IC-1006-V1.1

Administrator's Guide
Message Store

ISODE
CONSORTIUM

This manual is associated with ISODE Consortium Release 2.1, September 1994. It is the version 1.1, of ***IC-1006: Administrator's Guide: Message Store.*** Version 1.0: May 1994.This Version: September 1994.

This documentation and the software are based on contributions from multiple sources. Acknowledgement of contributions are given in ***IC-1001: Introduction to ISODE Consortium, its Products and Membership***.

Table of Contents

List of Figures

Preface

Software Version

This Guide is published in support of ISODE Consortium Release 2.1 (IC R2.1).

Readership

This Guide is intended for administrators who need to set up or manage Message Stores.

It is assumed that readers are already familiar with the contents of ***IC-1001: Introduction to ISODE Consortium, its Products and Membership*** and ***IC-1003: Administrator’s Guide: General Services.***

What this Guide Covers

This Guide covers the configuration and management of the IC R2.1 Message Store.

Chapter 1 is a general introduction. It includes information about the relationship between the Message Store and the MTA, and the function of User Agents with regard to the Message Store.

Chapter 2 describes the activities you must complete to set up and run the Message Store successfully.

Appendix A contains a Glossary and Abbreviations descriptions.

Appendix B contains the References to other documentation relevant to the current Guide or its area of functionality.

Related Publications

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

Volume	Title
1001	Introduction to ISODE Consortium, its Products and Membership
1002	Administrator’s Guide: ISODE Installation
1003	Administrator’s Guide: General Services
1004	Administrator’s Guide: Message Handling Services
1005	Administrator’s Guide: Directory Services
1011	Programmer’s Guide: Message Store

Typographical Conventions

The text of this Guide uses the typographical conventions listed below.

- File and directory names appear in 9pt italic Avantgarde type, thus - *isoentities, isoservices*.
- Program names use 9 pt bold Helvetica type, thus - **mkpasswd**.
- Data input to or output from the system is in 9pt Courier type, thus - `headertype p2 ipn p22`
- Required or literal data input is set in **`BOLD 9pt courier`** if it is in an example. Otherwise the status of options, attributes, etc, is indicated by the attachment of the words MANDATORY, OPTIONAL and/or DEFAULTED.
- Where a default exists, the default value is explicitly stated.
- You should use upper and lower case characters as indicated by examples, unless instructed that the case is unimportant.
- When options, attributes and so on are described, they are set in 10pt italic bold Palatino, followed by a colon and a description or discussion, set in 10pt regular Palatino. For example:

  directory_file: If present, specifies a file which maps Message Store users' Distinguished Names to root directories, as an alternative to using an X.500 directory service.

  When they are referred to in the flow of text, they are treated as input, so they are set in 9pt Courier, thus - `directory_file`.
- C code and ASN.1 is set in 9pt HELVETICA regular and HELVETICA bold, for example

  NonSpecificSubordinateReference **::= ATTRIBUTE**
  WITH ATTRIBUTE-SYNTAX AccessPoint
- References in the text to other documents or ISODE Volumes are set in 10 pt italic bold Palatino, thus - ***Administrator's Guide: Message Handling Services***.

Note: In the case of ISODE Volumes, use the table earlier in this Preface to find the volume number associated with the document title. Other documents can be located through the ***References*** Appendix at the end of this Guide.

Special Directory Names

There are several special directory names used in this document. They are place holders for the names of actual directories where particular ISODE objects are stored. The use of these special directory names is explained below. The actual names are defined during installation: refer to ***IC-1002:Administrator's Guide: Installation*** for information.

$(ETCDIR): Place holder for the name of the directory that holds configuration files. The default used in examples is */opt/IC/etc/*.

$(LIBDIR): Place holder for the name of the directory that holds the libraries. The default used in examples is */opt/IC/lib/*.

$(SBINDIR): Place holder for the name of the directory that holds the ISODE administration programs. The default used in examples is */opt/IC/sbin/*.

$(BINDIR): Place holder for the name of the directory that holds the ISODE user programs. The default used in examples is */opt/IC/bin/*.

$(PPDIR): Place holder for the PP MTA files and directories. The default used in examples is */opt/IC/pp/*.

Bug Reporting

ISODE Consortium welcomes bug reports concerning its software releases.

Bug reports may be sent to the Consortium by any means. However, we prefer to receive it by electronic mail to one of the addresses shown below. Please send proposed fixes with reports if possible. Changes resulting from bug reports may be included in future releases.

The electronic mail bug reporting addresses are:

- bug-isode@isode.com for general ISODE reports.
- bug-quipu@isode.com for reports on QUIPU (X.500 Directory).
- bug-pp@isode.com for reports on PP (the Message Handling Service) and the Message Store
- bug-snmp@isode.com for reports on SNMP.

Reports from non-members of the ISODE Consortium, will be acknowledged, but no further action is guaranteed.

Information on ISODE Consortium, its membership classes, and its postal addresses can be found in the Information section at the back of this Guide.

Revision History

This document is the first revision of I***C-1006-V1: Administrator's Guide: Message Store***. Changes and additions to the text of the preceding edition are marked by a bar in the left margin.

Substantive changes in this edition have taken place in the following sections:

- Section 2.1, ***Message Store Tailoring,*** on page 7
- Section 2.2, ***MTA Tailoring,*** on page 9
- Section 2.4, ***User Configuration,*** on page 11

The following section has been added:

- Section 2.5, ***Use of the X.500 Directory,*** on page 13

Chapter 1
Introduction

1.1 Role of the Message Store

The primary role of Message Store is to accept the delivery of messages from the MTA on behalf of a user, and to retain them for subsequent retrieval by the user's UA.

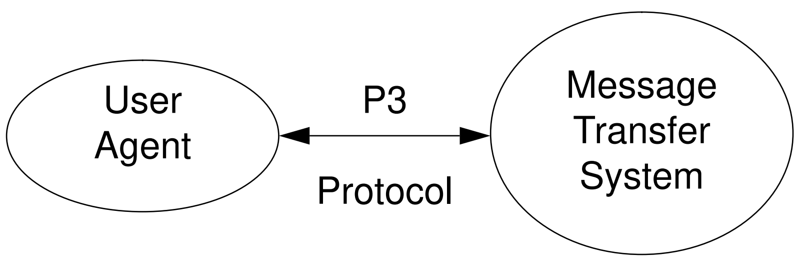
A secondary function is to provide indirect message-submission and message-administration services to the UA. This enables the Message Store to provide additional functionality to that available through direct submission to the MTA, such as forwarding messages residing in the Message Store.

A Message Store (MS) is, therefore:

- a provider of messaging services for User Agents (UAs), and
- a user of messaging services provided by Message Transfer Agents (MTAs).

In a messaging system which does not use a Message Store, outbound messages go directly and immediately from the User Agent (UA) to the Message Transfer Agent (MTA) and inbound messages go directly and immediately from the MTA to the UA, as shown in Figure 1, ***Message Transfer without a Message Store.,*** on page 1.

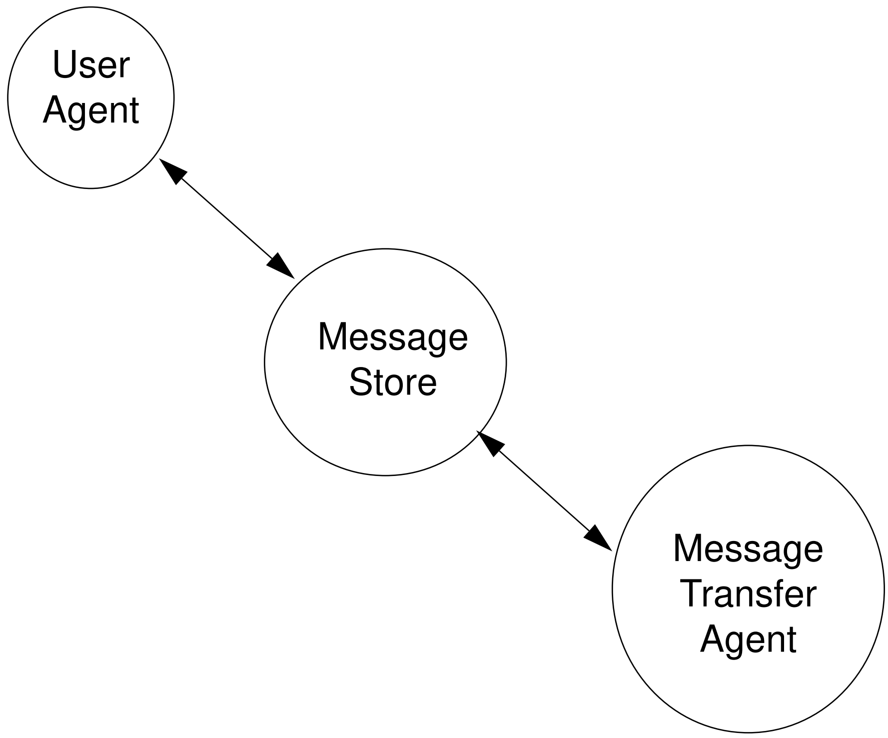
Figure 1: Message Transfer without a Message Store.



In the basic CCITT model, both inbound and outbound messages are transferred using the P3 protocol. However, the actual protocol used is dependent on the message handling system that is in place and whether or not the MTA and the UA are co-resident.

When a Message Store is in use, messages between a UA and the MTA do not have to be handled immediately; they can be stored for later collection or transmission. The basic relationships for a Message Handling System which includes a Message Store are shown in Figure 2, ***Message Transmission and Retrieval using a Message Store,*** on page 2.

Figure 2: Message Transmission and Retrieval using a Message Store



In the Open Systems Interconnection (OSI) model for implementation of a Message Store, only one protocol is used for communication between the UA and the Message Store - the P7 protocol.

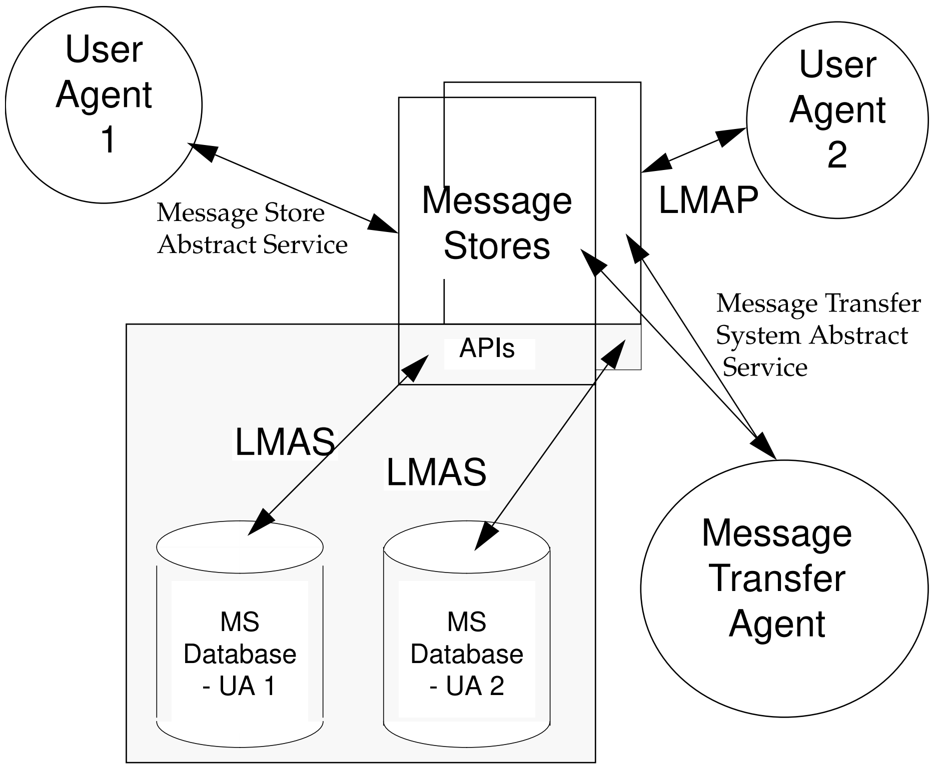
The ISODE Consortium Message Store, however, allows multiple protocols to be used, including P7 and the Lightweight Mail Access Protocol (LMAP), whose functionality is a superset of the P7 protocol. It also allows for the MTA, MS, and UA to be co-resident.

A Message Store has a one-to-one relationship with a UA; that is, a Message Store must be set up for each UA. There is no restriction on the number of Message Stores that can co-exist.

Message to and from a UA are stored in an associated database by the Message Store until they are ready for collection or transmission. There is one Message Store database per UA. Each database both forms an integral part of the Message Store and is a unique entity, which is accessible through a set of APIs.

The relationships between the Message Store implementation, its databases and the MTA and UAs are shown in Figure 3, ***Message Transmission and Retrieval with the ISODE R2 implementation of Message Store,*** on page 3.Refer to section 1.2 for more information on the services used, and to section 1.5 for a brief description of the Application Program Interfaces (APIs) to Message Store databases.

Figure 3: Message Transmission and Retrieval with the ISODE R2 implementation of Message Store



1.2 Services Used by the Message Store

Three services are used by the Message Store:

- Message Store Abstract Service for communication with User Agents;
- LMAS (Lightweight Mail Abstract Service) for access to the Message Store database;
- Message Transfer System Abstract Service for communication with the MTA.

The first two enable the Message Store to provide services to User Agents and the third is used by the Message Store in the provision of those services.

1.3 MTS/MS Interactions

The MTS interactions are performed through the Message Transfer System Abstract Service and a proprietary API, using the **/submit/** io protocol, equivalent to the P3 protocol.

The MTA must be co-located with the Message Store.

1.4 Message Store Processes

The Message Store consists of two processes:

- a delivery channel which delivers messages from a co-located MTA into the Message Store
- a server process which allows P7 User Agents (UAs) to access the Message Store and to (indirectly) submit messages to the MTA

Underlying both of these processes is an API to a specialized database which uses a combination of directory hierarchies, plain text and **ndbm** files to store and access messages in a protocol-independent manner. Refer to ***Figure 3***, above, for a schematic of the relationships.

Database Management and storage functionality are separated in a manner which allows the storage implementations to be replaced with alternative implementations supplied if required.

1.4.1 The P7 Server Process

The P7 Server process allows User Agents which use the P7 protocol to list, summarize, retrieve and delete messages from the Message Store and to submit messages to the MTA.

Each Message Store user (identified by a unique O/R address) is allocated a section of the Message Store database into which messages addressed to that user will be delivered. This configuration is performed using tables which are described in sections 2.1 and 2.4 of this Guide. Refer to ***IC-1004: Administrator’s Guide: Message Handling Services*** for more information.

1.4.2 The Message Store Delivery Channel

The Message Store Delivery Channel is invoked by the MTA to deliver messages addressed to Message Store users. It uses the same tables as the P7 Server process to determine the section of the Message Store database into which a message should be delivered. The Delivery Channel is also responsible for generation of delivery and non-delivery reports, if requested.

1.5 Message Store APIs

The ISODE Consortium implementation of Message Stores allows the use of multiple protocols. Therefore, the protocols used for message transfer between the MS and the UA can vary depending on the needs of the site.

At present, the protocol implementations that can be used to access the MS database are:

- P7 service, as described in section 1.3;
- LMAP;
- LMAS;
- the `ms-delivery` function of PP.

The construction of the APIs is such that the database appears as an extension of the UA and is, in practice, effectively transparent.

Refer to ***IC-1011-V1.0: Programmer's Guide: Message Store, IC-1014: Programmer's Guide: X.400 MTA*** and ***IC-1015: Programmer's Guide: X.400 UA*** for more information.

1.6 User Agent/Message Store/MTA Functional Relationship

Figure 4, ***Functional Relationships between UAs, the MTA, and Message Stores.,*** on page 6 shows the relationship between User Agent, Message Store and the MTA functions.

The UAs are shown as ovals in the upper portion of the diagram (left of page). The Message Store functions take place in the lightly shaded area on the left of the diagram (bottom of page). The MTA functions are shown in the more heavily shaded area at the right of the diagram (top of page). The striped area indicates the points of interaction between the MTA and the Message store.

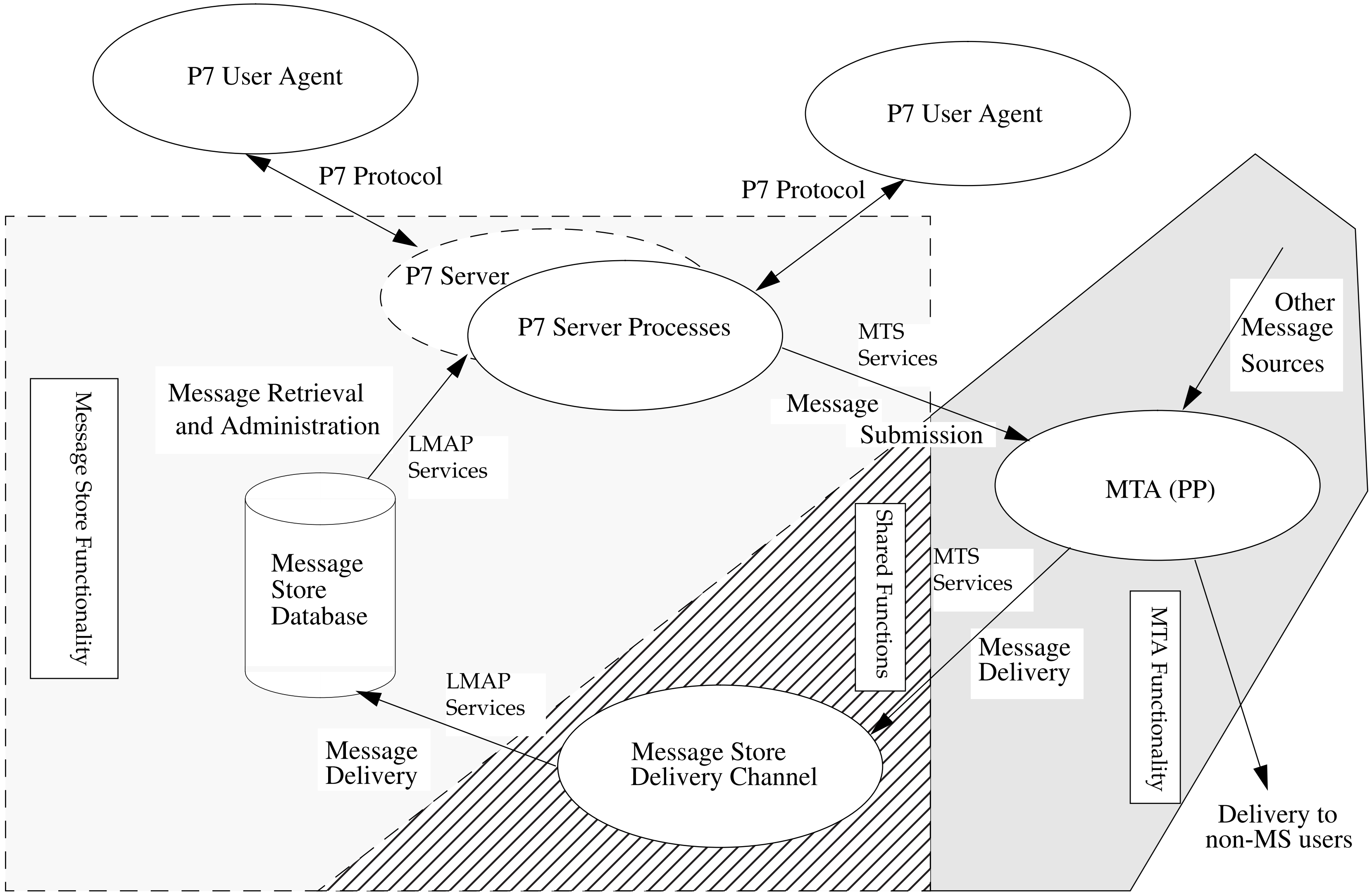


Figure 4: Functional Relationships between UAs, the MTA, and Message Stores.

Chapter 2 Configuration

2.1 Message Store Tailoring

A single tailoring file, *pumicetailor*, located in *$(ETCDIR)*, provides configuration information to both the delivery channel and the P7 server processes. An example file is shown below.

Section 2.1.1, below, describes each tailoring options and states whether it affects the delivery channel, the P7 server process, or both.

```
# Tailoring file for Message Store
ldap_host: glengoyne
ldap_port: 389
directory_file: /opt/IC/etc/lmap_dns
mhsds_file: /opt/IC/etc/lmap_oraddrs
p7server_password: p7serverpassword
dchan_name:"CN=msdliv/OU=MYORG/C=GB/"
dchan_password:nullpassword
submit_channel: ms-x400
lmap_submit_channel: x.400
loglevel: level=notice level=exceptions level=fatal
delete-correlated: FALSE
```

2.1.1 Tailoring Options

The tailoring options are:

`ldap_host`: OPTIONAL - Only required if directory look-up is to be used for user configuration. Specifies the domain name of the host on which an LDAP (Lightweight Directory Access Protocol) server process is running. Affects both the delivery channel and the P7 server process

`ldap_port`: OPTIONAL - Only required if directory look-up is to be used for user configuration. Specifies the TCP port number to use when attempting to connect to an LDAP server. DEFAULTS to 389. Affects both the delivery channel and the P7 server process.

`directory_file`: OPTIONAL - If present, causes file-based look-up to be used, replacing directory look-up, for user configuration. Refer to section 2.4.2.2 for more information about Directory files.

mhsds_file: MANDATORY - Specifies a file which maps Message Store users' O/R Addresses to Distinguished Names. Refer to section 2.4.2.1 for more information about file formats.

p7server_password: MANDATORY - The password which a P7 Server process will use to authenticate itself to User Agents.

dchan_name: MANDATORY - The Distinguished Name which the delivery channel will use when binding to the Message Store.

dchan_password: MANDATORY - The password which the delivery channel will use when binding to the Message Store.

submit_channel: MANDATORY - The name of the MTA inbound channel to which a P7 Server process will submit messages.

lmap_submit_channel: MANDATORY - May be x.400 or rfc822. Identifies the channel which the **Imapd** process uses to submit messages to the MTA.

The addressing style configured for the channel in the MTA tailoring file controls the content type which **Imapd** generates at submission time.

- If the configured addressing style is x.400, **Imapd** use X.400 addressing (that is, expects ORAddresses to be present in the envelope and header objects being submitted) and generates a P2 or P22 content type.
- If the addressing style is rfc822, **Imapd** expects rfc822-mailbox attributes to be present in the envelope and header and submits the message as an RFC822 header followed by individual body parts.

loglevel: OPTIONAL - Accepts a set of standard MTA logging switches to configure the level, filename and various other parameters of Message Store logging. DEFAULTS to notice and exceptions. Refer to the ***IC-1004: Administrator's Guide: Message Handling Services*** for further details.

delete-correlated: DEFAULTED - Controls whether the database performs automatic deletion of reports and messages that are correlated on mts-id or message-id. Defaults to TRUE, but must be set to FALSE to be used by the P7 server process.

2.2 MTA Tailoring

Modification of the MTA *tailor* file is required to enable messages to be delivered into the MS by the MTA, and to enable the P7 server process to submit messages to the MTA. The modification must configure at least one channel for the MS.

For the use of the P7 server process, the following entry for a local channel should be added:

```
chan ms-x400 prog=msdliv,show="Message Store(X.400)",type=both,
              access=mts,content-out="none", cost=1,
              sort="user time",maxproc=1,
              solo-proc=yes,adr=x400,hdrout="p2,p22",
              bptout="ia5,g3fax,external,bilateral,undefined,ttx,
                      videotex,national,encrypted,tif1"
```

If use of the **Imapd** server process with RFC822-type addressing and content is required, the following entry should also be added:

```
chan ms-822 prog=msdliv,show="Message Store(822)",type=both,
              access=mts,content-out="none", cost=1,
              sort="user time",maxproc=1,
              solo-proc=yes,adr=822,hdrout="822-us",
```

In addition, all body part types required to be delivered to or submitted by the MS need to be configured in the MTA *tailor* file, as shown below. Refer to the ***IC-1004: Administrator's Guide: Message Handling Services*** for further details.

```
# General body types
headertype p2 ipn p22
bodypart  ia5 external g3fax bilateral undefined ttx
```

2.2.1 Routing Authorization

On submission, an authorization table may be used to determine which user IDs are permitted to send as particular originator addresses. The format of this table is described in ***IC-1004:Administrator's Guide: Message Handling System***, in the section titled ***Submit Authentication***.The information in the table is used in the MTA routing authorization checks, also described in ***IC-1004***.

If you have two MS channels, each using a different addressing style, you must set up two authorization tables. This is because the table is required to be in the same addressing style as the channel for which it is used.

To enable authorization, you must add the following entries to the MTA *tailor* file:

```
tbl  auth.msx400 show="Auth table for MS X.400 submission", flags=linear
tbl  auth.ms822  show="Auth table for MS RFC822 submission", flags=linear
```

You must also add one of the following parameters to the appropriate channel entries:

```
,auth-table=auth.ms400
,auth-table=auth.ms822
```

If you do not add this configuration information, submission will proceed normally but a warning message will be generated in the MTA's **submit** log each time a message is submitted.

2.3 Process Configuration

2.3.1 P7 Server Configuration

The *isoservices* file must be edited to enable P7 Server processes to be started automatically when User Agents request connection. An example addition is shown below.

```
tsap/p7  "3001"  /opt/IC/MTA/cmds/chans/p7server
```

This configures **tsapd** to start a P7 Server process when a connection request is made on TSAP "3001".

2.3.2 User Agent Configuration

The *isoentities* file must be edited if a User Agent which is based on ISODE Consortium software is to be used. An example addition is shown below.

```
glengoyne  "p7server"  NULL  "3001"/Internet=glengoyne.isode.com
```

This allows a User Agent to connect to a P7 Server process started on request by **tsapd**, running on host `glengoyne`.

If a third-party User Agent is to be used, this configuration is not relevant.

2.3.3 Configuration for use with IAED

iaed and X.500 directory look-up may be used as an alternative to using **tsapd** and static tables to configure the manner in which P7 services are made available to UAs. Refer to ***IC-1002: Administrator's Guide: Installation,*** for information about how to set up **iaed**.

By convention, all OSI application services offered by a given end-system are placed in the location in the DIT, under an **applicationProcess** entry with the short name of the end-system; for example, "`cn=glengoyne`".

The two example file entries in sections 2.3.1 and 2.3.2 could be replaced by a single X.500 entry with the **DistinguishedName** "`cn=p7server,cn=glengoyne,o=ISODE Consortium, c=GB`", and containing the following attributes:

```
objectClass = iSODEApplicationEntity
commonName = p7server
description = Message Store server process for P7 User Agents
presentationAddress = "3001"H/Internet=glengoyne.isode.com
supportedApplicationContext  = 2.6.0.1.4
execVector = /opt/IC/pp/cmds/chans/p7server
```

Refer to ***RFC 1485: String Representation of Distinguished Names*** for information about defining and specifying Distinguished Names.

2.4 User Configuration

2.4.1 MTA Configuration

Each MS user requires configuration within the MTA. In its simplest form, all that this requires is an entry for the user in the *users* table. For example, if a user with the ORAddress

```
"/I=F/S=Bloggs/O=ISODE Consortium/P=ISODE/A=MAILNET/C=FI/"
```

were configured as an MS user, the *users* file for the MTA that receives mail for that ORAddress would require an entry of the form:

```
F.Bloggs: ms-x400
```

If the MTA has been configured to use authorization tables on submission, an equivalent entry must be added to the appropriate authorization table for the submission channel to be used by that user. For the user shown in the preceding example, this would require an entry in the auth.ms400 table, with the form:

```
/i=F/s=Bloggs/o=ISODE Consortium/P=ISODE/A-MAILNET/C=GB/:pp
```

Further information about the MTA *users* table can be found in ***IC-1004: Administrator's Guide: Message Handling Services***.

2.4.2 Message Store Configuration

2.4.2.1 Mapping O/R Addresses to Distinguished Names

On delivery, the Message Store uses either the file specified by the *mhsds_file* tailoring entry or an X.500 directory, through a Lightweight Directory Access Protocol (LDAP) interface, to perform a mapping from an user's O/R Address to a Distinguished Name.

An example section of this file is shown below.

```
# This file maps mailbox ORAddress to user DN. It is used as a substitute for a
# mhs-ds directory service.
# Format of records:
# Mailbox ORAddress: User DN
# Remember to quote any field which has embedded spaces!
#
"/I=F/S=Bloggs/O=ISODE Consortium/PRMD=ISODE/ADMD=MAILNET/C=FI/":
                                   "CN=Fred Bloggs;O=ISODE Consortium;C=GB"
```

Notes:

1. In IC R2.0, the Message Store always requires the use of the file-based mapping method
2. Each entry is a single line (that is, there is no carriage return after the colon characters).
3. Refer to ***RFC1327; Chapter 4*** for the syntax definition of O/R addresses.

2.4.2.2 Message Store User Configuration

The Message Store processes use X.500 Directory entries to store information about Message Store users and to perform user authentication. Directory access is performed through use of the LDAP protocol.

As an alternative, lookups in a text file (specified by the `directory_file` tailoring entry) can be performed instead.

If a real X.500 directory is used, an entry for each Message Store user must be configured with the following MANDATORY attributes:

mailboxRoot: The UNIX absolute pathname of the directory which is to form the user's Message Store root folder.

mailboxUid: The UNIX userid under which to access the user's Message Store entries.

mailboxGid: The UNIX groupid under which to access the user's Message Store entries.

The Directory entry should be configured with a password using simple authentication. This password will be the one required when performing a P7 bind operation to the P7 server process.

If X.500 Directory access is not desired, adding a `directory_file` entry to the *pumicetailor* file will cause the specified file to be used instead. An example set of entries is shown below.

```
# This file maps DN to mailbox ORAddress, location etc. It is used
# as a substitute for a directory service.
# Format of records:
# User DN: Mailbox location: Mailbox ORAddress: uid: gid: password
#
"CN=Fred Bloggs;O=MYORG;C=GB": /ic/dev/users/fred/.ms,
                   "/I=F/S=Bloggs/O=MYORG/PRMD=MYPRMD/ADMD=MYADMD/C=GB/",
                                                  125, 500, PTThhuzGW94As
"CN=msdliv;OU=MYOU;o=MYORG;C=GB": /null,
       "/S=msdliv/O=MYORG/PRMD=MYPRMD/ADMD=MYADMD/C=FI/",0,0,jLuNiYrjFzK/A
```

Note: Each entry is a single line (i.e. there is no carriage return after the comma characters).

The Mailbox O/R Address element of each record is not currently used. However, an entry is expected and at least a placeholder should be present.

The Mailbox location, uid and gid elements serve the same purpose as their corresponding Directory attributes. The password element is the encrypted form of the user's plain text password. This encryption should be generated by using the PP **mkpasswd** utility, which can be found in the *$(PPDIR)* directory.

A Directory or file entry is required for the Message Store delivery channel. The Distinguished Name and password used should match that configured in the *$(ETCDIR)/pumicetailor* file.

2.5 Use of the X.500 Directory

The MS has the ability to use an X.500 Directory to authenticate and configure users. The MS uses the LDAP protocol to access the X.500 Directory.

2.5.1 MS Configuration

You must perform the following steps to configure the MS to use the X.500 Directory:

1. Uncomment the definition of **LDAPTOO** contained in *Imap.h*. This causes code which makes use of LDAP to be compiled into the MS.
2. Rebuild and install the MS libraries. Refer to ***IC-1011-V1.0:Programmer's Guide: Message Store*** for a list of MS libraries.
3. Rebuild the MS processes (**Imapd, p7server**, and **msdliv**). Note that the Makefiles for these processes will need to be edited to enable the definition of **LIBBER**, which is normally commented out.
4. Install the MS processes.

5. Ensure that an LDAP server process is running on a suitable host, and that the LDAP server can access the portion of the DIT that holds entries for MS users.
6. Check the port on which the LDAP server is listening. For information on how to configure the LDAP server, consult the appropriate **man** pages.
7. Edit *pumicetailor*, the MS tailoring file, to remove (or comment out) any definition of the tailoring variable `lmap_dir`, and add a line in the form

   ```
   ldap_host:hostname
   ```

   where `hostname` is the name of the host on which the LDAP server is running.
8. If the LDAP server is listening on a non-default port, add to *pumicetailor* the tailoring option

   ```
   ldap_port:nnnn
   ```

   where `nnnn` is the port on which the LDAP server is listening.
9. To allow the MS delivery channel (**msdliv**) to bind successfully to the Directory, add an entry for it. The entry will usually be of object class **applicationEntity.** It does not need any of the MS-specific attributes.
10. The **DistinguishedName** under which **msdliv** binds to the Directory is configured by the `dchan_name` entry in the *pumicetailor* file.

    The DN assigned to **msdliv** usually follows the same conventions as for entries for **iaed**: that is, the name is consists of the *domain+host on which the process runs+process name*.

    For example, **msdliv** running on host **glengoyne** at the ISODE Consortium would have the DN of:

    ```
    CN=msdliv:CN=glengoyne:O=ISODE Consortium;C=gb
    ```
11. In addition to the Directory entry, you must make a corresponding entry for the delivery channel in the file that maps ORAddresses back to DNs. This file is tailored by the variable `mhsds_file`.
12. A plain text password can also be configured if needed, by adding a `dchan_password` tailoring variable.
13. Ensure that your directory OID tables contain the necessary definitions for MS users. Since LDAP is being used to access the Directory, only the textual names assigned to attributes are important, not the OIDs assigned to object classes, etc. An example of the additions required is given below in Figure 5, ***Additions to OID tables for MS use of X.500 Directory,*** on page 15.
14. Ensure that the OID tables are accessible to the LDAP server as well as the Directory Service Agent (DSA)itself. Refer to ***IC-1005:Administrator's Guide: Message Handling System*** for more information on adding new OIDs to the Directory.

Figure 5: Additions to OID tables for MS use of X.500 Directory

Additions to oidtable.gen
`ic-ms:        isode-consortium.8`
`ic-ms-oc:     ic-ms.1`
`ic-ms-at:      ic-ms.2`
Additions to oidtable.at
`mailboxRoot:   ic-ms-at.0   : PrintableString`
`mailboxAddr:   ic-ms-at.1   : PrintableString`
`mailboxUid:    ic-ms-at.2   : Integer`
`mailboxGid:    ic-ms-at.3   : Integer`
Additions to oidtable.oc
`msuser: ic-ms-oc.1: top : mailboxRoot, mailboxAddre, mailbocUid, mailboxGid ;`

2.5.2 User Configuration

You must carry out the following steps to configure MS users to use the Directory.

1. Add DIT entries for every user of the MS that you want to configure. Each entry must have:
   - **msUser** as an auxiliary object class
   - the attributes **mailboxRoot, mailboxAddr, mailboxUid, mailboxGid**

   The attributes have the following meanings and settings:

   mailboxRoot: Specifies the UNIX directory that forms the root of the user's MS database. An example for user **fred** would be */usr/spool/pp/ms/fred*. Note that this directory must be created by hand before the first bind attempt is made by the MS user. The directory should be set to have read, write and execute permissions for group and owner, with owner set to the userID identified by the symbol **PPUSER** in *src/config/Make.pp*.

   mailboxAddr: A printable string encoding of the ORAddress of this mail user. An example is:

```
/I=F/S=Bloggs/O=ISODE Consortium/PRMD=ISODE/ADMD=MAILNET/C=FI/
```

   mailboxUid: Can be set to 0 (zero). Not currently used.

   mailboxGid: Can be set to 0 (zero). Not currently used.

2. It is preferable, but not mandatory, that the DIT entry has a **userPassword** attribute, to allow the use of simple authentication between the UA and the MS.
3. For each DIT entry, create a corresponding entry in the file identified by the `mhsds_file` tailoring variable. This file maps from ORAddress back to Distinguished Name. The format of the file is:

```
"ORAddress":"DistinguishedName"
```

4. Finally, the user must be configured as an MS user in the MTA tailoring file. Refer to Section 2.4.1, ***MTA Configuration,*** on page 11 for more information.

2.6 Other Configuration

The directory specified as a user's root folder must exist prior to Message Store use. Since both the delivery channel and the P7 server process run under the userid of the MTA (usually **pp**), this root folder directory must be readable, writeable and searchable by the MTA userid.

2.7 Scan Default

The LMAP protocol permits Scanning and Finding (roughly equivalent to P7 List/Summarize operations) on a subset of the attributes valid for each type of primitive object within the Message Store database (that is, headers, bodyparts, delivery envelopes etc). Refer to ***Lightweight Mail Access Protocol V.1.8*** for further information regarding LMAP.

The configuration of scannable attributes ("scan defaults") is currently performed by a set of arrays contained within *pumice/Lib/lmapdb/dbscandf.c*. If scanning on additional attributes is required, these arrays can be modified. The Message Store database library, P7 server process and Message Store delivery channel should then be rebuilt.

The modified scan defaults will only take effect for messages delivered into the Message Store after the new version of the delivery channel has been installed.

Attempts to locate messages using selection criteria which include attributes not configured as scannable will fail.

Appendix A
Glossary & Abbreviations

The terms and abbreviations necessary for use of this book are explained below.

Distinguished Name: A set of attributes that uniquely identifies an object or user within a global set of network addresses. It is constructed of concatenated values, in a similar way to the construction of a file name. In the case of a Distinguished Name, the values identify such things as the country, organisation, organisational group and user name associated with the user.

Directory Information Tree: A way of defining relationships between, and access routes to, the complete set of information contained in the X.500 Directory.

DIT: See Directory Information Tree.

DN: See Distinguished Name.

LDAP: See Lightweight Directory Access Protocol.

Lightweight Directory Access Protocol: Protocol used to provided simple access to the X.500 Directory.

Lightweight Mail Access Protocol: Protocol used to access the Message Store Database.

LMAP: See Lightweight Mail Access Protocol.

Message Store: A means of providing a continuously available, secure storage place for messages being transferred either to or from a UA. Each Message Store services only one O/R address. However, a Message Store database consisting of many Message Stores can be configured to service multiple O/R addresses.

MS: See Message Store.

Message Transfer Agent: A component of the MTS, which, in co-operation with other MTAs, transfer and deliver messages to the intended recipients.

Message Transfer System: The MTS delivers, to one or more recipients, messages submitted to it by UAs. An MTS consists of a number of MTAs

MTA: See Message Transfer Agent.

MTS: See Message Transfer System.

O/R:Originator/Recipient - the User Agent sending a message or to whom a message is to be delivered.

O/R Address: A collection of information (*an attribute list*) that uniquely identifies the User Agent to whom a message is to be delivered or notification of delivery returned. The O/R address also identifies the user's point of access to the MHS.

Open Systems Interconnection: Set of design rules and protocol specifications defined to allow systems from any source to talk to each other, thus evading problems of hardware and software incompatibility.

OSI: See Open Systems Interconnection

P3: The X.400 protocol used in communications to and from an MTA.

P7: The X.400 message handling protocol used to access a Message Store.

UA: See User Agent.

User Agent: A User Agent is an application process that interacts with the MTS or a Message Store, to submit messages for a single user. It can also accept delivery of messages, either directly from the MTS or by retrieving them from a Message Store.

Appendix B References

B.1 Cited Documents

Refer to the documents listed below for more information regarding issues raised or referred to in this Guide.

ITU-T (formerly CCITT) Recommendation X.400: Message Handling Systems - System and Service Overview

ISO/IEC 10021-1:1990: Information Technology - Text Communication - Message-Oriented Text Interchange System (MOTIS) - System and Service Overview. [International Standards Organisation equivalent to X.400]

ITU-T Recommendation X.402: Message Handling Systems - Overall Architecture

ISO/IEC 10021-2:1990: Information Technology - Text Communication - Message-Oriented Text Interchange System (MOTIS) - Overall Architecture. [International Standards Organisation equivalent to X.402]

ITU-T Recommendation X.413: Message Handling Systems - Message Store: Abstract Service Definition

ISO/IEC 10021-5:1990: Information Technology - Text Communication - Message-Oriented Text Interchange System (MOTIS) - Message Store: Abstract Service Definition. [International Standards Organisation equivalent to X.413]

ITU-T Recommendation X.419: Message Handling Systems- Protocol Specifications

ISO/IEC 10021-6:1990: Information Technology - Text Communication - Message-Oriented Text Interchange System (MOTIS) - Protocol Specifications. [International Standards Organization equivalent to X.419]

ITU-T Recommendations Series X.500: The Directory

ISO/IEC 9594: 1990: [International Standards Organisation equivalent to X.500]

ISODE Volume 2: Administrator’s Guide: General Services

ISODE Volume 4: Administrator’s Guide: Message Handling System

RFC 1327: Mapping between X.400(1988)/ISO 10021 and RFC822. S. E. Hardcastle-Kille

RFC 1485: String Representation of Distinguished Names. S. E. Hardcastle-Kille

RFC 1487: X.500 Lightweight Directory Access Protocol; W. Yeong, T. Howes, & S. Kille.

Lightweight Mail Access Protocol V.1.8; S.E. Kille & M. Wahl; 1994, March 2. ISODE Consortium Limited internal document.

B.2 Obtaining Cited Documents

B.2.1 All Documents except ISODE Consortium Internal Papers

You can obtain ITU-T (CCITT) Recommendations, ISO/IEC Standards and draft standards, and paper copies of RFCs from:

Omnicom PPI Ltd. Forum Chambers The Forum Stevenage Herts SG1 1EL ENGLAND	Phillips Publishing Inc. 7811 Montrose Road Potomac MD 20854 USA
Telephone: +44 438 742424 Fax: +44 438 740154	+1 301 424 3338 +1 301 309 3847

B.2.2 ISO/IEC Documents

Contact your national Standards Organization.

B.2.3 ITU-T (CCITT) Documents

CCITT/ITU-T Recommendations and draft documents can be obtained from:

International Telecommunications Union
General Secretariat - Sales Section
Place des Nations
CH-1211 Geneva 20
Switzerland

B.2.4 RFCs

Electronic copies of RFCs can be obtained by an anonymous ftp call to nic.ddn.mil.

Paper documents can be obtained through Omnicom/Phillips.

B.2.5 ISODE Consortium Internal Papers

ISODE Consortium internal papers are obtainable on request to ISODE Consortium Limited at the addresses shown in Appendix C.

Index