiple applications to work simultaneously.

Instant Messaging (IM)
Real-time text-based chatting between two or more people. XMPP IM is described in [RFC6121].

IP address
An address which identifies a host machine on an Internet network. For IPv4, it is 32-bit number commonly written in dotted number notation of the form `192.0.1.100`. For IPv6, it is a 128-bit number commonly written in a notation of the form `2001:db8::100`.

Multi-User Chat (MUC)
An *instant messaging* service allowing multiple users to chat with each other; a group chat service. See [XEP-0045].

Publish-Subscribe (PubSub)
A service, based on the Observer design pattern, where users can subscribe to receive information published to nodes (topics). See [XEP-0060].

STANAG 5066 Subnet Interface Service (SIS)
Protocol that enables an application to connect to an HF modem through a *STANAG 5066* server over TCP/IP. Also known as “Subnet Interface Sublayer”.

SIS Node Address
An HF subnetwork node address. Usually represented in a form that resembles an IPv4 address in the range `0.0.0.0` — `31.255.255.255`.

SIS Layer Extension Protocol (SLEP)
Layer protocol that can be used over *SIS* to provide a set of core services that can be used by different applications. This protocol is designed as a replacement for RCOP and UDOP.
See Also ISODE-S5066-APP3.

STANAG 5066
STANAG 5066 "Profile for High Frequency (HF) Radio Data Communication" is a NATO specification to enable applications to communicate efficiently over HF Radio.

Transmission Control Protocol (TCP)
A stream-oriented protocol for providing reliable data communications over the Internet. TCP is the primary transport protocol for *XMPP*. See *RFC 0793*.

Transport Layer Security (TLS)
A cryptographic protocol designed to provide secure communications, widely used in applications such as email and instant messenging, and most visibly used to secure HTTP transfers.

Appendix F References

The documents listed in this appendix provide references to the appropriate standards and other sources of information.

If documents can be obtained electronically, the location is stated as part of the reference. For other documents, please see Section F.6, “Obtaining documents”.

F.1 RFCs

RFC 0793
Transmission Control Protocol [https://tools.ietf.org/html/rfc0793]. J. Postel, September 1981

RFC 1034
Domain names - concepts and facilities [https://tools.ietf.org/html/rfc1034]. P. Mockapetris, November 1987

RFC 1035
Domain names - implementation and specification [https://tools.ietf.org/html/rfc1035]. P. Mockapetris, November 1987

RFC 3454
Preparation of Internationalized Strings ("stringprep") [https://tools.ietf.org/html/rfc3454]. P. Hoffman, M. Blanchet, December 2002

RFC 6120
Extensible Messaging and Presence Protocol (XMPP): Core [https://tools.ietf.org/html/rfc6120]. P. Saint-Andre, March 2011

RFC 6121
Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence [https://tools.ietf.org/html/rfc6121]. P. Saint-Andre, March 2011

RFC 6122
Extensible Messaging and Presence Protocol (XMPP): Address Format [https://tools.ietf.org/html/rfc6122]. P. Saint-Andre, March 2011

RFC 6901
JavaScript Object Notation (JSON) Pointer [https://tools.ietf.org/html/rfc6901]. P. Bryan, K. Zyp, M. Nottingham, April 2013

RFC 6902
JavaScript Object Notation (JSON) Patch [https://tools.ietf.org/html/rfc6902]. P. Bryan, M. Nottingham, April 2013

RFC 7235
Hypertext Transfer Protocol (HTTP/1.1): Authentication [https://tools.ietf.org/html/rfc7235]. R. Fielding, J. Reschke, June 2014

RFC 7578
Returning Values from Forms: multipart/form-data [https://tools.ietf.org/html/rfc7578]. L. Masinter, July 2015

RFC 8259
The JavaScript Object Notation (JSON) Data Interchange Format [https://tools.ietf.org/html/rfc8259]. T. Bray, December 2017

F.2 XMPP Extension Protocols

XEP-0004
Data Forms [https://xmpp.org/extensions/xep-0004.html]. Ryan Eatmon, Joe Hildebrand, Jeremie Miller, Thomas Muldowney, Peter Saint-Andre, July 2022

XEP-0030
Service Discovery [https://xmpp.org/extensions/xep-0030.html]. Joe Hildebrand, Peter Millard, Ryan Eatmon, Peter Saint-Andre, October 2017

XEP-0045
Multi-User Chat [https://xmpp.org/extensions/xep-0045.html]. Peter Saint-Andre, December 2022

XEP-0047
In-Band Bytestreams [https://xmpp.org/extensions/xep-0047.html]. Justin Karneges, Peter Saint-Andre, January 2021

XEP-0049
Private XML Storage [https://xmpp.org/extensions/xep-0049.html]. Peter Saint-Andre, Russell Davis, March 2004

XEP-0054
vcard-temp [https://xmpp.org/extensions/xep-0054.html]. Peter Saint-Andre, July 2008

XEP-0060
Publish-Subscribe [https://xmpp.org/extensions/xep-0060.html]. Peter Millard, Peter Saint-Andre, Ralph Meijer, March 2023

XEP-0071
XHTML-IM [https://xmpp.org/extensions/xep-0071.html]. Peter Saint-Andre, March 2018

XEP-0082
XMPP Date and Time Profiles [https://xmpp.org/extensions/xep-0082.html]. Peter Saint-Andre, Tobias Markmann, August 2021

XEP-0085
Chat State Notifications [https://xmpp.org/extensions/xep-0085.html]. Peter Saint-Andre, Dave Smith, September 2009

XEP-0091
Legacy Delayed Delivery [https://xmpp.org/extensions/xep-0091.html]. Peter Saint-Andre, May 2009

XEP-0092
Software Version [https://xmpp.org/extensions/xep-0092.html]. Peter Saint-Andre, February 2007

XEP-0095
Stream Initiation [https://xmpp.org/extensions/xep-0095.html]. Thomas Muldowney, Matthew Miller, Ryan Eatmon, November 2017

XEP-0124
Bidirectional-streams Over Synchronous HTTP (BOSH) [https://xmpp.org/extensions/xep-0124.html]. Ian Paterson, Dave Smith, Peter Saint-Andre, Jack Moffitt, Lance Stout, Winfried Tilanus, May 2021

XEP-0166
Jingle [https://xmpp.org/extensions/xep-0166.html]. Scott Ludwig, Joe Beda, Peter Saint-Andre, Robert McQueen, Sean Egan, Joe Hildebrand, September 2018

XEP-0184
Message Delivery Receipts [https://xmpp.org/extensions/xep-0184.html]. Peter Saint-Andre, Joe Hildebrand, August 2018

XEP-0191
Blocking Command [https://xmpp.org/extensions/xep-0191.html]. Peter Saint-Andre, March 2015

XEP-0198
Stream Management [https://xmpp.org/extensions/xep-0198.html]. Justin Karneges, Peter Saint-Andre, Joe Hildebrand, Fabio Forno, Dave Cridland, Matthew Wild, July 2018

XEP-0199
XMPP Ping [https://xmpp.org/extensions/xep-0199.html]. Peter Saint-Andre, March 2019

XEP-0203
Delayed Delivery [https://xmpp.org/extensions/xep-0203.html]. Peter Saint-Andre, September 2009

XEP-0206
XMPP Over BOSH [https://xmpp.org/extensions/xep-0206.html]. Ian Paterson, Peter Saint-Andre, Lance Stout, Winfried Tilanus, April 2014

XEP-0220
Server Dialback [https://xmpp.org/extensions/xep-0220.html]. Jeremie Miller, Peter Saint-Andre, Philipp Hancke, March 2015

XEP-0227
Portable Import/Export Format for XMPP-IM Servers [https://xmpp.org/extensions/xep-0227.html]. Magnus Henoch, Waqas Hussain, Matthew Wild, June 2021

XEP-0258
Security Labels in XMPP [https://xmpp.org/extensions/xep-0258.html]. Kurt Zeilenga, November 2018

XEP-0280
Message Carbons [https://xmpp.org/extensions/xep-0280.html]. Joe Hildebrand, Matthew Miller, Georg Lukas, December 2021

XEP-0289
Federated MUC for Constrained Environments [https://xmpp.org/extensions/xep-0289.html]. Kevin Smith, March 2021

XEP-0313
Message Archive Management [https://xmpp.org/extensions/xep-0313.html]. Matthew Wild, Kevin Smith, March 2023

XEP-0334
Message Processing Hints [https://xmpp.org/extensions/xep-0334.html]. Matthew Wild, April 2024

XEP-0346
Form Discovery and Publishing [https://xmpp.org/extensions/xep-0346.html]. Kevin Smith, September 2017

XEP-0361
Zero Handshake Server to Server Protocol [https://xmpp.org/extensions/xep-0361.html]. Steve Kille, September 2017

XEP-0363
HTTP File Upload [https://xmpp.org/extensions/xep-0363.html]. Daniel Gultsch, January 2022

XEP-0365
Server to Server communication over STANAG 5066 ARQ [https://xmpp.org/extensions/xep-0365.html]. Steve Kille, March 2022

F.3 W3C Recommendations

XSLT 1.0
XSL Transformations (XSLT) Version 1.0 [https://www.w3.org/TR/xslt-10/]. W3C Recommendation, 16 November 1999

F.4 Other Publications

JSON Schema [https://json-schema.org/].

Open XML SPIF [http://www.xmlspif.org/].

F.5 Isode White Papers

ISODE-MSG-FLOW
Security Policy Based Access Controls [https://www.isode.com/products/security-policy-infrastructure.html].

ISODE-S5066
STANAG 5066 (The Standard for Data Applications over HF Radio) [https://www.isode.com/whitepapers/stanag-5066.html].

ISODE-S5066-APP3
SIS Layer Extension Protocol (SLEP) (S5066-APP3) [https://www.isode.com/whitepapers/S5066-APP3.html].

ISODE-XMPP-LABELS
Using Security Labels to Control Message Flow in XMPP Services [https://www.isode.com/whitepapers/controlling-message-flow.html].

F.6 Obtaining documents

F.6.1 ISO/IEC documents

ISO/IEC standards and draft documents may be obtained from:

ISO Central Secretariat
International Organization for Standardization (ISO)
1, rue de Varembé
Case postale 56
CH-1211 Geneva 20
Switzerland

Telephone: +41 22 749 01 11

Fax: +41 22 733 34 30

Web: http://www.iso.org/

F.6.2 ITU-T (CCITT) documents

International Telecommunications Union
Place des Nations
CH-1211 Geneva 20
Switzerland

Telephone: +41 22 730 61 41 Fax: +41 22 730 51 94

Email: sales@itu.int

Web: http://www.itu.int/

F.6.3 RFCs

Electronic copies of RFCs are available from the following servers:

- http://ftp.isi.edu/in-notes/
- http://www.rfc-editor.org/

Appendix G Message Archive Format

This chapter provides an example of the format of an import or export of the message archive.

XML Comments are included in the below example as documentation of the format. These are not generated by M-Link during export and will be ignored during import. An ellipsis (...) is sometimes used to elide uninteresting content of stanzas.

```
<!-- v1.1 --->
<archives xmlns="http://isode.com/xmpp/archiving" name="example"
          start="2013-07-21T02:58:15.000Z"
          end="2014-07-21T02:58:15.000Z">
  <!-- Archives might also have a domain attribute
       if this file was generated as an export operation
       for a single domain --->
  <users>
    <user jid="localuser1@example.com">
      <!-- all archived stanzas in order,
           one per item element --->
      <item uid="a" timestamp="2014-07-21T02:56:15.000Z"
            order="0">
        <!-- The stanza is embedded here unmodified other than
             injecting the correct namespace --->
        <message to="localuser1@example.com/etuhet"
                 from="remoteuser1@example.org" type="normal"
                 id="maybe" xmlns="jabber:client" lang="en">
          <!-- stanza payloads remain here-->
          <body>Hi Kev</body>
          <xhtml-im xmlns="..."><b>
            Hi user
          </b></xhtml-im>
        </message>
      </item>

      <item uid="b" timestamp="2014-07-21T02:58:15.000Z">
        <message from="localuser1@example.com/etuhet"
                 to="remoteuser1@example.org" type="normal"
                 id="maybe2" xmlns="jabber:client">
          <body>Hi back</body>
        </message>
      </item>
    </user>
  </users>
  <pubsub>
    <!-- Publish-Subscribe history is split per service (including
         PEP services) and per node --->
    <service jid="localuser1@example.com">
      <node node="http://jabber.org/protocol/geoloc">
        <!-- Node creations are logged with the user responsible
             and timestamp --->
        <item publisher="localuser1@example.com/bou.adg2d98213"
              timestamp="1979-07-21T02:55:15.000Z" uid="b"
              order="0">
          <create></create>
        </item>

        <!-- Item publication wraps the payload in an
             <item><publish>... format --->
        <item publisher="localuser1@example.com/bou.adg2d98213"
              id="uteutudaos"
              timestamp="1979-07-21T02:56:15.000Z"
              uid="c" order="0">
          <publish>
            <!--payload here-->
            <geoloc
```

```
                xmlns='http://jabber.org/protocol/geoloc'
                xml:lang='en'>
              <accuracy>20</accuracy>
              <country>Italy</country>
              <lat>45.44</lat>
              <locality>Venice</locality>
              <lon>12.33</lon>
            </geoloc>
          </publish>
        </item>

        <!-- Node deletions are logged with the user responsible
             and timestamp --->
        <item publisher="localuser1@example.com/bou.adg2d98213"
              timestamp="1979-07-21T02:57:15.000Z" uid="d"
              order="0">
          <destroy></destroy>
        </item>
      </node>
    </service>
    <service jid="pubsub.isode.com">
      <!-- Where an item is published and replaced by another with
           the same pubsub id, both publishes are archived --->
      <node node="notes/for/user2">
        <item publisher="localuser2@example.com/etuhoe"
              id="euth.tu" timestamp="1979-07-21T02:56:15.000Z">
          <publish>
            <note>this is note 1</note>
          </publish>
        </item>

        <item publisher="localuser3@example.com/Psi" uid="e"
              id="euth.tu" timestamp="1989-07-21T02:57:15.000Z">
          <publish>
            <note>this is note 2</note>
          </publish>
        </item>
      </node>
    </service>
  </pubsub>
  <mucs>
    <!-- MUC history is stored sequentially per room --->
    <room jid="room1@talk.example.com">
      <item uid="f" publisher="localuser1@example.com/etuhet"
            timestamp="2014-07-21T02:56:15.000Z" order="0"
            nick="User 1">
        <message from="localuser1@example.com/etuhet"
                 to="room1@talk.example.com" type="groupchat"
                 id="maybe" xmlns="jabber:client">
          <!-- payloads here-->
          <body>Hi Everyone</body>
        </message>
      </item>

      <item uid="g" publisher="localuser1@example.com/etuhet"
            timestamp="2014-07-21T02:56:15.000Z" nick="User 1">
        <presence></presence>
      </item>

      <item uid="h" publisher="localuser5@example.com/oued"
            timestamp="2014-07-21T02:56:16.000Z" nick="User5">
        <subject>Changing the subject</subject>
      </item>

      <item uid="i" publisher="localuser5@example.com/oued"
            timestamp="2014-07-21T02:56:17.000Z" nick="User5" >
        <nick>Old Nick</nick>
```

```
      </item>

      <item uid="j" publisher="localuser5@example.com/oued"
            timestamp="2014-07-21T02:56:16.000Z" nick="User5">
        <!-- the new config is stored when it changes --->
        <config>...</config>
      </item>

      <item uid="k" publisher="localuser5@example.com/oued"
            timestamp="2014-07-21T02:56:16.000Z" nick="User5">
        <kicked>Go away</kicked>
      </item>

      <item uid="l" publisher="localuser5@example.com/oued"
            timestamp="2014-07-21T02:56:16.000Z" nick="User5">
        <leave>I'm going home</leave>
      </item>

      <item uid="m" publisher="localuser5@example.com/oued"
            timestamp="2014-07-21T02:56:16.000Z" nick="User5">
        <join>status</join>
      </item>

      <item uid="n" publisher="localuser5@example.com/oued"
            timestamp="2014-07-21T02:56:16.000Z" nick="User5">
        <destroy>reason</destroy>
      </item>

      <item uid="o" publisher="localuser5@example.com/oued"
            timestamp="2014-07-21T02:56:16.000Z" nick="User5">
        <create>reason</create>
      </item>
    </room>
  </mucs>
</archives>
```

Appendix H Customizing Archive PDF/A Files

This appendix discusses customization of the Archive PDF/A files.

Archive PDF/A files generated using Archive Admin Tool for exporting archives can be customized using the property file. A sample property file is installed in *(AATSHAREDIR)*. A Local copy in *(AATETCDIR)* can override the file in *(AATSHAREDIR)*.

H.1 Property File

The file *export.props* in *(AATETCDIR)* can be used to customise PDF/A files generated from exported archive data see(Section A.2.2, “Export parameters”)

H.2 Customisation

H.2.1 Metadata

The PDF metadata is used to provide additional information about the PDF file. This data can be modified using the properties for the keys starting with *META_* in the corresponding property file. The properties allow you to provide the title, author, subject and keywords metadata for the PDF.

As an example the following line in the property file sets the title metadata to “XMPP Archives.”

```
META_TITLE=XMPP Archives
```

H.2.2 Cover Page

It is also possible to customise the cover page and security label on the header and footer of pages on the export and search query PDFs. Note that this customisation is not supported for the statistics graph PDF.

In order to modify the image logo on the cover page, provide the absolute path of the image file for the *IMAGE_LOGO* key using the format as specified in the example below.

Linux example:

```
IMAGE_LOGO=file:////home/user/logo.svg
```

Windows example:

```
IMAGE_LOGO=file:///C:/Isode/etc/image/logo.svg
```

The value of the *TITLE* key provides the title on the cover page.

The label that appears on the header and footer is derived from the value of the *LABEL* key. This should be left blank if no label is to be displayed on the PDF.

The cover page displays a table that lists information about the archive. If required, more rows can be added to this table using the property file. All properties starting with *TABLE_* will appear as additional rows on the cover page table.

The following is a sample property to add a row with columns “Point Of Contact” as “Henry Brown” to the table.

```
TABLE_Point Of Contact=Henry Brown
```

Appendix I Upgrading R17.0 Configurations

This appendix discusses upgrading existing R17.0 M-Link configurations so that they can be used by the current M-Link release. It is important that this appendix is read and understood in its entirety before starting an upgrade, and that the steps are followed in order. Particular care must be paid to ensuring that you have backups or snapshots allowing you to restore your R17.0 system to its previous state if you abort the upgrade.

An R17.0 M-Link configuration consists of either one or two configuration files (normally *`(ETCDIR17)`mlink.conf* and *`(ETCDIR17)`mlink.conf.node*), a directory containing user database files and a directory containing Publish-Subscribe database files. Additional files (e.g. TLS certificates, Key Pair files, Security Policies etc) may be referenced from the configuration. In addition, passwords within the configuration file may have been encrypted using the Isode ServPass facility; if so there will be an associated configuration file in the *`(ETCDIR17)`servpass* directory.

Note: R17.0 M-Link installations which include IRC Gateway support cannot yet be upgraded.

I.1 Procedure for Upgrading from R17.0

- Stop your R17.0 M-Link server.
- You must securely back up your R17.0 configuration. You will need to make sure you have saved the contents of *`(ETCDIR17)`* and any subdirectories, any TLS certificate or key-pair files, security policy files, clearance or label catalogs and individual label or clearance files which your configuration references, plus your User and Publish-Subscribe database directories. More information about backing up your configuration can be found in the R17.0 M-Link Administration Manual.
- The current M-Link release uses ports 5001, 5221 and 5224 by default for services which were not part of M-Link R17.0. This means that upgrade of an R17.0 configuration which used these ports (e.g. as listening addresses for Links) will fail with an error which reports an address-port pair conflict. If you have configured these ports in your R17.0 config, you should first deconfigure the use of the ports; after upgrade is complete you can then reconfigure use of the ports. Additionally as 5221 is the port for the administrative interface, this must be accessible through any firewalls for the system administrator.
- R17.0 allowed listen addresses to be set by hostname rather than IP in some situations. This is not supported by the current release; please replace any such cases with IP addresses before attempting the upgrade.
- On Windows, run the "Uninstall M-Link services" tool.
- Remove your R17.0 M-Link installation (this will leave your data in place).
- If you have the R17.0 version of the Isode M-Vault Directory Server installed on the same system as your R17.0 M-Link, you will need to upgrade this at the same time. You should install the most recent available version of the Isode M-Vault packages that is supported by your system and follow the upgrade instructions in the associated Isode Release Notes. Note that M-Link 19.4 is decoupled from M-Vault, so that you do not need to keep the versions of the two products synchronised.

  If your R17.0 system is configured so that your M-Link Archive Server runs on a separate system, refer to Section I.5, “Non-local R17.0 Archiving”.
- Ensure that your system is running an operating system version which is supported by the new version of M-Link and is fully patched with any required updates.

- Install the new M-Link version. This will automatically start the M-Link service. Stop the M-Link services before performing the upgrade.
- If you are using a Linux operating system, modify the ownership of your *(VARDIR)* directory and *(ETCDIR17)servpass/xmppd.pass* file to the mlink runtime user:

```
chown -R mlink /var/isode/mlink
```

```
chown mlink /etc/isode/servpass/xmppd.pass
```

If your User and Publish-Subscribe database directories are held under a different file path, you will need change the ownership of that in the same way, e.g.:

```
chown -R mlink /var/isode/ms
```

- Run the Upgrade tool, using the command line options described in Section I.2, “Running the Upgrade Tool” if needed. The command lines:

```
(SBINDIR)mlink_upgrade_17_to_19
```

for a dry-run, which checks that an upgrade can be performed, and:

```
(SBINDIR)mlink_upgrade_17_to_19 --commit
```

to cause output of a new mlink.xml and associated on-disk changes should be all that is required to upgrade a standard installation.

Note: You should always perform a dry-run and check the output before attempting an upgrade.

Note: If you are upgrading a Windows configuration which was installed using a non-default path, it will be necessary to use the `--old-etc-path` (and `--old-config-file` if not the default) switches when running the upgrade.

Note: You must run the configuration upgrade tool as the userid as which the new M-Link will run (mlink), and you must ensure this userid has read-write access to both the R17.0 configuration and database locations and the new configuration location (*(VARDIR)*), as in the above Linux chown steps.

- Restart the M-Link service and access it via Chapter 8, *M-Link Configuration and Management*. You will be prompted to create an initial administration user, and then to activate the product.
- If your activation key includes clustering support, activation of your M-Link Server will result in you being asked whether you want to create a new configuration or join an existing cluster. Selecting "Create a new config" will allow you to access your upgraded R17.0 configuration. If you are upgrading a clustered R17.0 configuration, see Section I.4, “Upgrading a Clustered R17.0 Configuration”.
- If the R17.0 configuration includes Security Labelling configuration, refer to Section I.6, “Security Labelling”.

Note: Some behaviours have changed slightly between R17.0 and the current release. Once the upgrade process is complete, you should go through the resulting configuration in M-Link Administration Interface and confirm that the configuration is as you want it.

Note: An upgrade may result in the renaming of filenames or file path components within the User and Publish-Subscribe database, due to changes in file path encoding rules. As a result, User and Publish-Subscribe databases which have been upgraded cannot be used if the configuration is rolled back to use an R17.0 M-Link, and instead the R17.0 system must be restored from backup.

Note: The upgrade procedure does not import the R17.0 logging configuration (held in *(ETCDIR17)/mlinklogging.xml*). If you have a customised logging configuration, you will need to recreate this using M-Link Administration Interface once the upgrade is complete.

Note: LDAP configurations in a clustered R17.0 setup may be node-specific (i.e. allowing each node of the cluster to access a different LDAP server). LDAP configurations are shared across cluster nodes in the current version. If your R17.0 configuration was set up in this way please contact isode.support@isode.com for advice.

I.2 Running the Upgrade Tool

The configuration upgrade tool *(SBINDIR)mlink_upgrade_17_to_19* can be used to upgrade an R17.0 configuration to one which can be used by the current M-Link. The tool takes the following command line options:

--help
: Display help.

--commit
: Commit the new configuration to disk. By default, the tool will only check whether the upgrade can be performed (i.e. a dry-run), and report the changes it will make.

--old-config-file
: The path to the M-Link R17.0 configuration file. This defaults to *(ETCDIR)mlink.conf*.

--old-etc-path
: The path to the R17.0 directory beneath which the ServPass configuration directory is located. This defaults to *(ETCDIR)*.

--verbose
: Enable verbose reporting.

--version
: Display the exact M-Link version string.

I.2.1 Example Upgrade Output

Example I.1. An example of running the upgrade tool against a sample *mlink.conf* file

```
tc@dystopia:/opt/isode/sbin/mlink_upgrade_17_to_19 --verbose
Info: Upgrading configuration file /etc/isode/mlink.conf
Info: Old configuration was generated by version 17.0v24-1
Reading from file sio_policy.xml
Info: Loaded Server default Security Policy 4ea15f31-dbd3-4a28-99b0-2d1a004cb6df
Info: Stored keyPair with subject /O=Isode/CN=Mlink as id d348bc5d-e536-461c-908b-d9e55c852e4b
Info: Importing user admin@example.com from XMLDB file xmldb.xml for domain example.com
Ignoring unknown xmldb element maillocaladdress
Ignoring unknown xmldb element mail
Info: Importing user user1@example.com from XMLDB file xmldb.xml for domain example.com
Ignoring unknown xmldb element maillocaladdress
Ignoring unknown xmldb element mail
Info: Loaded clearance  into store cef8b184-ee9f-45f8-a0df-aa937b839b7b
Info: Importing user user2@example.com from XMLDB file xmldb.xml for domain example.com
Ignoring unknown xmldb element maillocaladdress
Ignoring unknown xmldb element mail
Info: Importing user user3@example.com from XMLDB file xmldb.xml for domain example.com
Ignoring unknown xmldb element maillocaladdress
Ignoring unknown xmldb element mail
Info: Importing user user4@example.com from XMLDB file xmldb.xml for domain example.com
Ignoring unknown xmldb element maillocaladdress
Ignoring unknown xmldb element mail
Info: MUC domain muc.example.com inherits its TLS key pair from parent domain example.com
Info: PubSub domain pubsub.example.com inherits its TLS key pair from parent domain example.com
Info: PubSub domain forms.example.com inherits its TLS key pair from parent domain example.com
Set cipherSuites to TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
Info: Migrating path /var/isode/ms/user
Info: Migrating path /etc/isode/ms/pubsub
```

I.3 Unsupported R17.0 Features

A configuration upgrade will fail if the R17.0 configuration includes any features which are not supported by M-Link. Some unsupported features are planned to be supported in future versions, and a description of these can be found at https://www.isode.com/m-link-19-4-limited-release/.

If your R17.0 configuration includes features which are not yet supported (see also Appendix J, *Options not carried over from R17.0 configurations*), you will either need to deconfigure these, or delay upgrading until a version of M-Link which supports them is available. You should also contact isode.support@isode.com for advice.

The Upgrade process reports two different types of unsupported feature:

Not Yet Supported
: This is a feature that M-Link will support in a later release of R19.4.

No Longer Supported
: This is an end-of-life feature that has been removed from M-Link.

I.3.1 Warning and Information Reported During Upgrade Process

Warning and Information reports may be emitted during the upgrade process. Detailed reports may also be emitted if you run the update process with the --verbose flag, which is not necessary unless requested by Isode Support.

Warnings generally report minor errors found during the upgrade process or highlight the fact that additional manual configuration of the upgraded configuration will be needed. Warnings are also emitted during a "dry run" if filesystem operations fail. The R17.0 configuration of Server Administrators (which will always be configured in

R17.0) is also reported as a warning; the current release uses a different mechanism for administrator configuration.

Information reports simply report progress on configuration upgrade actions.

I.4 Upgrading a Clustered R17.0 Configuration

The procedure for upgrading a clustered R17.0 M-Link configuration is:

- Upgrade each node of the cluster, following the procedure described in Section I.1, “Procedure for Upgrading from R17.0”.
- On one node of the cluster choose the "Create a New Configuration" after product activation using Chapter 8, *M-Link Configuration and Management*. On each of the other nodes, activate the server using Chapter 8, *M-Link Configuration and Management* in the same way, but then choose the "Join an Existing Cluster" option and follow the instructions to join to the first node which was upgraded.

Note: M-Link R17.0 allowed almost any element of a clustered configuration to be given a node-specific setting. The current release restricts node-specific configuration to a more limited set of options. You should inspect a clustered configuration after upgrade to ensure that node-specific settings are set correctly.

I.5 Non-local R17.0 Archiving

Using R17.0 it was possible to configure the M-Link Archive Server to run on a separate system from the M-Link. This is not possible with the current release, and an attempt to upgrade a configuration which is set up in this way will fail.

To upgrade such a configuration, you should do the following before attempting the upgrade:

- Move or copy your archive data onto the system on which your upgraded M-Link will run.
- Ensure that your R17.0 system has its archive data directory set to the location into which you have moved the archive data.
- Ensure that your R17.0 system is configured to use a local M-Link Archive Server.

I.6 Security Labelling

Some changes in Security Labelling behavior between R17.0 and the current release mean that you may need to make manual configuration changes after upgrade:

- Support for STANAG 4774 labels was experimental in R17.0. An upgrade of a configuration which uses these labels cannot be upgraded automatically. You should deconfigure use of these labels before attempting an upgrade.

- In R17.0, if a user with an associated clearance enters a Multi-User Chat, Instant Messaging or Publish-Subscribe domain which has been configured without a clearance, the user is presented with a catalog of labels which can be used. In the current release, the catalog of available labels presented to a user depends on the combination of the user's clearance and the domain's clearance; so that no labels will be available for a domain which does not have a clearance.

  This means that you will need to configure a clearance for any domain where a label catalog is required. This can either be done explicitly for the domain or via the Server Default Clearance option.
- Peer Controls do not inherit a default Security Clearance in the current release. This means that if there is no Clearance explicitly configured for a Peer in your R17.0 configuration, you will need to use M-Link Administration Interface to configure a suitable Security Clearance for the Peer before labelled messages can be accepted (subject label/clearance permissions).

I.7 TLS Key Size

In R17.0 the default TLS security level was set to TLS v1.0. This meant that TLS key pairs which used a 1024-bit RSA key could be used. In the current release 1024-bit RSA keys cannot be used for non-XMPP communication (e.g. Child Service connections used for interprocess communication), and so any key pairs should be regenerated with at least a 2048-bit RSA key.

The current release supports configuration of the TLS security level, but this is only used for XMPP interactions.

I.8 Connections and Links

R17.0 Peers can specify host/port pairs which override the use of DNS when making S2S connections. These map to elements in the current release's Peer Connections list. Alternatively, the Peer can specify a list of links which will be used (providing X2X or S5066 connections). If neither are specified, the normal DNS-based S2S connection will be used.

R17.0 Links map to Link configurations in the current release. Only R17.0 links of type Direct I/O (X2X) and S5066 are supported, and S5066 is replaced by SLEP for the current release. S5066 Server configurations will be created as needed and referenced from Links which are of type SLEP, but these will require inspection and potential modification to work correctly. Please contact isode.support@isode.com for advice.

BOSH configuration options have changed in R19.4, and will need reconfiguration after upgrade. Most significantly, the path component of the BOSH URL is now fixed to `/http-bind/`.

I.9 Folding and Filtering

R17.0 supported filtering and folding of Message and Presence stanzas. For both types of stanza, lists of XML elements to keep and strip can be configured; for Messages it is also possible to configure a list of required XML elements. An empty stanza after stripping or one which does not contain the required elements is rejected.

An upgrade to the current release converts these element lists into XSLT transforms and associated configuration and associates them with the Peer.

For domain, IQ filtering is supported in the same way.

I.10 JID Filtering

R17.0 JID filtering is mapped into an equivalent configuration in the current release.

I.11 Authorization

R17.0 authorization settings are mapped directly across into equivalent R19.4 settings, with the exception of support for XEP-0078 Non-SASL Authentication, which has been dropped.

Appendix J Options not carried over from R17.0 configurations

This appendix discusses the options of R17.0 configurations that are not upgraded in this version of M-Link, either because the feature is end of life, or because the feature is coming in a future version.

J.1 Clustering Options

Upgrades of clustered systems need to follow the instructions in Section I.4, “Upgrading a Clustered R17.0 Configuration”. If any of the following options are set, you will be asked to perform the required additional steps.

Table J.1. Dropped server tuning options

Name	Description
cell_xmpp_uri	XMPP node URI for clustering
cell_xmpp	XMPP peer node URI for clustering
cell_wabac_uri	Archive server node URI for clustering
cell_wabac	Archive server peer node URI for clustering
cell_tls	Require TLS in clustering

J.2 Other TLS Options

The following TLS-related options are not supported in the current version:

Table J.2. Dropped server tuning options

Name	Description
fips140_mode	FIPS-140 mode
tls_self_certs	Self certificates
tls_compression	TLS compression
tls_user_ca_certs	User certificate trust anchors
tls_user_check_crls	Check user certificate revocation
tls_user_check_leaf	Check user end-entity certificate revocation
tls_user_domain	Alternative domain for user certificates
tls_user_match_email	Match user to SAN email address
tls_user_match_jid	Match user to SAN XMPPAddress
tls_user_match_upn	Match user to SAN User Principal Name
tls_user_match_subject_cn	Match user to Subject Common Name
tls_user_match_subject_email	Match user to Subject Email
tls_host_simple_wildcards	Allow simple wildcards in matching

J.3 IRC Gateway

R17.0 is the latest release of the M-Link IRC Gateway product. The following related options are not supported in the current release:

Table J.3. IRC Gateway Options

Name	Description
irc	Enable IRC gateway
irc_kick_on_error	Propagate IRC join error to MUC
irc_prefix_size	Maximum IRC prefix length
irc_outgoing_nick_prefix	IRC outgoing nickname prefix

J.4 XMPP over Direct TLS for C2S Options

XMPP over Direct TLS (XMPPS) is not supported in the current release. The following options are not supported:

Table J.4. Dropped server tuning options

Name	Description
enable_xmpps	Enable XMPPS
xmpps_client_host	XMPPS C2S listener host
xmpps_client_port	XMPPS C2S listener port

J.5 Server Tuning Options

The following server tuning options have been removed from the current release. They are no longer relevant to the current release. These options are not expected to return in future releases.

Table J.5. Dropped server tuning options

Name	Description
cache_tmpdir	Temporary cache directory
run_dir	Runtime directory
stack_tracing	Enable stack tracing
thread_pool_min	Minimum number of worker threads
thread_pool_max	Maximum number of worker threads
use_mmap	User memory maps
max_net_buffer	Maximum network buffer
resolver	The DNS resolver to use
service_group	Windows service group name
xmpp_c2s_ack_request_period	C2S acknowledgement request period

Name	Description
xmpp_x2x_transfer_pending	Transfer pending X2X stanzas
xmpp_ping	Ping time
xmpp_ping_timeout	Ping timeout
xmpp_probe_timeout	Probe timeout
xmpp_reprobe_time	Reprobe timeout
xmpp_probe_cache	Return probe from local cache
xmpp_probe_suppress_dup	Suppress duplicate presence
max_decompression_size	Maximum decompression buffer size
max_auth_failures	Maximum number of auth failures
last_login_fail_max	Maximum auth fail reports
xmpp_session_limit	Per-user session limit
nonexists_cache_period	Non-exists cache period
offline_message_size	Offline message size
stats_dir	Queue statistics directory
xmpp_disco_cache	Cache XMPP service discovery information
xmpp_max_muc_history_limit	XMPP maximum MUC history limit
pubsub_flush_delay	Delay PubSub flush
tcs_compat	Enable TCS compatibility mode
send_multiple_muc_status_codes	Send multiple MUC status codes
authdb_ldap_timestamp_window	Authdb timestamp window

J.6 Other unsupported options

The following options are not supported in the current release. Unless otherwise stated, these options are not expected to return in future versions.

Table J.6. Server-wide Options

Name	Description	Support
user_hash	User database name hashing algorithm	M-Link R19 only supports the default hashing algorithm.
agentx_slave	Act as AgentX slave for SNMP	Support for AgentX has been removed from M-Link R19
agentx_socket	AgentX socket	Support for AgentX has been removed from M-Link R19
domain	Primary domain	M-Link R19 no longer supports the concept of a primary domain.
ms_user	Runtime user id	The M-Link run-time user is governed by the service description
bosh_tls	Enable BOSH TLS	TLS for BOSH is always enabled, if available, by

Name	Description	Support
		the upgrade; it can later be disabled if desired
bosh_files	Sets the BOSH files directory	Additional files, such as the Archive web interface, are always served from a fixed location
server_admins	Server administrators	Not yet supported. A single Server Administrator will be available.
xmpp_motd	XMPP Message of the Day	This feature is End of Life
last_login	Enable Last Login reporting	This feature is End of Life
autoaccept_local_default	Auto-accept presence subscription request defaults	This feature is End of Life
autosubscribe_local_default	Auto-subscribe when receiving a subscription request	This feature is End of Life
xmpp_process_roster _groups	Enable XMPP roster groups	M-Link R19 always enables XMPP roster groups when configured
groups_discovery	Groups discovery	Groups are managed through M-Link Administration Interface in R19
xmpp_muc_domain	XMPP default MUC domain	M-Link R19 no longer requires the concept of a default MUC domain
stats_domain	Statistics domain	This feature is End of Life
muc_archive_dir	MUC audit archive directory	File-based archiving has been replaced with Message Archive Management
user_archive_dir	User audit archive directory	File-based archiving has been replaced with Message Archive Management
audit	Enable audit module	Auditing is always enabled
amp_http_host	AMP HTTP host	This feature is End of Life
amp_https_port	AMP HTTP port	This feature is End of Life
amp_http_tls	Enable HTTP API TLS	This feature is End of Life
amp_http_files	HTTP API support files path	This feature is End of Life
amp_pubsub_presence	Enable non-PEP XEP-0060 presence processing	This feature is End of Life
amp_monitor_muc	Enable monitor MUC domain	This feature is End of Life
amp_iq_delegation_timeout	Timeout for delgated user IQs	This feature is End of Life
wabac_host	Archive server host	Will cause an error if set: see Section I.5, “Non-local R17.0 Archiving”
wabac_port	Archive server port	No longer required for operation

Name	Description	Support
wabac_config_share	XMPP server shares archive server config	No longer required for operation
carbons	Enable XEP-0280 message carbons	Carbons support is always available in the current release
schematron	Schematron rules	This feature is replaced by Stanza Transformations
cdcie_ccp	CDCIE CCP enabled	This feature is End of Life
flot	Inject FLOT for labels	FLOT configuration is reset by the upgrade; please configure after completing the upgrade
flot_default	Inject FLOT for default labels	FLOT configuration is reset by the upgrade; please configure after completing the upgrade
sio_deny_unrecognized _cdcie_ccp_labels	Deny unrecognized CDCIE CCP labels	This feature is End of Life
authdb_refresh	Authdb refresh interval	The refresh interval is not carried over to the current release

Table J.7. Peer Options

Name	Description	Support
accept (address)	Accept address	This feature is End of Life
sio_relabel_out	Enable SIO Relabel Out	SIO relabelling configuration is reset by the upgrade; please configure after completing the upgrade.

Table J.8. Common Domain Options

Name	Description	Support
domain_admins	Domain administrators	Not yet supported

Table J.9. MUC Domain Specific Options

Name	Description	Support
sio_default_clearance	Default clearance	This option had no effect on MUC domains in R17

Table J.10. PubSub Domain Specific Options

Name	Description	Support
identity_type	Supported identity type	Enables FDP support, but only if type is `urn:xmpp:fdp:0`. All other identities will be ignored.
sio_default_clearance	Default clearance	This option had no effect on PubSub domains in R17

Appendix K Summary of R17.0 Option Mapping

This appendix discusses the mapping of R17.0 configuration options to the current release of M-Link.

The tables below show which R17.0 configuration options are mapped into equivalent settings in the current version.

Table K.1. Server-wide Options

Name	Description	Support
userdir	Absolute or relative path for user database.	Mapped to userDir, including mailstore_dir prefix if appropriate
telemetry_log	Telemetry log directory	Mapped to Telemetry Path
telemetry_auto_create	Automatically create user telemetry directories	Mapped to Automatically Create Telemetry Directories
cluster_timeout	Sets the cluster timeout	Mapped to Cluster Timeout
dcons_port	Debug console port	Sets Debugging Console Port
xmpp_server_host	XMPP S2S listener host	Sets the Server Connection Listening Address, which must be an IP address for a successful upgrade
xmpp_server_port	XMPP S2S listener post	Sets the Server Connection Port
xmpp_client_host	XMPP C2S listener host	Sets the Client Connection Listening Address, which must be an IP address for a successful upgrade
xmpp_client_port	XMPP C2S listener port	Sets the Client Connection Port
bosh_uri	BOSH URI	Used to set the BOSH Connection Listening Address and BOSH Connection Port
bosh_host	BOSH listener host	Sets the BOSH Connection Listening Address, which must be an IP address for a successful upgrade
bosh_port	BOSH listener port	Sets the BOSH Connection Port
xmpp_s2s_startup_timeout	S2s startup timeout	Sets the Remote Session Authentication Timeout
xmpp_s2s_timeout	S2S timeout	Sets the Remote Session Connection Timeout
xmpp_whing	Whing timer	Link Whitespace Default Ping Time
xmpp_db_secret	Dialback secret	Sets Dialback Secret
xep_138	Enable XEP-0138 compression	Sets "Enable Compression for X2X Links by Default"

Name	Description	Support
tls_cipher_list	TLS cipher list	Sets TLS 1.3 cipher suites
tls_security_level	TLS security level	Sets minimum TLS version supported
tls_ca_certs	Trust anchors	Sets default trust anchors
tls_ca_file	Trust anchors file	Sets default trust anchors
tls_cert	TLS certificate	Sets certificate in default TLS identity
tls_cert_file	TLS certificate file	Sets certificate in default TLS identity
tls_key	TLS key	Sets private key in default TLS identity
tls_key_file	TLS key file	Sets private key in default TLS identity
tls_key_password	TLS key password	Sets password in default TLS identity
require_tls	Require TLS	Sets Require TLS by Default for IM Domains
tls_verify_timeout	Verify timeout	Sets TLS Certificate Verification Timeout
tls_verify_depth	Verify depth	Sets TLS Verify Depth
tls_ldap_server	LDAP host for CRL lookup	Sets LDAP Directory hostname for CRL Retrieval
tls_ldap_port	LDAP port for CRL lookup	Sets LDAP Directory port for CRL Retrieval
tls_check_crls	Check revocation	Sets Check Certificate Revocation Check Type to "Check All Certificates in Chain"
tls_check_leaf	Check revocation on end-entity certificates	Sets Check Certificate Revocation Check Type to "Only Check End-Entity Certificates"
tls_ocsp_nonce	Use nonce in OCSP	Sets Use OCSP nonce
tls_ocsp_uri	URI for OCSP	Sets OCSP query URI
tls_ocsp_responder_cert	OCSP responder certificate	Sets OCSP Responder Certificate
tls_ocsp_responder	OCSP responder certificate file	Sets OCSP Responder Certificate
tls_lookup_avoid_ocsp_configured	Don't use configured OCSP URI	Sets Enable OCSP URI to false
tls_lookup_avoid_ocsp_uri	Don't use OCSP URIs from certificate extensions	Sets Don't use OCSP URIs from certificate extensions
tls_lookup_avoid_crl_configured	Don't get CRLs from configured LDAP server	Sets Don't get CRLs from configured LDAP server
tls_lookup_avoid_cert_configured	Don't get certs from configured LDAP server	Sets Don't get certs from configured LDAP server
tls_lookup_avoid_crl_uri	Don't use URIs from extensions to lookup CRLs	Sets Don't use URIs from extensions to look up CRLs

Name	Description	Support
tls_lookup_avoid_cert_uri	Don't use URIs from extensions to lookup certs	Sets Don't use URIs from extensions to look up certificates
tls_lookup_avoid _freshestcrl	Ignore freshestCRL extensions	Sets Ignore freshestCRL extensions
tls_lookup_avoid_ocsp _httpget	Always use HTTP POST for OCSP requests	Sets Always use HTTP POST for OCSP requests
xep55_search_defaults	XEP-0055 search defaults	Sets Default User Search Scope
offline_messaging_default	Offline messaging default	If false, sets User Offline Messages Max to zero
offline_messaging	Offline messaging	If false, sets User Offline Messages Max to zero
offline_message_max	Offline message limit	Sets User Offline Messages Max
pubsub_dir	Publish-Subscribe directory	Sets Pubsub Path
userdb_timeout	Timeout for userdb sync	Sets User Database Timeout
fmuc	Enable XEP-0289 FMUC	Sets Enable FMUC
fmuc_rejoin_frequency	FMUC rejoin frequency	Sets FMUC Connection Retry Timer
wabac_data_dir	Archive database directory	Sets Archive Path
wabac_queue_dir	Archive queue directory	Sets Archive Queue Directory
wabac_http_host	Archive HTTP server host	Sets Archive Server Listening Address
wabac_http_port	Archive HTTP Server Port	Sets Archiving API Port
wabac_http_tls	Enable archive HTTP TLS	Sets Use HTTPS for the API
wabac_timeout	Timeout for WABAC operations	Sets Archiving Queue Timeout
wabac_journal_mode	Archive database journal mode	Sets Database Journal Mode
wabac_include_remote_ mucs_in_user_results	Include remote MUCs in archive user results	Sets Include Remote MUCs in User Results
wabac_sqlite_heap_limit	Archive server database memory limit	Sets Database Heap Limit
max_mam_items	Maximum number of MAM results per request	Sets Maximum Page Size
authdb_excluded_ldap _attribute	Authdb excluded attributes	Applies to all added LDAP configurations

Table K.2. Peer Options

Name	Description	Support
domain	Peer domain name	Sets Peer Domain Name
connect	Connection information	See Section I.8, “Connections and Links”
deny	Deny	Sets Peer Deny
require_tls	Require TLS	Sets Peer TLS to Required
require_tls_auth	Require authenticated TLS	Sets Peer TLS to Required With Auth

Name	Description	Support
require_strong_auth	Require strong authentication	Sets Peer TLS to Required With Auth
tls_cert	TLS certificate	Sets Pinned Peer TLS Certificate
tls_cert_file	TLS certificate	Sets Pinned Peer TLS Certificate
relay (zone)	Relay zone	Sets Peer Relay Zone
message_fold	Message folding and filtering	See Section I.9, “Folding and Filtering”
presence_fold	Presence folding and filtering	See Section I.9, “Folding and Filtering”
jid_filter	JID filtering	See Section I.10, “JID Filtering”
raw_label_match	Input XEP-0258 label must match raw label	Sets Peer Mandatory XEP-0258 Label and Require Mandatory Label
sio_policy	SIO policy	Sets Peer Relabelling Security Policy
sio_clearance	SIO clearance	Sets Peer Security Clearance
sio_label	SIO label	Sets Peer Security Label
sio_default_label	SIO default label	Sets Peer Default Stanza Security Label
raw_label	Raw label	Sets Peer Mandatory XEP-0258 Label
xep258_format	XEP-0258 security label format	Sets Peer Outbound Security Label Format
flot	Inject FLOT for labels	Sets Peer Add Flot
provide_default	Provide default labels	Sets Peer Outbound Default Label
sio_deny_unrecognized_labels	Deny unrecognized XEP-0258 labels	Sets Peer Reject Unrecognized Label

Table K.3. Common Domain Options

Name	Description	Support
name	Domain name	Sets Domain Name and Service Name
archive	Archiving configuration	Sets domain ArchiveType
tls_cert	TLS certificate	Sets domain TLS Key Pair
tls_cert_file	TLS certificate	Sets domain TLS Key Pair
tls_key	TLS Key	Sets domain TLS Key Pair
tls_key_file	TLS Key	Sets domain TLS Key Pair
tls_key_password	TlS key password	Sets domain TLS key pair
jid_filter	Jid filter	See Section I.10, “JID Filtering”
iq_filter	IQ filter	See Section I.9, “Folding and Filtering”
sio_clearance	SIO clearance	Sets Domain Clearance
sio_label	SIO label	Sets Domain Security Label

Name	Description	Support
sio_default_label	SIO default label	Sets Domain Default Stanza Security Label

Table K.4. IM Domain Specific Options

Name	Description	Support
auth	Authorization	Configures Authorization, see Section I.11, “Authorization”
sio_default_clearance	Default clearance	Sets User Default Clearance

Table K.5. PubSub Domain Specific Options

Name	Description	Support
identity_type	Supported identity type	Enables FDP support, but only if type is `urn:xmpp:fdp:0`. All other identities will be ignored.