Network Working Group
Request for Comments: 2156
Obsoletes: 987, 1026, 1138, 1148, 1327, 1495
Updates: 822
Category: Standards Track

S. Kille
Isode Ltd.

January 1998

MIXER (Mime Internet X.400 Enhanced Relay): Mapping between X.400 and RFC 822/MIME

Status of this Memo

Copyright Notice

Table of Contents

1 - Overview 3
1.1 - X.400 3
1.2 - RFC 822 and MIME 3
1.3 - The need for conversion 4
1.4 - General approach 4
1.5 - Gatewaying Model 5
1.6 - Support of X.400 (1984) 8
1.7 - X.400 (1992) 8
1.8 - MIME 8
1.9 - Body Parts 8
1.10 - Local and Global Scenarios 9
1.11 - Compatibility with previous versions 10
1.12 - Aspects not covered 10
1.13 - Subsetting 11
1.14 - Specification Language 11
1.15 - Related Specifications 11
1.16 - Document Structure 12
1.17 - Acknowledgements 12
2 - Service Elements 13
2.1 - The Notion of Service Across a Gateway 13
2.2 - RFC 822 15
2.3 - X.400 18
3 - Basic Mappings 27
3.1 - Notation 27

3.2 - ASCII and IA5 29
3.3 - Standard Types 29
3.4 - Encoding ASCII in Printable String 33
3.5 - RFC 1522 34
4 - Addressing and Message IDs 35
4.1 - A textual representation of MTS.ORAddress 36
4.2 - Global Address Mapping 43
4.3 - EBNF.822-address <-> MTS.ORAddress 46
4.4 - Repeated Mappings 59
4.5 - Directory Names 62
4.6 - MTS Mappings 62
4.7 - IPMS Mappings 67
5 - Detailed Mappings 71
5.1 - RFC 822 -> X.400: Detailed Mappings 71
5.2 - Return of Contents 86
5.3 - X.400 -> RFC 822: Detailed Mappings 86
Appendix A - Mappings Specific to SMTP 114
1 - Probes 114
2 - Long Lines 114
3 - SMTP Extensions 114
3.1 - SMTP Extension mapping to X.400 114
3.2 - X.400 Mapping to SMTP Extensions 115
Appendix B - Mapping with X.400(1984) 116
Appendix C - RFC 822 Extensions for X.400 access 118
Appendix D - Object Identifier Assignment 119
Appendix E - BNF Summary 120
Appendix F - Text format for MCGAM distribution 127
1 - Text Formats 127
2 - Mechanisms to register and to distribute MCGAMs 127
3 - Syntax Definitions 128
4 - Table Lookups 129
5 - Domain -> OR Address MCGAM format 129
6 - OR Address -> Domain MCGAM format 129
7 - Domain -> OR Address of Preferred Gateway table 130
8 - OR Addresss -> domain of Preferred Gateway table 130
Appendix G - Conformance 131
Appendix H - Change History: RFC 987, 1026, 1138, 1148 133
1 - Introduction 133
2 - Service Elements 133
3 - Basic Mappings 133
4 - Addressing 134
5 - Detailed Mappings 134
6 - Appendices 134
Appendix I - Change History: RFC 1148 to RFC 1327 135

1 - General 135
2 - Basic Mappings 135
3 - Addressing 135
4 - Detailed Mappings 135
5 - Appendices 136
Appendix J - Change History: RFC 1327 to this Document
..... 137
1 - General 137
2 - Service Elements 137
3 - Basic Mappings 137
4 - Addressing 137
5 - Detailed Mappings 138
6 - Appendices 138
Appendix L - ASN.1 Summary 139
Security Considerations 141
Author's Address 141
References 141
Full Copyright Statement 144

Chapter 1 -- Overview

1.1. X.400

This document relates primarily to the ITU-T 1988 and 1992 X.400 Series Recommendations / ISO IEC 10021 International Standard. This ISO/ITU-T standard is referred to in this document as "X.400", which is a convenient shorthand. Any reference to the 1984 Recommendations will be explicit. Any mappings relating to elements which are in the 1992 version and not in the 1988 version will be noted explicitly. X.400 defines an Interpersonal Messaging System (IPMS), making use of a store and forward Message Transfer System. This document relates to the IPMS, and not to wider application of X.400, such as EDI as defined in X.435.

1.2. RFC 822 and MIME

RFC 822 evolved as a messaging standard on the DARPA (the US Defense Advanced Research Projects Agency) Internet. RFC 822 specifies an end to end message format, consisting of a header and an unstructured text body. MIME (Multipurpose Internet Mail Extensions) specifies a structured message body format for use with RFC 822. The term "RFC 822" is used in this document to refer to the combination of MIME and RFC 822. RFC 822 and MIME are used in conjunction with a number of different message transfer protocol environments. The core of the MIXER specification is designed to work with any supporting message transfer protocol.

One transfer protocol, SMTP, is of particular importance and is covered in MIXER. On the Internet and other TCP/IP networks, RFC 822 is used in conjunction with RFC 821, also known as Simple Mail

Transfer Protocol (SMTP) [30], in a manner conformant with the host requirements specification [10]. Use of MIXER with SMTP is defined in Appendix A.

1.3. The need for conversion

There is a large community using RFC 822 based protocols for mail services, who will wish to communicate with users of the IPMS provided by X.400 systems. This will also be a requirement in cases where communities intend to make a transition between the different technologies, as conversion will be needed to ensure a smooth service transition. It is expected that there will be more than one gateway, and this specification will enable them to behave in a consistent manner. Note that the term gateway is used to describe a component performing the mapping between RFC 822 and X.400. This is standard usage amongst mail implementors, but differs from that used by transport and network service implementors.

Consistency between gateways is desirable to provide:

1. Consistent service to users.

2. The best service in cases where a message passes through multiple gateways.

1.4. General approach

There are a number of basic principles underlying the details of the specification. These principles are goals, and are not achieved in all aspects of the specification.

1. The specification should be pragmatic. There should not be a requirement for complex mappings for "Academic" reasons. Complex mappings should not be required to support trivial additional functionality.

2. Subject to 1), functionality across a gateway should be as high as possible.

3. It is always a bad idea to lose information as a result of any transformation. Hence, it is a bad idea for a gateway to discard information in the objects it processes. This includes requested services which cannot be fully mapped.

4. Mail gateways operate at a level above the layer on which they perform mappings. This implies that the gateway shall not only be cognisant of the semantics of objects at the gateway level, but also be cognisant of higher level semantics. If meaningful transformation of the objects that the gateway operates on is to occur, then the gateway needs to understand more than the objects themselves.

5. Subject to 1), the mapping should be reversible. That is, a double transformation should bring you back to where you started.

1.5. Gatewaying Model

1.5.1. X.400

X.400 defines the IPMS Abstract Service in X.420 , [11] which comprises of three basic services:

1. Origination

2. Reception

3. Management

Management is a local interaction between the user and the IPMS, and is therefore not relevant to gatewaying. The first two services consist of operations to originate and receive the following two objects:

1. IPM (Interpersonal Message). This has two components: a heading, and a body. The body is structured as a sequence of body parts, which may be basic components (e.g., IA5 text, or G3 fax), or forwarded Interpersonal Messages. The heading consists of fields containing end to end user information, such as subject, primary recipients (To:), and importance.

2. IPN (Inter Personal Notification). A notification about receipt of a given IPM at the UA level.

The Origination service also allows for origination of a probe, which is an object to test whether a given IPM could be correctly received.

The Reception service also allows for receipt of Delivery Reports (DR), which indicate delivery success or failure.

These IPMS Services utilise the Message Transfer System (MTS) Abstract Service [12]. The MTS Abstract Service provides the following three basic services:

1. Submission (used by IPMS Origination)

2. Delivery (used by IPMS Reception)

3. Administration (used by IPMS Management)

Administration is a local issue, and so does not affect this standard. Submission and delivery relate primarily to the MTS Message (comprising Envelope and Content), which carries an IPM or IPN (or other uninterpreted contents). The Envelope includes a message identifier, an originator, and a list of recipients. Submission also includes the probe service, which supports the MTS Probe. Delivery also includes Reports, which indicate whether a given MTS Message has been delivered or not (or for a probe if delivery would have happened).

The MTS is provided by MTAs which interact using the MTA (Message Transfer Agent) Service, which defines the interaction between MTAs, along with the procedures for distributed operation. This service provides for transfer of MTS Messages, Probes, and Reports.

1.5.2. RFC 822

RFC 822 is based on the assumption that there is an underlying service, which is here called the 822-MTS service. The 822-MTS service provides three basic functions:

1. Identification of a list of recipients.

2. Identification of an error return address.

3. Transfer of an RFC 822 message.

It is possible to achieve 2) within the RFC 822 header.

This specification will be used most commonly with SMTP as the 822-MTS service. The core MIXER specification is written so that it does not rely on non-basic 822-MTS services. Use of non-basic SMTP services is described in Appendix A. The core of this document is written using SMTP terminology for 822-MTS services.

An RFC 822 message consists of a header, and content which is uninterpreted ASCII text. The header is divided into fields, which are the protocol elements. Most of these fields are analogous to IPM

heading fields, although some are analogous to MTS Service Elements or MTA Service Elements.

RFC 822 supports delivery status notifications by use of the NOTARY mechanisms [28].

1.5.3. The Gateway

Given this functional description of the two services, the functional nature of a gateway can now be considered. It would be elegant to consider the SMTP (822-MTS) service mapping onto the MTS Service Elements and RFC 822 mapping onto an IPM, but there is a not a clear match between these services. Another elegant approach would be to treat this document as the definition of an X.400 Access Unit (AU). In this case, the abstraction level is too high, and some necessary mapping function is lost. It is necessary to consider that the IPM format definition, the IPMS Service Elements, the MTS Service Elements, and MTA Service Elements on one side are mapped into RFC 822 + SMTP on the other in a slightly tangled manner. The details of the tangle will be made clear in Chapter 5. Access to the MTA Service Elements is minimised.

The following basic mappings are thus defined. When going from RFC 822 to X.400, an RFC 822 message and the associated SMTP information is always mapped into an IPM (MTA, MTS, and IPMS Services) and a Delivery Status Notification is mapped onto a Report. Going from X.400 to RFC 822, an RFC 822 message and the associated SMTP information may be derived from:

1. An IPN (MTA, MTS, and IPMS services)

2. An IPM (MTA, MTS, and IPMS services)

A Report (MTA, and MTS Services) is mapped onto a delivery status notification.

Probes (MTA Service) shall be processed by the gateway, as discussed in Chapter 5. MTS Messages containing Content Types other than those defined by the IPMS are not mapped by the gateway, and shall be rejected at the gateway if no other gatewaying procedure is defined.

This specification is concerned with X.400 IPMS. Future specifications may defined mappings for other X.400 content types.

1.5.4. Repeated Mappings

The primary goal of this specification is to support single mappings, so that X.400 and RFC 822 users can communicate with maximum functionality.

The mappings specified here are designed to work where a message traverses multiple times between X.400 and RFC 822. This is often essential, particularly in the case of distribution lists. However, in general, this will lead to a level of service which is the lowest common denominator (approximately the services offered by RFC 822).

Some RFC 822 networks may wish to use X.400 as an interconnection mechanism (typically for policy reasons), and this is fully supported.

Where an X.400 message transfers to RFC 822 and then back to X.400, there is no expectation of X.400 services which do not have an equivalent service in standard RFC 822 being preserved - although this may be possible in some cases.

1.6. Support of X.400 (1984)

The MIXER definition is based on the initial specification of RFC 987 and in its addendum RFC 1026, which defined a mapping between X.400(1984) and RFC 822. The core MIXER mapping is defined using the full 1988 version of X.400, and not to a 1984 compatible subset. New features of X.400(1988) can be used to provide a much cleaner mapping than that defined in RFC 987. To interwork with 1984 systems, Appendix B shall be followed.

If a message is being transferred to an X.400(1984) system by way of X.400(1988) MTA it will give a slightly better service to follow the rules of Appendix B, than to downgrade without this knowledge. Downgrading specifications which supplement those specified in X.400 (X.419) are given in RFC 1328 [22] and RFC 1496 (HARPOON) [5].

1.7. X.400 (1992)

X.400 (1992) features are not used by the core of this mapping, and so there is not an equivalent downgrade problem.

1.8. MIME

MIME format messages are generated by this mapping. As MIME messages are fully RFC 822 compliant, this will not cause problems with systems which are not MIME capable.

1.9. Body Parts

MIME and X.400 IPMS can both carry arbitrary body parts. MIME defines a mechanism for adding new body parts, and new body parts are registered with the IANA. X.400 defines a mechanism adding new body parts, usually referred to as Body Part 15. Extensions are defined by Object Identifiers, so there is no requirement for a central body part registration authority. The Electronic Messaging Association (EMA) maintains a list of some commonly used body parts. The EMA has specified a mechanism to use the File Transfer Body Part (FTBP) as a more generic means to support message attachments. This approach is gaining widespread commercial support.

The mapping between X.400 and MIME body parts is defined in the companion MIXER specification, referenced here as RFC 2157 [8]. This document is an update of RFC 1494 [6].

Editor's Note:
References to 2157 will be resolved as these two documents are expected to progress in parallel.

These two specifications together form the complete MIXER Mapping.

1.10. Local and Global Scenarios

There are two basic scenarios for X.400/MIME interworking:

Global Scenario

There are two global mail networks (Internet/MIME and X.400), interconnected by multiple gateways. Objects may be transferred over multiple gateways, and so it is important that gateways behave in a coherent fashion. MIXER is critical to support this scenario.

Local Scenario

A gateway is used to connect a closed community to a global mail network (this could be enforced by connectivity or gateway authorisation policy). This is a common commercial scenario. MIXER is useful to support this scenario, as it allows an industry standard provision of service, but this could be supported by something which was MIXER-like.

A solution for the global scenario will work for the local scenario. However, there are aspects of MIXER which have significant implementation or deployment effort (the global mapping is the major one, but there are other details too) which and are needed to support

the global scenario, but are not needed in the local scenario.

Note that the local scenario may be the driving force for most deployments, and support of the global scenario may be an important secondary goal.

There is also a transition effect. Gateways which are initially deployed in a strict local scenario situation start to find themselves in a global scenario. A common case is ADMD provided gateways, which are targeted strictly at the local scenario. In practice they soon start to operate in the global scenario, because of distribution lists and messages exchanged with X.400 users that are not customers of the ADMD. At this point, users are hurt by the restrictions of a local scenario gateway.

Note that conformance to MIXER applies to an instantiation of a gateway, not just an implementation (although clearly it is critical that the implementation is capable of being operated in a conformant manner).

MIXER's conformance target is the global scenario, and the specification of MIXER defines operation in this way.

1.11. Compatibility with previous versions

The changes between this and older versions of the document are given in Appendices H, I and J. These are RFCs 987, 1026, 1138, 1148 and 1327. This document is a revision of RFC 1327 [21]. As far as possible, changes have been made in a compatible fashion.

1.12. Aspects not covered

There have been a number of cases where previous versions of this document were used in a manner which was not intended. This section is to make clear some limitations of scope. In particular, this specification does not specify:

- Extensions of RFC 822 to provide access to all X.400 services

- X.400 user interface definition

These are really coupled. To map the X.400 services, this specification defines a number of extensions to RFC 822. As a side effect, these give the 822 user access to SOME X.400 services. However, the aim on the RFC 822 side is to preserve current service, and it is intentional that access is not given to all X.400 services. Thus, it will be a poor choice for X.400 implementors to use MIXER as

an interface - there are too many aspects of X.400 which cannot be accessed through it. If a text interface is desired, a specification targeted at X.400, without RFC 822 restrictions, would be more appropriate. Some optional and limited extensions in this area have proved useful, and are defined in Appendix C.

1.13. Subsetting

This proposal specifies a mapping which is appropriate to preserve services in existing RFC 822 communities. Implementations and specifications which subset this specification are non-conformant and strongly discouraged.

1.14. Specification Language

ISO and Internet standards have clear definitions as to the style of language used. This specification maps between ISO/ITU-T protocol and Internet protocols. This document uses ISO terminology for the following reasons:

1. This was done in previous versions.

2. ISO language may be mechanically converted to Internet language, but not vice versa.

The key elements of the ISO rules are:

1. All mandatory features shall clearly be indicated by imperative statements or the word "shall" or "shall not".

2. Optional features shall be indicated by the word "may".

3. The word "should" and the phrase "may not" shall not be used.

In some cases the specification issues guidance on use of optional features, by use of the the phrase word "recommended" or "not recommended".

To interpet this document according to Internet rules, replace every occurrence of "shall" with "must".

1.15. Related Specifications

Mappings between Mail-11 and X.400 and Mail-11 and RFC 822 are described in RFC 2162, using mappings related to those defined here [2].

1.16. Document Structure

This document has five chapters:

1. Overview - this chapter.

2. Service Elements - This describes the (end user) services mapped by a gateway.

3. Basic mappings - This describes some basic notation used in Chapters 3-5, the mappings between character sets, and some fundamental protocol elements.

4. Addressing - This considers the mapping between X.400 OR names and RFC 822 addresses, which is a fundamental gateway component.

5. Detailed Mappings - This describes the details of all other mappings.

There are also ten appendices.

WARNING:

> THE REMAINDER OF THIS SPECIFICATION IS TECHNICALLY DETAILED. IT WILL NOT MAKE SENSE, EXCEPT IN THE CONTEXT OF RFC 822 AND X.400 (1988). DO NOT ATTEMPT TO READ THIS DOCUMENT UNLESS YOU ARE FAMILIAR WITH THESE SPECIFICATIONS.

1.17. Acknowledgements

The work in this specification was substantially based on RFC 987 and RFC 1148, which had input from many people, who are credited in the respective documents.

A number of comments from people on RFC 1148 lead to RFC 1327. In particular, there were comments and suggestions from: Maurice Abraham (HP); Harald Alvestrand (Sintef); Peter Cowen (X-Tel); Jim Craigie (JNT); Ella Gardner (MITRE); Christian Huitema (Inria); Erik Huizer (SURFnet); Neil Jones (DEC); Ignacio Martinez (IRIS); Julian Onions (X-Tel); Simon Poole (SWITCH); Clive Roberts (Data General); Pete Vanderbilt (SUN); Alan Young (Concurrent).

RFC 1327 has been widely adopted, and a review team was formed. This comprised of: Urs Eppenberger (SWITCH)(Chair); Claudio Allocchio (INFN); Harald Alvestrand (UNINETT); Dave Crocker (Brandenburg); Ned Freed (Innosoft); Erik Huizer (SURFnet); Steve Kille (Isode); Peter Sylvester (GC Tech).

Harald Alvestrand also supplied the tables mapping DSN status codes with X.400 codes. Ned Freed defined parts of the File Transfer Body Part mapping.

Comment and input has also been received from: Bengt Ackzell (Generic Systems); Samir Albadine (Transpac); Mark Boyes (DEC); Larry Campbell (Boston Software Works); Jacqui Caren (Cray); Allan Cargille (MCI); Kevin Carrosso (Innosoft); Charlie Combs (OIW); Jim Craigie (Net-Tel); Eamon Doyle (Isocor); Efifion Edem (SITA); Jyrki Heikkinen (ICL); Edward Hibbert (DCL); Jeroun Houttin (Terena); Kevin Jordan (CDS); Paul Kingsnorth (DEC); Carl-Uno Manros (Manros Consulting); Suzan Mendes (Telis); Robert Miles (Softswitch); Roger Mizumorri (Enterprise Solutions Ltd); Keith Moore (University of Tennessee); Ruth Moulton (Net-Tel) Michel Musy (Bull); Kenji Nonaka (NTT): The OIW MHSIG; Tom Oliphant (SWITCH); Julian Onions (NEXOR); Jacob Palme (KTH); Olivier Paridaens (ULB); Mary la Roche (Citicorp); John Setsaas (Maxware); Russell Sharpe (DCL); Patrick Soulier (CCETT); Eftimios Tsigros (Universite Libre de Bruxelles); Sean Turner (IECA); Mark Wahl (Isode); David Wilson (Isode); Bill Wohler (Worldtalk); Alan Young (Isode); Alain Zahm (Telis).

Chapter 2 - Service Elements

This chapter considers the services offered across a gateway built according to this specification. It gives a view of the functionality provided by such a gateway for communication with users in the opposite domain. This chapter considers service mappings in the context of SINGLE transfers only, and not repeated mappings through multiple gateways.

2.1. The Notion of Service Across a Gateway

RFC 822 and X.400 provide a number of services to the end user. This chapter describes the extent to which each service can be supported across an X.400 <-> RFC 822 gateway. The cases considered are single transfers across such a gateway, although the problems of multiple crossings are noted where appropriate.

2.1.1. Origination of Messages

When a user originates a message, a number of services are available. Some of these imply actions (e.g., delivery to a recipient), and some are insertion of known data (e.g., specification of a subject field). This chapter describes, for each offered service, to what extent it is supported for a recipient accessed through a gateway. There are three levels of support:

Supported
The corresponding protocol elements map well, and so the service can be fully provided.

Not Supported
The service cannot be provided, as there is a complete mismatch.

Partial Support
The service can be partially fulfilled.

In the first two cases, the service is simply marked as "Supported" or "Not Supported". Some explanation may be given if there are additional implications, or the (non) support is not intuitive. For partial support, the level of partial support is summarised. Where partial support is good, this will be described by a phrase such as "Supported by use of.....". A common case of this is where the service is mapped onto a non-standard service on the other side of the gateway, and this would have lead to support if it had been a standard service. In many cases, this is equivalent to support. For partial support, an indication of the mechanism is given, in order to give a feel for the level of support provided. Note that this is not a replacement for Chapter 5, where the mapping is fully specified.

If a service is described as supported, this implies:

- Semantic correspondence.

- No (significant) loss of information.

- Any actions required by the service element.

An example of a service gaining full support: If an RFC 822 originator specifies a Subject: field, this is considered to be supported, as an X.400 recipient will get a subject indication.

In many cases, the required action will simply be to make the information available to the end user. In other cases, actions may imply generating a delivery report.

All RFC 822 services are supported or partially supported for origination. The implications of non-supported X.400 services is described under X.400.

2.1.2. Reception of Messages

For reception, the list of service elements required to support this mapping is specified. This is really an indication of what a recipient might expect to see in a message which has been remotely

originated.

2.2. RFC 822

RFC 822 does not explicitly define service elements, as distinct from protocol elements. However, all of the RFC 822 header fields, with the exception of trace, can be regarded as corresponding to implicit RFC 822 service elements.

2.2.1. Origination in RFC 822

A mechanism of mapping, used in several cases, is to map the RFC 822 header into a heading extension in the IPM (InterPersonal Message). This can be regarded as partial support, as it makes the information available to any X.400 implementations which are interested in these services. Communities which require significant RFC 822 interworking are recommended to require that their X.400 User Agents are able to display these heading extensions. Support for the various service elements (headers) is now listed.

Date:
: Supported.

From:
: Supported. For messages where there is also a sender field, the mapping is to "Authorising Users Indication", which has subtly different semantics to the general RFC 822 usage of From:.

Sender: Supported.

Reply-To: Supported.

To: Supported.

Cc: Supported.

Bcc: Supported.

Message-Id: Supported.

In-Reply-To:
: Supported, for a single reference. Where multiple references are given, partial support is given by mapping to "Cross Referencing Indication". This gives similar semantics.

References: Supported.

Keywords: Supported by use of a heading extension.

Subject: Supported.

Comments: Supported by use of a heading extension.

Encrypted: Supported by use of a heading extension.

Content-Language: Supported.

Resent-*

Supported by use of a heading extension. Note that addresses in these fields are mapped onto text, and so are not accessible to the X.400 user as addresses. In principle, fuller support would be possible by mapping onto a forwarded IP Message, but this is not suggested.

Other Fields

In particular X-* fields, and "illegal" fields in common usage (e.g., "Fruit-of-the-day:") are supported by use of heading extensions.

MIME introduces a number of headings. Support is defined in RFC 2157.

2.2.2. Reception by RFC 822

This considers reception by an RFC 822 User Agent of a message originated in an X.400 system and transferred across a gateway. The following standard services (headers) may be present in such a message:

Date:

From:

Sender:

Reply-To:

To:

Cc:

Bcc:

Message-Id:

In-Reply-To:

References:

Subject:

Content-Type: (See RFC 2157)

Content-Transfer-Encoding: (See RFC 2157)

MIME-Version: (See RFC 2157)

The following services (headers) may be present in the header of a message. These are defined in more detail in Chapter 5 (5.3.4, 5.3.6, 5.3.7):

Autoforwarded:

Autosubmitted:

X400-Content-Identifier:

Content-Language:

Conversion:

Conversion-With-Loss:

Delivery-Date:

Discarded-X400-IPMS-Extensions:

Discarded-X400-MTS-Extensions:

DL-Expansion-History:

Deferred-Delivery:

Expires:

Importance:

Incomplete-Copy:

Latest-Delivery-Time:

Message-Type:

Original-Encoded-Information-Types:

Originator-Return-Address:

Priority:

Reply-By:

Sensitivity:

Supersedes:

X400-Content-Type:

X400-MTS-Identifier:

X400-Originator:

X400-Received:

X400-Recipients:

2.3. X.400

2.3.1. Origination in X.400

When mapping services from X.400 to RFC 822 which are not supported by RFC 822, new RFC 822 headers are defined, and registered by publication in this standard. It is intended that co-operating RFC 822 systems may also use them. Where these new fields are used, and no system action is implied, the service can be regarded as being partially supported. Chapter 5 describes how to map X.400 services onto these new headers. Other elements are provided, in part, by the gateway as they cannot be provided by RFC 822.

Some service elements are marked N/A (not applicable). There are five cases, which are marked with different comments:

N/A (local)
: These elements are only applicable to User Agent / Message Transfer Agent interaction and so they cannot apply to RFC 822 recipients.

N/A (PDAU)
These service elements are only applicable where the recipient is reached by use of a Physical Delivery Access Unit (PDAU), and so do not need to be mapped by the gateway.

N/A (reception)
These services are only applicable for reception.

N/A (prior)
If requested, this service shall be performed prior to the gateway.

N/A (MS)
These services are only applicable to Message Store (i.e., a local service).

Finally, some service elements are not supported. In particular, the new security services are not mapped onto RFC 822. Unless otherwise indicated, the behaviour of service elements marked as not supported will depend on the criticality marking supplied by the user. If the element is marked as critical for transfer or delivery, a non-delivery notification will be generated. Otherwise, the service request will be ignored.

2.3.1.1. Basic Interpersonal Messaging Service

These are the mandatory IPM services as listed in Section 19.8 of X.400 / ISO/IEC 10021-1, listed here in the order given. Section 19.8 has cross references to short definitions of each service.

Access management
N/A (local).

Content Type Indication
Supported by a new RFC 822 header (X400-Content-Type:).

Converted Indication
Supported by a new RFC 822 header (X400-Received:).

Delivery Time Stamp Indication
N/A (reception).

IP Message Identification
Supported.

Message Identification
Supported, by use of a new RFC 822 header (X400-MTS-Identifier). This new header is required, as X.400 has two message-ids whereas

RFC 822 has only one (see IP Message Identification

Non-delivery Notification
Not supported in all cases. Supported where the recipient system supports NOTARY DSNs. In general all RFC 822 systems will return error reports by use of IP messages. In other service elements, this pragmatic result can be treated as effective support of this service element.

Original Encoded Information Types Indication
Supported as a new RFC 822 header (Original-Encoded-Information-Types:).

Submission Time Stamp Indication
Supported.

Typed Body
Support is defined in RFC 2157.

User Capabilities Registration
N/A (local).

2.3.1.2. IPM Service Optional User Facilities

This section describes support for the optional (user selectable) IPM services as listed in Section 19.9 of X.400 / ISO/IEC 10021- 1, listed here in the order given. Section 19.9 has cross references to short definitions of each service.

Additional Physical Rendition
N/A (PDAU).

Alternate Recipient Allowed
Not supported. There is no RFC 822 service equivalent to prohibition of alternate recipient assignment (e.g., an RFC 822 system may freely send an undeliverable message to a local postmaster). A MIXER gateway has two conformant options. The first is not to gateway a message requesting prohibition of alternate recipient, as this control cannot be guaranteed. This option supports the service, but may cause unacceptable level of message rejections. The second is to gateway the message on the basis that there is no alternate recipient service in RFC 822. RFC 1327 allowed only the second option. If the first option is shown to be operationally effective, it may be the only option in future versions of MIXER.

Authorising User’s Indication
Supported.

Auto-forwarded Indication
Supported as new RFC 822 header (Auto-Forwarded:).

Basic Physical Rendition
N/A (PDAU).

Blind Copy Recipient Indication
Supported.

Body Part Encryption Indication
Supported by use of a new RFC 822 header (Original-Encoded-Information-Types:), although in most cases it will not be possible to map the body part in question.

Content Confidentiality
Not supported.

Content Integrity
Not supported.

Conversion Prohibition
Supported. Operation defined in RFC 2157.

Conversion Prohibition in Case of Loss of Information
Supported. Operation defined in RFC 2157.

Counter Collection
N/A (PDAU).

Counter Collection with Advice
N/A (PDAU).

Cross Referencing Indication
Supported.

Deferred Delivery
N/A (prior). This service shall always be provided by the MTS prior to the gateway. A new RFC 822 header (Deferred-Delivery:) is provided to transfer information on this service to the recipient.

Deferred Delivery Cancellation
N/A (local).

Delivery Notification
Supported. This is performed at the gateway, but may be performed at the end system if the end system supports NOTARY. Thus, a notification is sent by the gateway to the originator.

Delivery via Bureaufax Service
N/A (PDAU).

Designation of Recipient by Directory Name
N/A (local).

Disclosure of Other Recipients
Supported by use of a new RFC 822 header (X400-Recipients:). This is descriptive information for the RFC 822 recipient, and is not reverse mappable.

DL Expansion History Indication
Supported by use of a new RFC 822 header (DL-Expansion-History:).

DL Expansion Prohibited
Distribution List means MTS supported distribution list, in the manner of X.400. This service does not exist in the RFC 822 world, although RFC 822 supports distribution list functionality. There is no SMTP leve control to prohibit distribution list expansion. A MIXER gateway has two conformant options. The first is not to gateway a message requesting DL expansion prohibition, as this control cannot be guaranteed. This option supports the service, but may cause unacceptable level of message rejections. The second is to gateway the message on the basis that there is no distribution list service in RFC 822. RFC 1327 allowed only the second option. If the first option is shown to be operationally effective, it may be the only option in future versions of MIXER.

Express Mail Service
N/A (PDAU).

Expiry Date Indication
Supported as new RFC 822 header (Expires:). In general, no automatic action can be expected.

Explicit Conversion
N/A (prior).

Forwarded IP Message Indication
Supported.

Grade of Delivery Selection
Not Supported. There is no equivalent service in RFC 822.

Importance Indication
Supported as new RFC 822 header (Importance:).

Incomplete Copy Indication
Supported as new RFC 822 header (Incomplete-Copy:).

Language Indication
Supported as new RFC 822 header (Content-Language:).

Latest Delivery Designation
Not supported. A new RFC 822 header (Latest-Delivery-Time:) is provided, which may be used by the recipient for general information, but will not be acted on by the SMTP infrastrucuture.

Message Flow Confidentiality
Not supported.

Message Origin Authentication
N/A (reception).

Message Security Labelling
Not supported.

Message Sequence Integrity
Not supported.

Multi-Destination Delivery Supported.

Multi-part Body
Supported.

Non Receipt Notification Request
Not supported.

Non Repudiation of Delivery
Not supported.

Non Repudiation of Origin
N/A (reception).

Non Repudiation of Submission
N/A (local).

Obsoleting Indication
Supported as new RFC 822 header (Supersedes:).

Ordinary Mail
N/A (PDAU).

Originator Indication
Supported.

Originator Requested Alternate Recipient
Not supported, but is placed as comment next to address (X400-Recipients:).

Physical Delivery Notification by MHS
N/A (PDAU).

Physical Delivery Notification by PDS
N/A (PDAU).

Physical Forwarding Allowed
Supported by use of a comment in a new RFC 822 header (X400-Recipients:), associated with the recipient in question.

Physical Forwarding Prohibited
Supported by use of a comment in a new RFC 822 header (X400-Recipients:), associated with the recipient in question.

Prevention of Non-delivery notification
Supported where SMTP and NOTARY are available. In other cases formally supported, as delivery notifications cannot be generated by RFC 822. In practice, errors will be returned as IP Messages, and so this service may appear not to be supported (see Non-delivery Notification).

Primary and Copy Recipients Indication
Supported

Probe
Supported at the gateway (i.e., the gateway services the probe).

Probe Origin Authentication
N/A (reception).

Proof of Delivery
Not supported.

Proof of Submission
N/A (local).

Receipt Notification Request Indication
Not supported.

Redirection Disallowed by Originator
Redirection means MTS supported redirection, in the manner of X.400. This service does not exist in the RFC 822 world. RFC 822 redirection (e.g., aliasing) is regarded as an informal redirection mechanism, beyond the scope of this control. Messages will be sent to RFC 822, irrespective of whether this service is requested. In practice, control of this service is not supported.

Registered Mail
N/A (PDAU).

Registered Mail to Addressee in Person
N/A (PDAU).

Reply Request Indication
Supported as comment next to address.

Replying IP Message Indication
Supported.

Report Origin Authentication
N/A (reception).

Request for Forwarding Address
N/A (PDAU).

Requested Delivery Method
N/A (local). The service request is dealt with at submission time. Any such request is made available through the gateway by use of a comment associated with the recipient in question.

Return of Content
Supported where SMTP and NOTARY are used. In principle for other situations, this is N/A, as non-delivery notifications are not supported. In practice, most RFC 822 systems will return part or all of the content along with the IP Message indicating an error (see Non-delivery Notification).

Sensitivity Indication
Supported as new RFC 822 header (Sensitivity:).

Special Delivery
N/A (PDAU).

Stored Message Deletion
N/A (MS).

Stored Message Fetching
N/A (MS).

Stored Message Listing
N/A (MS).

Stored Message Summary
N/A (MS).

Subject Indication
Supported.

Undeliverable Mail with Return of Physical Message
N/A (PDAU).

Use of Distribution List
In principle this applies only to X.400 supported distribution lists (see DL Expansion Prohibited). Theoretically, this service is N/A (prior). In practice, because of informal RFC 822 lists, this service can be regarded as supported.

Auto-Submitted Indication
Supported

2.3.2. Reception by X.400

2.3.2.1. Standard Mandatory Services

The following standard IPM mandatory user facilities are required for reception of RFC 822 originated mail by an X.400 UA.

Content Type Indication

Delivery Time Stamp Indication

IP Message Identification

Message Identification

Non-delivery Notification

Original Encoded Information Types Indication

Submission Time Stamp Indication

Typed Body

2.3.2.2. Standard Optional Services

The following standard IPM optional user facilities are required for reception of RFC 822 originated mail by an X.400 UA.

Authorising User's Indication

Blind Copy Recipient Indication

Cross Referencing Indication

Originator Indication

Primary and Copy Recipients Indication

Replying IP Message Indication

Subject Indication

2.3.2.3. New Services

A new X.400 service "RFC 822 Header Field" is defined using the extension facilities. This allows for any RFC 822 header field to be represented. It may be present in RFC 822 originated messages which are received by an X.400 UA.

Chapter 3 Basic Mappings

3.1. Notation

The X.400 protocols are encoded in a structured manner according to ASN.1, whereas RFC 822 is text encoded. To define a detailed mapping, it is necessary to refer to detailed protocol elements in each format. A notation to achieve this is described in this section.

3.1.1. RFC 822

Structured text is defined according to the Extended Backus Naur Form (EBNF) defined in Section 2 of RFC 822 [16]. In the EBNF definitions used in this specification, the syntax rules given in Appendix D of RFC 822 are assumed. When these EBNF tokens are referred to outside an EBNF definition, they are identified by the string "822." appended to the beginning of the string (e.g., 822.addr-spec). Additional syntax rules, to be used throughout this specification, are defined in this chapter.

The EBNF is used in two ways.

1. To describe components of RFC 822 messages (or of SMTP components). When these new EBNF tokens are referred to outside an EBNF definition, they are identified by the string "EBNF." appended to the beginning of the string (e.g., EBNF.importance).

2. To describe the structure of IA5 or ASCII information not in an RFC 822 message.

For all new EBNF, tokens will either be self delimiting, or be delimited by self delimiting tokens. Comments and LWSP are not used as delimiters, except for the following cases, where LWSP may be inserted according to RFC 822 rules.

- Around the ":" in all headers

- EBNF.labelled-integer

- EBNF.object-identifier

- EBNF.encoded-info

RFC 822 folding rules are applied to all headers. Comments are never used in these new headers.

This notation is used in a modified form to refer to NOTARY EBNF [28]. For this EBNF, the keyword EBNF it replaces with DSN, for example DSN.final-recipient-field fields.

3.1.2. ASN.1

An element is referred to with the following syntax, defined in EBNF:

```
element         = service "." definition *( "." definition )
service         = "IPMS" / "MTS" / "MTA"
definition      = identifier / context
identifier      = ALPHA *< ALPHA or DIGIT or "-" >
context         = "[" 1*DIGIT "]"
```

The EBNF.service keys are shorthand for the following service specifications:

IPMS IPMSInformationObjects defined in Annex E of X.420 / ISO 10021-7.

MTS MTSAbstractService defined in Section 9 of X.411 / ISO 10021-4.

TA MTAAbstractService defined in Section 13 of X.411 / ISO 10021-4.

FTBP File Transfer Body Part, as defined in [27].

The first EBNF.identifier identifies a type or value key in the context of the defined service specification. Subsequent EBNF.identifiers identify a value label or type in the context of the first identifier (SET or SEQUENCE). EBNF.context indicates a context tag, and is used where there is no label or type to uniquely identify a component. The special EBNF.identifier keyword "value" is used to denote an element of a sequence. For example, IPMS.Heading.subject defines the subject element of the IPMS heading. The same syntax is also used to refer to element values. For example, MTS.EncodedInformationTypes.[0].g3Fax refers to a value of MTS.EncodedInformationTypes.[0] .

3.2. ASCII and IA5

A gateway will interpret all IA5 as ASCII. Thus, mapping between these forms is conceptual.

3.3. Standard Types

There is a need to convert between ASCII text and some of the types defined in ASN.1 [14]. For each case, an EBNF syntax definition is given, for use in all of this specification, which leads to a mapping between ASN.1, and an EBNF construct. All EBNF syntax definitions of ASN.1 types are in lower case, whereas ASN.1 types are referred to with the first letter in upper case. Except as noted, all mappings are symmetrical.

3.3.1. Boolean

Boolean is encoded as:

```
boolean = "TRUE" / "FALSE"
```

3.3.2. NumericString

NumericString is encoded as:

```
numericstring = *(DIGIT / " ")
```

3.3.3. PrintableString

PrintableString is a restricted IA5String defined as:

```
printablestring  = *( ps-char )
ps-restricted-char      = 1DIGIT /  1ALPHA / " " / "'" / "+"
                    / "," / "-" / "." / "/" / ":" / "=" / "?"
ps-delim         = "(" / ")"
ps-char          = ps-delim / ps-restricted-char
```

This can be used to represent real printable strings in EBNF.

3.3.4. T.61String

In cases where T.61 strings are only used for conveying human interpreted information, the aim of a mapping is to render the characters appropriately in the remote character set, rather than to maximise reversibility. For these cases, there are two options, both of which are conformant to this specification:

1. The mappings to IA5 defined in ITU-T Recommendation X.408 (1988) may be used [13]. These will then be encoded in ASCII. This is the approach mandated in RFC 1327.

2. This mapping may be used if the characters are not contained within ASCII repertoire, but are all in an IANA-registered character set. Use the encoding defined in RFC 1522 [9] to generate appropriate encoded-words. If this mapping is used, the character set ISO-8859-1 shall be used if all of the characters needed are available in this repertoire. In other cases, the character set TELETEX shall be used. The details of this character set is defined in the Appendix C of RFC 2157.

There is also a need to represent Teletex Strings in ASCII, for some aspects of OR Address. For these, the following encoding is used:

```
teletex-string   = *( ps-char / t61-encoded )
t61-encoded      = "{" 1* t61-encoded-char "}"
t61-encoded-char = 3DIGIT
```

Characters in EBNF.ps-char are mapped simply. Other octets, including control characters, are mapped using a quoting mechanism similar to the printable string mechanism. Each octet is represented as 3 decimal digits. For example, the Yen character (hex A5) is represented as {165}. As the three character string, a, yen character, b, would be represented as either "a{165}b".

The use of escape sequences follows that set down for ASN1. in ISO 8825-1, with the additional specifiction that the default G1 page is ISO Latin 1. The page settings may be changed by escape sequences. Changes of the settings hold within a pair of curly brackets ({}), and the settings revert to the default after the right bracket (}) (i.e., they do not carry forward to subsequent T.61 encoding).

There are a number of places where a string may have a Teletex and/or Printable String representation. The following EBNF is used to represent this.

```
teletex-and-or-ps = [ printablestring ] [ "*" teletex-string ]
```

The natural mapping is restricted to EBNF.ps-char, in order to make the full BNF easier to parse. An example is:

```
"yen*{165}"
```

3.3.5. UTCTime

Both UTCTime and the RFC 822 822.date-time syntax contain: Year, Month, Day of Month, hour, minute, second (optional), and Timezone (technically a time differential in UTCTime). 822.date-time also contains an optional day of the week, but this is redundant. With the exception of Year, a symmetrical mapping can be made between these constructs.

Note:
In practice, a gateway will need to parse various illegal variants on 822.date-time. In cases where 822.date-time cannot be parsed, it is recommended that the derived UTCTime is set to the value at the time of translation. Such errors may be noted in an RFC 822 comment, to aid detection and correction.

When mapping to X.400, the UTCTime format which specifies the timezone offset shall be used.

When mapping to RFC 822, the 822.date-time format shall include a numeric timezone offset (e.g., -0500).

When mapping time values, the timezone shall be preserved as specified. The date shall not be normalised to any other timezone.

RFC 822, as modified by RFC 1123, requires use of a four digit year. Note that the original RFC 822 uses a two digit date, which is no longer legal. UTCTime uses a two digit date. To map a year from RFC 822 to X.400, simply use the last two digits. To map a year from X.400 to RFC 822, assume that the two digit year refers to a year in the 10 year epoch 1980-2079.

3.3.6. Integer

A basic ASN.1 Integer will be mapped onto EBNF.numericstring. In many cases ASN.1 will enumerate Integer values or use ENUMERATED. An EBNF encoding labelled-integer is provided. When mapping from EBNF to ASN.1, only the integer value is mapped, and the associated text is discarded. When mapping from ASN.1 to EBNF, a text label may be added. It is recommended that this is done wherever possible and that clear text labels are chosen.

A second encoding labelled-integer-2 is provided. This is used in DSNs, where the parsing rules will treat the text as a comment. This definition was not present in RFC 1327.

```
labelled-integer ::= [ key-string ] "(" numericstring ")"

labelled-integer-2 ::= [ numericstring ] "(" key-string ")"

key-string      = *key-char
key-char        = <a-z, A-Z, 0-9, and "-">
```

3.3.7. Object Identifier

Object identifiers are represented in a form similar to that given in ASN.1. The order is the same as for ASN.1 (big-endian). The numbers are mandatory, and used when mapping from the ASCII to ASN.1. The key-strings are optional. It is recommended that as many strings as possible are generated when mapping from ASN.1 to ASCII, to facilitate user recognition.

```
object-identifier  ::= oid-comp object-identifier
                     | oid-comp

oid-comp ::= [ key-string ] "(" numericstring ")"
```

An example representation of an object identifier is:

joint-iso-ccitt(2) mhs (6) ipms (1) ep (11) ia5-text (0)

or

(2) (6) (1)(11)(0)

Because of the use of brackets and the conflict with the RFC 822 comment convention, MIXER is defines so that the EBNFobject-identifier definition is not used in structured fields.

3.4. Encoding ASCII in Printable String

Some information in RFC 822 is represented in ASCII, and needs to be mapped into X.400 elements encoded as printable string. For this reason, a mechanism to represent ASCII encoded as PrintableString is needed.

A structured subset of EBNF.printablestring is now defined. This shall be used to encode ASCII in the PrintableString character set.

```
ps-encoded       = *( ps-restricted-char / ps-encoded-char )
ps-encoded-char  = "(a)"               ; (@)
                 / "(p)"               ; (%)
                 / "(b)"               ; (!)
                 / "(q)"               ; (")
                 / "(u)"               ; (_)
                 / "(l)"               ; "("
                 / "(r)"               ; ")"
                 / "(" 3DIGIT ")"
```

The 822.3DIGIT in EBNF.ps-encoded-char shall have range 0-127, and is interpreted in decimal as the corresponding ASCII character. Special encodings are given for: at sign (@), percent (%), exclamation mark/bang (!), double quote ("), underscore (_), left bracket ((), and right bracket ()). These characters, with the exception of round brackets, are not included in PrintableString, but are common in RFC 822 addresses. The abbreviations will ease specification of RFC 822 addresses from an X.400 system. These special encodings shall be interpreted in a case insensitive manner, but always generated in lower case.

A reversible mapping between PrintableString and ASCII can now be defined. The reversibility means that some values of printable string (containing round braces) cannot be generated from ASCII. Therefore, this mapping shall only be used in cases where the printable strings have been derived from ASCII (and will therefore

have a restricted domain). For example, in this specification, it is only applied to a Domain Defined Attribute which will have been generated by use of this specification and a value such as "(" would not be possible.

To encode ASCII as PrintableString, the EBNF.ps-encoded syntax is used, with all EBNF.ps-restricted-char mapped directly. All other 822.CHAR are encoded as EBNF.ps-encoded-char.

To encode PrintableString as ASCII, parse PrintableString as EBNF.ps-encoded, and then reverse the previous mapping. If the PrintableString cannot be parsed, then the mapping is being applied in to an inappropriate value, and an error shall be given to the procedure doing the mapping. In some cases, it may be preferable to pass the printable string through unaltered.

Some examples are now given. Note the arrows which indicate asymmetrical mappings:

PrintableString		ASCII
'a demo.'	<->	'a demo.'
foo(a)bar	<->	foo@bar
(q)(u)(p)(q)	<->	"_%"
(a)	<->	@
(A)	->	@
(l)a(r)	<->	(a)
(126)	<->	~
(	->	(
(l)	<->	(

3.5. RFC 1522

RFC 1522 defines a mechanism for encoding other character set information into elements of RFC 822 Headers. A gateway may ignore this encoding and treat the elements as ASCII.

A preferred approach is for the gateway to interpret the RFC 1522 encoding. This will not always be straightforward, because:

1. RFC 1522 permits an openly extensible character set choice, which may be broader than T.61.

2. It is not always possible to map all characters into the equivalent X.400 field.

RFC 1522 is only applied to fields which are "for information only". A gateway which interprets header elements according to RFC 1522 may

apply reasonable heuristics to minimise information loss.

Chapter 4 - Addressing and Message IDs

Addressing is the most complex aspect of X.400 <-> RFC 822 gateway and is therefore given a separate chapter. This chapter also discusses message identifiers, as they are closely linked to addresses. This chapter, as a side effect, also defines a textual representation of an X.400 OR Address. This specification has much similarity to the X.400(92) representation of addresses. This was because early versions of this specification were a major input to this work. This specification retains compatibility with earlier versions. The X.400 specification of address representation can be parsed but is not generated.

Initially we consider an address in the (human) mail user sense of "what is typed at the mailsystem to reference a mail user". A basic RFC 822 address is defined by the EBNF EBNF.822-address:

```
822-address     = [ route ] addr-spec
```

These definitions are taken from RFC 822. In SMTP (or another 822-MTS protocol), the originator and each recipient are considered to be defined by such a construct. In an RFC 822 header, the EBNF.822-address is encapsulated in the 822-address syntax rule, and there may also be associated comments. None of this extra information has any semantics, other than to the end user.

The basic X.400 OR Address, used by the MTS for routing, is defined by MTS.ORAddress. In IPMS, the MTS.ORAddress is encapsulated within IPMS.ORDescriptor.

The RFC 822 822.address is mapped with IPMS.ORDescriptor, and that RFC 822 EBNF.822-address is mapped with MTS.ORAddress.

Section 4.1 defines a textual representation of an OR Address, which is used throughout the rest of this specification. This text representation is designed to represent an X.400 address in the LHS (left hand side) or local part of an RFC 822 address, and so this representation gives a mechanism to represent X.400 addresses within RFC 822 addresses.

Section 4.2 describes global equivalence mapping between parts of the X.400 and RFC 822 name spaces, and defines the concept of a MIXER Conformant Global Address Mapping (MCGAM). Gateways conforming to this specification shall support MCGAMs.

Section 4.3 is the core part of this chapter, and defines the mapping mechanism.

4.1. A textual representation of MTS.ORAddress

MTS.ORAddress is structured as an ordered set of attributes (type/value pairs). It is clearly necessary to be able to encode this in ASCII for gatewaying purposes. All components shall be encoded, in order to guarantee return of error messages, and to optimise third party replies.

4.1.1. Basic OR Address Representation

An OR Address has a number of structured and unstructured attributes. For each unstructured attribute, a key and an encoding is specified. For structured attributes, the X.400 attribute is mapped onto one or more attribute value pairs. For domain defined attributes, each element of the sequence will be mapped onto a triple (key and two values), with each value having the same encoding. The attributes are as follows, with 1984 attributes given in the first part of the attribute key table. For each attribute, a reference is given, consisting of the relevant sections in X.402 / ISO 10021-2, and the extension identifier for 88 only attributes. The attribute key table follows:

Attribute (Component)	Key	Enc	Ref	Id
84/88 Attributes				
MTS.CountryName	C		P	18.3.3
MTS.AdministrationDomainName	ADMD		P	18.3.1
MTS.PrivateDomainName	PRMD		P	18.3.21
MTS.NetworkAddress	X121		N	18.3.7
MTS.TerminalIdentifier	T-ID		P	18.3.23
MTS.OrganizationName	O		P/T	18.3.9
MTS.OrganizationalUnitNames.value	OU		P/T	18.3.10
MTS.NumericUserIdentifier	UA-ID		N	18.3.8
MTS.PersonalName	PN		P/T	18.3.12
MTS.PersonalName.surname	S		P/T	18.3.12
MTS.PersonalName.given-name	G		P/T	18.3.12
MTS.PersonalName.initials	I		P/T	18.3.12
MTS.PersonalName .generation-qualifier	GQ		P/T	18.3.12
MTS.DomainDefineAttribute.value	DD		P/T	18.1

88 Attributes

MTS.CommonName	CN	P/T	18.3.2	1
MTS.TeletexCommonName	CN	P/T	18.3.2	2
MTS.TeletexOrganizationName	O	P/T	18.3.9	3
MTS.TeletexPersonalName	PN	P/T	18.3.12	4
MTS.TeletexPersonalName.surname	S	P/T	18.3.12	4
MTS.TeletexPersonalName.given-name	G	P/T	18.3.12	4
MTS.TeletexPersonalName.initials	I	P/T	18.3.12	4
MTS.TeletexPersonalName .generation-qualifier	GQ	P/T	18.3.12	4
MTS.TeletexOrganizationalUnitNames .value	OU	P/T	18.3.10	5
MTS.TeletexDomainDefinedAttribute .value	DD	P/T	18.1	6
MTS.PDSName	PD-SERVICE	P	18.3.11	7
MTS.PhysicalDeliveryCountryName	PD-C	P	18.3.13	8
MTS.PostalCode	PD-CODE	P	18.3.19	9
MTS.PhysicalDeliveryOfficeName	PD-OFFICE	P/T	18.3.14	10
MTS.PhysicalDeliveryOfficeNumber	PD-OFFICE-NUM	P/T	18.3.15	11
MTS.ExtensionORAddressComponents	PD-EXT-ADDRESS	P/T	18.3.4	12
MTS.PhysicalDeliveryPersonName	PD-PN	P/T	18.3.17	13
MTS.PhysicalDeliveryOrganizationName	PD-O	P/T	18.3.16	14
MTS.ExtensionPhysicalDelivery AddressComponents	PD-EXT-DELIVERY	P/T	18.3.5	15
MTS.UnformattedPostalAddress	PD-ADDRESS	UPA	18.3.25	16
MTS.StreetAddress	PD-STREET	P/T	18.3.22	17
MTS.PostOfficeBoxAddress	PD-BOX	P/T	18.3.18	18
MTS.PosteRestanteAddress	PD-RESTANTE	P/T	18.3.20	19
MTS.UniquePostalName	PD-UNIQUE	P/T	18.3.26	20
MTS.LocalPostalAttributes	PD-LOCAL	P/T	18.3.6	21
MTS.ExtendedNetworkAddress .e163-4-address.number	NET-NUM	N	18.3.7	22
MTS.ExtendedNetworkAddress .e163-4-address.sub-address	NET-SUB	N	18.3.7	22
MTS.ExtendedNetworkAddress .psap-address	NET-PSAP	X	18.3.7	22
MTS.TerminalType	T-TY	I	18.3.24	23

The following keys identify different EBNF encodings, which are associated with the ASCII representation of MTS.ORAddress.

Key	Encoding
P	printablestring
N	numericstring
T	teletex-string
P/T	teletex-and-or-ps
UPA	upa-string
I	labelled-integer
X	presentation-address

The EBNF for presentation-address is taken from the specification RFC 1278 "A String Encoding of Presentation Address" [23].

In most cases, the EBNF encoding maps directly to the ASN.1 encoding of the attribute. There are a few exceptions. In cases where an attribute can be encoded as either a PrintableString or NumericString (Country, ADMD, PRMD), either form is mapped into the EBNF. When generating ASN.1, the NumericString encoding shall be used if the string contains digits and only digits.

There are a number of cases where the P/T (teletex-and-or-ps) representation is used. Where the key maps to a single attribute, this choice is reflected in the encoding of the attribute (attributes 10-21). For example:

```
/CN=yen*{165}/
```

For most of the 1984 attributes and common name, there is a printablestring and a teletex variant. This pair of attributes is mapped onto the single component here. This will give a clean mapping for the common cases where only one form of the name is used. If there is teletex attribute or teletex component only, and it contains only characters in the printable string character set, it shall be represented in the EBNF as if it had been encoded as printable string. A single printable string representation shall also be done when both forms are present and they have the same printable string representation.

The Unformatted Postal Address has a slightly more complex mapping onto a variant of (teletex-and-or-ps), defined as:

```
upa-string = [ printable-upa ] [ "*" teletex-string ]
printable-upa = printablestring *( "|" printablestring )
```

The optional teletex part is straightforward. There is an (optional) sequence of printable strings which are mapped in order. For example:

/PD-ADDRESS=The Dome|The Square|Richmond|England/

X.400 (1992) has introduced a string representation of OR Addresses (see F.401, Annex B). This has specified a number of string keywords for attributes. As earlier versions of this specification were an input to this work, many of the keywords are the same. To increase compatibility, the following alternative values shall be recognised when mapping from RFC 822 to X.400. These shall not be generated when mapping from X.400 to RFC 822. The following keyword alternative table and the subsequent paragraph lists alternative keywords.

Keyword	Alternative
ADMD	A
PRMD	P
GQ	Q
X121	X.121
UA-ID	N-ID
PD-OFFICE-NUM	PD-OFFICE NUMBER
PD-OFFICE-NUM	PD-OFN
PD-EXT-ADDRESS	PD-EA
PD-EXT-DELIVERY	PD-ED
PD-OFFICE	PD-OF
PD-STREET	PD-S
PD-UNIQUE	PD-U
PD-LOCAL	PD-L
PD-RESTANTE	PD-R
PD-BOX	PD-B
PD-CODE	PD-PC
PD-SERVICE	PD-SN
DD	DDA
NET-NUM	E.164
NET-PSAP	PSAP
PD-ADDRESS	PD-A

When mapping from RFC 822 to X.400, the keywords defined in this paragraph shall be recognized. The ordered keywords: OU1, OU2, OU3, and OU4, shall be recognised. If these are present, no keyword OU shall be present. These will be treated as ordered values of OU. PD-A1, PD-A2, PD-A3, PD-A4, PD-A5, PD-A6 shall be treated as ordered lines. If present, these will be assembled with separating line feeds to form a single physical address. In

this case PD-ADDRESS (or PD-A) shall not be present. Similarly, there are ordered keywords for domain defined attributes: DD1, DD2, DD3, DD4,

If ISDN is present, it may be interpreted as an E.163/164 address, using local heuristics to parse the string. X.400 defines the key, but does not give an interpretation of the value.

For T-TY (Terminal Type), the X.400 recommended values are preferred, but other values are allowed. These values are: tlx (3); ttx (4); g3fax (5); g4fax (6); ia5 (7); and vtx (8).

4.1.2. Encoding of Personal Name

Handling of Personal Name and Teletex Personal Name is a common requirement. Therefore MIXER defines an alternative to the EBNF.standard-type syntax, which utilises the "human" conventions for encoding these components. A syntax is defined, which is designed to provide a clean encoding for the common cases of OR Address specification where:

1. There is no generational qualifier

2. Initials, if present, contain only letters

3. Given Name, if present, does not contain full stop ("."), and is at least two characters long.

4. Surname does not contain full stop in the first two characters.

5 If Surname is the only component, it does not contain full stop.

The following EBNF is defined:

```
encoded-pn      = [ given "." ] *( initial "." ) surname

given           = 2*<ps-char not including ".">

initial         = ALPHA

surname         = printablestring
```

This is used to map from any string containing only printable string characters to an OR address personal name. To map from a string to OR Address components, parse the string according to the EBNF. The given name and surname are assigned directly. All EBNF.initial tokens are concatenated without intervening full stops to generate the initials component.

For an OR address which follows the above restrictions, a string is derived in the natural manner. In this case, the mapping will be reversible.

For example:

```
GivenName       = "Marshall"
Surname         = "Rose"

Maps with  "Marshall.Rose"

Initials        = "MT"
Surname         = "Rose"

Maps with  "M.T.Rose"

GivenName       = "Marshall"
Initials        = "MT"
Surname         = "Rose"

Maps with  "Marshall.M.T.Rose"
```

Note that X.400 suggests that Initials is used to encode all initials except the surname (X.402 section 18.3.12). Therefore, the defined encoding is "natural" when either GivenName or Initials, but not both, are present. The case where both are present can be encoded.

4.1.3. Standard Encoding of MTS.ORAddress

Given this structure, we can specify an EBNF representation of an OR Address. The output format of addresses is defined by EBNF.std-or-address. The more flexible input format is defined by EBNF.std-or-address-input. The input EBNF has been added subsequent to RFC 1327, to reflect the formal incorporation of a number of heuristics. The address element separator on input may be "/", ";", or a mixture of these. The output format is used in all examples.

```
std-or-address  = 1*( "/" attribute "=" value ) "/"
attribute       = standard-type
                / "RFC-822"
                / dd-key "." std-printablestring
```

```
std-or-address-input =  [ sep pair ] sep  pair *( sep pair )
                        sep  [ pair sep ]

sep             = "/" / ";"
pair            = input-attribute "=" value
input-attribute = attribute
                / dd-key ":" std-printablestring

standard-type   = key-string

dd-key          = key-string

value           = std-printablestring

std-printablestring
                = *( std-char / std-pair )

std-char        = <"{", "}", "*", and any ps-char
                                  except "/" and "=" >
std-pair        = "$" ps-char
```

For address generation, the standard-type is any key defined in the key table in Section 4.1, except PN, and DD. For address parsing, other key values from Section 4.1 are also valid. The EBNF leads to a set of attribute/value pairs. The value is interpreted according to the EBNF encoding defined in the table.

If the standard-type is PN, the value is interpreted according to EBNF.encoded-pn, and the components of MTS.PersonalName and/or MTS.TeletexPersonalName derived accordingly.

If dd-key is the recognised Domain Defined string (DD) or one of the alternatives defined in Section 4.1, then the type and value are interpreted according to the syntax implied from the encoding, and aligned to either the teletex or printable string form. Key and value shall have the same encoding.

If value is "RFC-822", then the (printable string) Domain Defined Type of "RFC-822" is assumed. This is an optimised encoding of the domain defined type defined by this specification.

The matching of all keywords shall be done in a case-independent manner.

EBNF.std-or-address uses the characters "/" and "=" as delimiters. Domain Defined Attributes and any value may contain these characters. A quoting mechanism, using the non-printable string "$" is used to allow these characters to be represented.

If an address of this syntax is parsed, and a country value is present, but no ADMD, the string shall be interpreted as if an ADMD value of single space had been specified.

4.2. Global Address Mapping

From a user perspective, the ideal mapping would be entirely symmetrical and global, to enable addresses to be referred to transparently in the remote system, with the choice of gateway being left to the Message Transfer Service. There are two fundamental reasons why this is not possible:

1. The syntaxes are sufficiently different to make this impossible.

2 There is insufficient administrative co-operation between the X.400 and RFC 822 name registration authorities for this to work.

Another way to view this situation is to see that there is not a full global equivalence between X.400 and RFC 822 addressing. To meet user needs to the extent possible, this specification provides for equivalence where there is sufficient co-operation. To be useful, this equivalence shall be recognised and interpreted in the same way by all gateways. Therefore, an asymmetrical mapping is defined, which can be symmetrical where there is appropriate administrative co-operation. Section 4.3 describes the asymetrical aspects. This section describes a mechanism to enable the administrative co-ordination for symmetrical mappings.

In order to achieve a symmetrical mapping there is a need to define an administrative equivalence between parts of the OR Address and Domain namespaces. Previous version of this specification did this by definition of a global set of mappings. MIXER defines the concept of a MIXER Conformant Global Address Mapping (MCGAM). This acronym is defined so that it is very clear what is being referenced.

The X.400 and Internet Mail address spaces are hierarchical. It is possible to define an equivalence between two points in the hierarchies, such that addresses below that point can be derived in an algorithmic manner. An MCGAM is a mapping from a point in one hierarchy to a point in the other hierarchy. An "MGGAM pair" is a pair of symmetrical mappings between two points. To define an MCGAM, the following shall apply:

1. The authority defining the MCGAM shall have responsibility for BOTH of the namespaces between which the MCGAM is defined.

2. The authority defining the MCGAM is responsible to ensure that addresses allocated below the two equivalence points conform to the rules set out below.

3. The authority defining the MCGAM is responsible to ensure that addresses which are generated according to the MCGAM are routed correctly.

In general, MCGAMs will be independent. In some cases, a set of MCGAMs may be related (e.g., where one MCGAM defines a mapping for an organization and a second MCGAM defines an excpetion for a subtree within the organization). In this case, the related set of MCGAMs shall be treated as a single MCGAM for distribution purposes.

The existence of an MCGAM does not imply routability and access for all users.

The authority defining an MCGAM may simply use this mapping locally. This will often be the case in a "local scenario" gateway. Because of third party addressing, a MIXER gateway will work best with the maximum number of MCGAMs. Therefore, three mechanisms are defined to enable publication and exchange of MCGAMs:

1. Distribution of text tables. This is described in Appendix F of this specification.

2. Distribution by Domain Name Service. This is described in RFC 2163 [3].

3. Distribution by X.500 Directory Service. This is defined in RFC 2164 [26].

The following sections define how the MCGAM namespace equivalence is modelled. The Internet Domain Namespace defines a simple hierarchy. For the purposes of this mapping, only parts of the namespace where domains conform to the EBNF domain-syntax are allowed.

```
domain-syntax   = alphanum [ *alphanumhyphen alphanum ]
alphanum        = <ALPHA or DIGIT>
alphanumhyphen  = <ALPHA or DIGIT or HYPHEN>
```

Although RFC 822 allows for a more general syntax, this restricted syntax is used in MIXER as it is the one chosen by the various domain service administrations. In practice, it reflects all RFC 822 usage.

The following OR Address attributes are considered as a hierarchy, and may be specified by the domain. They are (in order of the hierarchy defined by MIXER):

Country, ADMD, PRMD, Organization, Organizational Units

There may be up to four ordered Organizational Units. This hierarchy reflects most usage of X.400, although X.400 may be used in other ways. In particular, it covers the Mnemonic OR Address using a 1984 compatible encoding. This is seen as the dominant form of OR Address. MCGAMs may only be used when this hierarchy applies.

An equivalence mapping is defined between two nodes in the respective hierarchies. For example:

=> "AC.UK" might be mapped with
PRMD="UK.AC", ADMD="GOLD 400", C="GB"

The mapping identifies that the management of these points in the respective hierarchies is the same (or co-operate very closely). The equivalence means that the namespaces below this equivalence point map 1:1, except where the mapping is overridden by further equivalence mappings lower down the hierarchy. This equivalence may be achieved in three ways:

1. All of the nodes below this point are RFC 822, and the MIXER mapping defines the X.400 addresses for these nodes.

2. All of the nodes below this point are X.400, and the MIXER mapping defines the RFC 822 addresses for these nodes.

3. There are X.400 and RFC 822 nodes below this point, and addressing is managed in a manner which ensures the equivalence. The rules to achieve this are defined by MIXER.

Each of these ways gives a framework for MCGAM definition.

When an MCGAM is defined, a systematic mapping for the inferior nodes in the two hierarchies follows. This is a 1:1 mapping between the nodes in the subtrees. For example, given the MCGAM pair defined above:

the domain "R-D.Salford.AC.UK" algorithmically maps with
OU="R-D", O="Salford", PRMD="UK.AC", ADMD="GOLD 400", C="GB"

Note that when an equivalence is defined, that this can be re-defined for lower points in the hierarchy. However, it is not possible to declare contained subtrees to be un-mappable.

The equivalence mapping also provides a mechanism to deal with missing elements in the X.400 hierarchy (most commonly the PRMD, which is the only element that may be ommitted when conforming to recent versions of X.400). A domain may be associated with an omitted attribute in conjunction with several present ones. When performing the algorithmic insertion of components lower in the hierarchy, the omitted value shall be skipped. For example:

> If there is an MCGAM pair between domain HNE.EGM" and "O=HNE", "ADMD=ECQ", "C=TC", and omitted PRMD

then

> "ZI.HNE.EGM" is algorithmically mapped with "OU=I", "O=HNE", "ADMD=ECQ", "C=TC"

Attributes may have null values, and this is treated separately from omitted attributes (while it is not ideal to make this distinction, it is useful in practice).

4.2.1. Directory and Nameserver Mappings

When a set of MCGAMs are supported by X.500 or DNS, there is the possibility that results will be indeterminate due to timeout. Lookup shall be repeated until a value is determined, in order to maintain consistent gateway operation.

Where the mapping relates to an envelope address, the gateway shall non-deliver messages according to the associated MTA's normal timeout policy. Where the mapping relates to addresses in the message header, there shall be a timeout in the range of 1-4 hours or shorter if this is required to maintain quality of service constraints. If a mapping cannot be done in this time, address encapsulation shall be used.

4.3. EBNF.822-address <-> MTS.ORAddress

This section defines the basic address mapping.

4.3.1. X.400 encoded in RFC 822

This section defines how X.400 addresses are represented in RFC 822 addresses.

The std-or-address syntax is used to encode OR Address information in the 822.local-part of EBNF.822-address. Where there is an applicable equivalence mapping, further OR Address information is associated with the 822.domain component. This cannot be used in the general case, due to character set problems, and to the variants of X.400 OR Addresses which use different attribute types. The only way to encode the full PrintableString character set in a domain is by use of the 822.domain-ref syntax (i.e. 822.atom). This is likely to cause problems on many systems. The effective character set of domains is in practice reduced from the RFC 822 set, by restrictions imposed by domain conventions and policy [10], and by the EBNF definition in SMTP.

A generic 822.address consists of a 822.local-part and a sequence of 822.domains (e.g., <@domain1,@domain2:user@domain3>). All except the 822.domain associated with the 822.local-part (domain3 in this case) are considered to specify routing within the RFC 822 world, and will not be interpreted by the gateway (although they may have identified the gateway from within the RFC 822 world).

The 822.domain associated with the 822.local-part identifies the gateway from within the RFC 822 world. This final 822.domain may be used to determine some number of OR Address attributes, where this does not conflict with the first role. RFC 822 routing to gateways will usually be set up to facilitate the 822.domain being used for both purposes.

In the case that there is no applicable equivalence mapping, all of the X.400 address is encoded in the 822.local-part and the 822.domain identifies the gateway to which the message is being sent. This technique may be used by the RFC 822 user for any X.400 address where the equivalence mapping is not known.

In the case that there is an applicable MCGAM, the maximum number of attributes are encoded in the 822.domain. The remaining attributes are encoded on the LHS, using the EBNF.std-or-address syntax. For example:

```
/I=J/S=Linnimouth/GQ=5/@Marketing.Widget.COM
```

encodes the MTS.ORAddress consisting of:

```
MTS.CountryName                       = "TC"
MTS.AdministrationDomainName          = "BTT"
MTS.OrganizationName                  = "Widget"
MTS.OrganizationalUnitNames.value     = "Marketing"
MTS.PersonalName.surname              = "Linnimouth"
```

```
MTS.PersonalName.initials             = "J"
MTS.PersonalName.generation-qualifier = "5"
```

on the basis of an MCGAM pair between:

```
Domain: Widget.COM
OR Address: O="Widget", ADMD="BTT", C="TC"
```

Given the OR address, the domain Widget.COM is determined from the equivalence mapping and the next component is determined algorithmically to give Marketing.Widget.COM. The remaining attributes are encoded on the LHS in 822.local-part.

There is a further mechanism to simplify the encoding of common cases, where the only attributes to be encoded on the LHS are (non-Teletex) Personal Name attributes which comply with the restrictions of 4.1.2. To achieve this, the 822.local-part shall be encoded as EBNF.encoded-pn. In the previous example, if the GenerationQualifier was not present in the OR Address, it would map with the RFC 822 address: J.Linnimouth@Marketing.Widget.COM.

From the standpoint of the RFC 822 Message Transfer System, the domain specification is used to route the message in the standard manner. The standard domain mechanisms are used to select appropriate gateways for the corresponding OR Address space. It is the responsibility of the management that defines the equivalence mapping to define routing in the manner which will enable the message to be delivered.

4.3.2. RFC 822 encoded in X.400

The previous section showed a mapping from X.400 to RFC 822. In the case where the mapping was symmetrical and based on the equivalence mapping, this has also shown how RFC 822 is encoded in the X.400. This equivalence cannot be used for all RFC 822 addresses.

The general case is mapped by use of domain defined attributes. A (Printable String) Domain defined type "RFC-822" is defined. The associated attribute value is an ASCII string encoded according to Section 3.3.3 of this specification. The interpretation of the ASCII string follows RFC 822, and RFC 1123 [10,16]. Domains shall always be fully qualified.

Other OR Address attributes will be used to identify a context in which the OR Address will be interpreted. This might be a Management Domain, or some part of a Management Domain which identifies a gateway MTA. For example:

```
C               = "GB"
ADMD            = "GOLD 400"
PRMD            = "UK.AC"
O               = "UCL"
OU              = "CS"
"RFC-822"      =  "Jimmy(a)WIDGET-LABS.CO.UK"
```

OR

```
C               = "TC"
ADMD            = "Wizz.mail"
PRMD            = "42"
"rfc-822"       = "postel(a)venera.isi.edu"
```

Note in each case the PrintableString encoding of "@" as "(a)". In the second example, the "RFC-822" domain defined attribute is interpreted everywhere within the (Private) Management Domain. In the first example, further attributes are needed within the Management Domain to identify a gateway. Thus, this scheme can be used with varying levels of Management Domain co-operation.

There is a limit of 128 characters in the length of value of a domain defined attribute, and an OR Address can have a maxmimum of four domain defined attributes. Where the printable string generated from the RFC 822 address exceeds 128 characters, additional domain defined attributes are used to enable up to 512 characters to be encoded. These attributes shall be filled completely before the next one is started. The (Printable String) DDA keywords are: RFC822C1; RFC822C2; RFC822C3. Longer addresses cannot be encoded.

MIXER defines a representation of RFC 822 addresses in printable string domain defined attributes. Teletex domain defined attributes with a key of RFC-822, RFC822C1; RFC822C2; RFC822C3 shall not be generated. This is for backwards compatibility reasons.

Reception of these attributes in the manner defined below is mandatory. This is to allow the possibility for future versions of MIXER to allow generation of teletex domain defined attributes. Where the values of all of these teletex domain defined attributes are printable string characters, they shall be interpreted in the same way as the printable string domain defined attributes. If this is not the case, the printable string encoding translation shall be omitted. If both teletex and printable string attributes are present, this is valid if and only if they represent exactly the same RFC 822 address.

4.3.3. Component Ordering

In most cases, ordering of OR Address components is not significant for the mappings specified. However, Organizational Units (printable string and teletex forms) and Domain Defined Attributes are specified as SEQUENCE in MTS.ORAddress, and so their order may be significant. This specification needs to take account of this:

1. To allow consistent mapping into the domain hierarchy

2. To ensure preservation of order over multiple mappings.

There are three places where an order is specified:

1. The text encoding (std-or-address) of MTS.ORAddress as used in the local-part of an RFC 822 address. An order is needed for those components which may have multiple values (Organizational Unit, and Domain Defined Attributes). When generating an 822.std-or-address, components of a given type shall be in hierarchical order with the most significant component on the RHS (right hand side or domain part). If there is an Organization Attribute, it shall be to the right of any Organizational Unit attributes. These requirements are for the following reasons:

- Alignment to the hierarchy of other components in RFC 822 addresses (thus, Organizational Units will appear in the same order, whether encoded on the RHS or LHS).

- Backwards compatibility with RFC 987/1026.

- To ensure that gateways generate consistent addresses. This is both to help end users, and to generate identical message ids.

Further, it is recommended that all other attributes are generated according to this ordering, so that all attributes so encoded follow a consistent hierarchy. When generating 822.msg-id, this order shall be followed.

2. For the Organizational Units (OU) in MTS.ORAddress, the first OU in the SEQUENCE is the most significant, as specified in X.400.

3. For the Domain Defined Attributes in MTS.ORAddress, the First Domain Defined Attribute in the SEQUENCE is the most significant.

Note that although this ordering is mandatory for this mapping, MIXER does not give additional implications on the ordering significance within X.400.

4.3.4. RFC 822 -> X.400 Basic Address Mapping

There are two basic cases:

1. X.400 addresses encoded in RFC 822. This will also include RFC 822 addresses which are given reversible encodings.

2. "Genuine" RFC 822 addresses.

The mapping shall proceed as follows, by first assuming case 1).

STAGE I.

1. If the 822-address is not of the form:

local-part "@" domain

take the domain which will be routed on and apply step 2 of stage 1 to derive (a possibly null) set of attributes. Then go to stage II.

The gateway may reduce a source route address to this form by removal of all but the last domain. In terms of the design intentions of RFC 822, this would be an incorrect action. (Note that an address of the form local%part@domain is not a source route). However, in most cases, it will provide a better service

to the end user, and is in line with the Internet Host Requirements. This is a reflection on the common inappropriate use of source routing in RFC 822 based systems, despite the discussion in the Host Requirements [10]. Either approach, or the intermediate approach of stripping only domain references which reference the local gateway are conformant to this specification.

2. If the 822.local-part uses the 822.quoted-string encoding, remove this quoting. If the resulting unquoted 822.local-part has leading space, trailing space, or two adjacent spaces go to stage II.

3. If the unquoted 822.local-part contains any characters not in PrintableString, "{", "}", "*", and "$", go to stage II.

4. Parse the (unquoted) 822.local-part according to the EBNF EBNF.std-or-address-input. Checking of upper bounds shall not be done at this point. If this parse fails, parse the local-part according to the EBNF EBNF.encoded-pn. If this parse fails, go to stage II. The result is a set of type/value pairs.

5. Associate the EBNF.attribute-value syntax (determined from the identified type) with each value, and check that it conforms. If not, go to stage II.

6. If the set of attributes forms a valid X.400 address, according to X.402, then go to step 9. All forms of X.400 address are allowed at this stage. Steps 7-8 default attributes for certain types of OR Address.

7. If the set of attributes cannot form a mnemonic form of X.400 address after addition of attributes which may be derived from the EBNF.domain (C, ADMD, PRMD, O, OU), go to stage II.

8. Attempt to parse EBNF.domain as:

   *(domain-syntax ".") known-domain

   Where EBNF.known-domain is the longest possible match in the set of MCGAMs being used by the gateway (described in Section 4.2). EBNF.domain-syntax is the restricted domain syntax defined in Section 4.2, to which all of the domain components shall conform for the parse to be successful. If this fails, go to stage II.

For each component, systematically allocate the attribute implied by each EBNF.domain-syntax component in the order: C, ADMD, PRMD, O, OU. Note that if the MCGAM used identifies an "omitted attribute", then this attribute shall be omitted in the systematic allocation. If this new component exceed an upper bound (ADMD: 16; PRMD: 16; O: 64; OU: 32) or it would lead to more than four OUs, then go to stage II with the attributes derived.

The attributes derived in this step (referred to as RHS attributes) are merged with the ones derived from the LHS (step 6). In some cases, not all of the RHF attributes are used. LHS attributes are all used. C will not be in the LHS attributes. If ADMD is in the LHS attributes, only C is taken from the RHS attributes. If PRMD is in the LHS attributes, C and ADMD are taken from the RHS attributes. If O is on the LHS, C, ADMD and PRMD (if present) are taken from the RHS attributes. In other cases all RHS attributes are taken.

9. Ensure that the set of attributes conforms both to the MTS.ORAddress specification and to the restrictions on this set given in X.400, and that no upper bounds are exceeded for any attribute. If not go to stage II.

10. Build the OR Address from this information.

STAGE II.

This will only be reached if the RFC 822 EBNF.822-address is not a valid X.400 encoding. This implies that the address refers to a recipient on an RFC 822 system or that the encoding of the address is invalid. Such addresses shall be encoded in an X.400 OR Address using a domain defined attribute.

1. Convert the EBNF.822-address to PrintableString, as specified in Chapter 3.

2. Generate the "RFC-822" domain defined attribute from this string.

3. Build the rest of the OR Address in the manner described below.

It is not always possible to encode the domain defined attribute due to length restrictions. If the limit is exceeded by a mapping at the MTS level, then the gateway shall reject the message in question. If this occurs at the IPMS level, then the action will depend on the policy being taken for IPMS encoding, which is discussed in Section 5.1.3.

Use Stage I, step 8, to generate a set of attributes to build the remainder of the address. The administrative equivalence of the mappings will ensure correct routing through X.400 to a gateway back to RFC 822.

If Stage I, step 8 does not generate a set of attributes or the address generated is unroutable, the remained of the OR address is generated as follows. The remainder of the OR address effectively identifies a source route to a gateway from the X.400 side. There are three cases, which are handled differently:

SMTP Return Address
: This shall be set up so that errors are returned through the same gateway. Therefore, the OR Address of the local gateway shall be used.

IPMS Addresses
: These are optimised for replying. In general, the message may end up anywhere within the X.400 world, and so this optimisation identifies a gateway appropriate for the RFC 822 address being converted. The 822.domain to which the address would be routed is used to select an appropriate gateway.

  In this case, it may be useful to use a non-local gateway, which will optimise the reply address. This information may be looked up in gateway tables in a manner equivalent to the MCGAM lookup. Because of the similarity of lookup, the three MCGAM lookup mechanisms (table, X.500, DNS) are also available to look up this information. This information is local, and a gateway may insert any appropriate (gateway) OR Address. The longest possible match on the 822.domain defines which gateway to use. This mechanism is used for any part of the X.400 namespace for which it is desirable to identify a preferred X.400 gateway in order to optimise routing.

  If no mapping is found for the 822.domain, a default value (typically that of the local gateway) is used. It is never appropriate to ignore the locally used MCGAMs.

SMTP Recipient
As the RFC 822 and X.400 worlds are in principle fully connected, there is no technical reason for this situation to occur. In practice, this is not the case. In some cases, routing may be configured to use X.400 to connect an RFC 822 island to the Internet. The information that this part of the domain space is to be routed by X.400 rather than remaining within the RFC 822 world shall be configured privately into the gateway in question. X.400 routing shall not make use of the presence of the RFC-822 DDA to perform X.400 routing. The OR address shall then be generated in the same manner as for an IPMS address, using the locally available MCGAMs. It is to support this case that the definition of the global domain to gateway mapping is important, as the use of this mapping will lead to a remote X.400 address, which can be routed by X.400 routing procedures. The information in this mapping shall not be used as a basis for deciding to convert a message from RFC 822 to X.400.

Three examples are given, neither of which has applicable MCGAMs.

Example 1: (Address not in "localpart" "@" "domainpart")

@relay.co.uk:userb@host2

maps to

c=gb; a= ; p=uk.ac; o=mr; dd.rfc-822=(a)relay.co.uk:userb(a)host2;

Example 2: (Address with non printablestring characters)

Tom_Harris@cs.widget.com

maps to

c=us; a=MCI; P=relay; dd.rfc-822=Tom(u)Harris(a)cs.widget.com;

Example 3: (Address with an entry for alter.net into the OR Address of Preferred Gateway table, pointing to c=gb; A=BTglobal; P=relay)

postmaster@UK.alter.net

maps to

c=gb; a=BTglobal; P=relay; dd.rfc-822=postmaster(a)UK.alter.net;

4.3.4.1. Heuristic for mapping RFC 822 to X.400

The following heuristic, which relates to ordering of address components, may be used when mapping from RFC 822 to X.400. The ordering of attributes may be inverted or mixed, and so the following heuristics may be applied:

> If there is an Organization attribute to the left of any Org Unit attribute, assume that the hierarchy is inverted. This is to facilitate the situation where a user has input the attributes in reverse hierarchical order. To do this the gateway shall first map according to the order defined in 4.3.3. If this mapping generates an address which X.400 address verification shows to be invalid, this heuristic may be applied as an alternative to immediate rejection of the address.

4.3.5. X.400 -> RFC 822 Basic Address Mapping

There are two basic cases:

1. RFC 822 addresses encoded in X.400.

2. "Genuine" X.400 addresses. This may include symmetrically encoded RFC 822 addresses.

When an MTS Recipient OR Address is interpreted, gatewaying will be selected if there is a single "RFC-822" domain defined attribute present. In this case, use mapping A and in other cases, use mapping B.

RFC 1327 specified that this shall only be done when the gateway identfied is local or otherwise known, and identified the approach specified here as a pragmatic option. Experience has shown that this is effective in practice, despite theoretical problems.

If a gateway wishes to make a mapping in a manner similar to RFC 1327, but does not wish for this global interpretation (e.g., to support an RFC 822 local system, which does not use global addressing), then it may choose a private domain defined attribute, different to "RFC-822". An RFC 1327 gateway might be configurable to operate in this manner.

Mapping A

1. Map the domain defined attribute value to ASCII, as defined in Chapter 3, and drop all other attributes.

Mapping B

This is used for X.400 addresses which do not use the explicit RFC 822 encoding.

1. For all string encoded attributes, remove any leading or trailing spaces, and replace adjacent spaces with a single space.

   The only attribute which is permitted to have zero length is the ADMD. This shall be mapped onto a single space.

   These transformations are for lookup only. If an EBNF.std-or-address mapping is used as in 4), then the original values shall be used.

2. The numeric country codes may be mapped to the two letter values (as defined in ISO 3166). Global mappings are usually only defined in terms of the ISO 3166 codes.

3. Noting the hierarchy specified in 4.3.1 and including omitted attributes, determine the maximum set of attributes which have an associated domain specification in the local set of MCGAMs. If no match is found, allocate the domain as described below, and go to step 5. The default domain to be used is the specification of the local gateway. A gateway may use other domains according to private mapping tables or heuristics. For example, it may choose a domain which it knows to provide a free gateway service to the mapped address.

   In cases where the address refers to an X.400 UA, it is important that the generated domain will correctly route to a gateway. In general, this is achieved by carefully co-ordinating RFC 822 routing with the definition of the MCGAMs, as there is no easy way for the gateway to make this check. One rule that shall be used is that domains with only one component will not route to a gateway. If the generated domain does not route correctly, the address is treated as if no match is found.

   The gateway may also make use of a mapping equivalent to the MCGAM mapping to determine the domain to use. This mapping is done from the OR Address hierarchy. This is not a global mapping, but is a routing style mapping from the OR Address space, to enable a best choice domain to be inserted. This mapping is supported by the three MCGAM lookup mechanisms.

4. The mapping identified in 3) gives a domain, and an OR address prefix. Follow the hierarchy: C, ADMD, PRMD, O, OU. For each successive component below the OR address prefix, which conforms to the syntax EBNF.domain-syntax (as defined in 4.3.1), allocate the next subdomain. At least one attribute of the X.400 address shall not be mapped onto subdomain, as 822.local-part cannot be null. If there are omitted attributes in the OR address prefix, these will have correctly and uniquely mapped to a domain component. Where there is an attribute omitted below the prefix, all attributes remaining in the OR address shall be encoded on the LHS. This is to ensure a reversible mapping. For example, if there is an address /S=XX/O=YY/ADMD=A/C=NN/ and a mapping for /ADMD=A/C=NN/ is used, then /S=XX/O=YY/ is encoded on the LHS.

5. If the address contains any attribute not used in mnemonic form, then all of the attributes in the address shall be encoded on the LHS in EBNF.std-or-address syntax, as described below.

   For addresses of mnemonic form, if the remaining components are personal-name components, conforming to the restrictions of 4.2.1, then EBNF.encoded-pn is derived to form 822.local-part. In other cases the remaining components are simply encoded as 822.local-part using the EBNF.std-or-address syntax. If necessary, the 822.quoted-string encoding is used. The following are examples of legal quoting: "a b".c@x; "a b.c"@x. Either form may be generated. Generation of the latter style is strongly recommended.

Four examples are given.

Example 1: (Address with missing X.400 elements and no specific mapping rule for "o=sales; a=Master400; C=it", where a mapping exists for a=master400; C=it;)

S=Support; O=sales; A=Master400; C=it;

maps to

/S=Support/o=sales/@Master400.it

Example 2: (Address with illegal characters in RFC822 generated domain if default hierarchical translation (specific mapping rule is existing for c=fr; a=atlas; p=autoroutes) is used)

S=renseignements; O=Region Parisienne; P=autoroutes; A=atlas; C=fr;

maps to

"/S=renseignements/o=Region Parisienne/"@autoroutes.fr

Example 3: (Address containing elements not mappable into RFC822 local part)

S=Rossi; DD.cap=20100; DD.ph1=Via Larga 11; DDA.city=Milano; A=PtPostel; C=it;

maps to

"/DD.cap=20100/DD.ph1=Via Larga 11/DD.city=Milano/S=Rossi/"@ptpostel.it

Example 4: (Address with an entry for A=ATT; C=us; into the domain of Preferred Gateway table, pointing to attmail.com)

G=Andy; S=Wharol; O=MMNY; A=ATT; C=us;

maps to

/G=Andy/S=Wharol/O=MMNY@attmail.com

4.4. Repeated Mappings

There are two types of repeated mapping:

1. A recursive mapping, where the repeat is within one gateway

2 A source route, where the repetition occurs across multiple gateways

4.4.1. Recursive Mappings

It is possible to supply an address which is recursive at a single gateway. For example:

```
C          = "XX"
ADMD       = "YY"
O          = "ZZ"
"RFC-822"  = "Smith(a)ZZ.YY.XX"
```

This is mapped first to an RFC 822 address, and then back to the X.400 address:

```
C          = "XX"
ADMD       = "YY"
O          = "ZZ"
Surname    = "Smith"
```

In some situations this type of recursion may be frequent. It is important where this occurs, that no unnecessary protocol conversion occurs. This will minimise loss of service.

4.4.2. Source Routes

The mappings defined are symmetrical and reversible across a single gateway. The symmetry is particularly useful in cases of (mail exploder type) distribution list expansion. For example, an X.400 user sends to a list on an RFC 822 system which he belongs to. The received message will have the originator and any 3rd party X.400 OR Addresses in correct format (rather than doubly encoded). In cases (X.400 or RFC 822) where there is common agreement on gateway identification, then this will apply to multiple gateways.

When a message traverses multiple gateways, the mapping will always be reversible, in that a reply can be generated which will correctly reverse the path. In many cases, the mapping will also be symmetrical, which will appear clean to the end user. For example, if countries "AB" and "XY" have RFC 822 networks, but are interconnected by X.400, the following may happen: The originator specifies:

```
Joe.Soap@Widget.PTT.XY
```

This is routed to a gateway, which generates:

```
C               = "XY"
ADMD            = "PTT"
PRMD            = "Griddle MHS Providers"
Organization    = "Widget Corporation"
Surname         = "Soap"
Given Name      = "Joe"
```

This is then routed to another gateway where the mapping is reversed to give:

```
Joe.Soap@Widget.PTT.XY
```

Here, use of the gateway is transparent.

Mappings will only be symmetrical where mapping equivalences are defined. In other cases, the reversibility is more important, due to the (far too frequent) cases where RFC 822 and X.400 services are partitioned.

The syntax may be used to source route. THIS IS STRONGLY DISCOURAGED. For example:

```
X.400 -> RFC 822  -> X.400

C             = "UK"
ADMD          = "Gold 400"
PRMD          = "UK.AC"
"RFC-822"     = "/PN=Duval/DD.Title=Manager/(a)Inria.ATLAS.FR"
```

This will be sent to an arbitrary UK Academic Community gateway by X.400. Then it will be sent by JNT Mail to another gateway determined by the domain Inria.ATLAS.FR (FR.ATLAS.Inria). This will then derive the X.400 OR Address:

```
C             = "FR"
ADMD          = "ATLAS"
PRMD          = "Inria"
PN.S          = "Duval"
"Title"       = "Manager"
```

Similarly:

RFC 822 -> X.400 -> RFC 822

"/RFC-822=jj(a)seismo.css.gov/PRMD=AC/ADMD=BT/C=GB/"@monet.berkeley.edu

This will be sent to monet.berkeley.edu by RFC 822, then to the AC PRMD by X.400, and then to jj@seismo.css.gov by RFC 822.

4.5. Directory Names

Directory Names are an optional part of OR Name, along with OR Address. The RFC 822 addresses are mapped onto the OR Address component. As there is no functional mapping for the Directory Name on the RFC 822 side, a textual mapping is used. There is no requirement for reversibility in terms of the goals of this specification. There may be some loss of functionality in terms of third party recipients where only a directory name is given, but this seems preferable to the significant extra complexity of adding a full mapping for Directory Names.

The Directory Name shall be represented within an RFC 822 comment using the comaptible formats of RFC 1484 or RFC 1485. It is recommended that the directory string format of RFC 1485 is used [24]. The User Friendly Name form of RFC 1484 may be used [25].

4.6. MTS Mappings

The basic mappings at the MTS level are:

1) SMTP originator ->
   MTS.PerMessageSubmissionFields.originator-name
   MTS.OtherMessageDeliveryFields.originator-name ->
   SMTP originator

2) SMTP recipient ->
   MTS.PerRecipientMessageSubmissionFields
   MTS.OtherMessageDeliveryFields.this-recipient-name ->
   SMTP recipient

SMTP recipients and return addresses are encoded as EBNF.822-address.

The MTS Originator is always encoded as MTS.OriginatorName, which maps onto MTS.ORAddressAndOptionalDirectoryName, which in turn maps onto MTS.ORName.

4.6.1. RFC 822 -> X.400 MTS Mappings

From the SMTP Originator, use the basic ORAddress mapping, to generate MTS.PerMessageSubmissionFields.originator-name (MTS.ORName), without a DirectoryName.

For recipients, the following settings are made for each component of MTS.PerRecipientMessageSubmissionFields.

recipient-name
: This is derived from the SMTP recipient by the basic ORAddress mapping.

originator-report-request
: This may either be set to "delivery-report", or set according to SMTP extensions as set out in Appendix A.

explicit-conversion
: This optional component is omitted, as this service is not needed

extensions
: The default value (no extensions) is used

4.6.2. X.400 -> RFC 822 MTS Mappings

The basic functionality is to generate the SMTP originator and recipients. There is information present on the X.400 side, which cannot be mapped into analogous SMTP services. For this reason, new RFC 822 fields are added for the MTS Originator and Recipients. The information discarded at the SMTP level will be present in these fields. In some cases a (positive) delivery report will be generated.

4.6.2.1. SMTP Mappings

Use the basic ORAddress mapping, to generate the SMTP originator (return address) from MTS.OtherMessageDeliveryFields.originator-name (MTS.ORName). If MTS.ORName.directory-name is present, it is discarded. (Note that it will be presented to the user, as described in 4.6.2.2).

The mapping uses the MTA level information, and maps each value of MTA.PerRecipientMessageTransferFields.recipient-name, where the responsibility bit is set, onto an SMTP recipient.

> Note:The SMTP recipient is conceptually generated from MTS.OtherMessageDeliveryFields.this-recipient-name. This is done by taking MTS.OtherMessageDeliveryFields.this-recipient-name, and generating an SMTP recipient according to the basic ORAddress

mapping, discarding MTS.ORName.directory-name if present. However, if this model was followed exactly, there would be no possibility to have multiple SMTP recipients on a single message. This is unacceptable, and so layering is violated.

4.6.2.2. Generation of RFC 822 Headers

Not all per-recipient information can be passed at the SMTP level. For this reason, two new RFC 822 headers are created, in order to carry this information to the RFC 822 recipient. These fields are "X400-Originator:" and "X400-Recipients:".

The "X400-Originator:" field is set to the same value as the SMTP originator. In addition, if MTS.OtherMessageDeliveryFields.originator-name (MTS.ORName) contains MTS.ORName.directory-name then this Directory Name shall be represented in an 822.comment.

Recipient names, taken from each value of MTS.OtherMessageDeliveryFields.this-recipient-name and MTS.OtherMessageDeliveryFields.other-recipient-names are made available to the RFC 822 user by use of the "X400-Recipients:" field. By taking the recipients at the MTS level, disclosure of recipients will be dealt with correctly. However, this conflicts with a desire to optimise mail transfer. There is no problem when disclosure of recipients is allowed. Similarly, there is no problem if there is only one RFC 822 recipient, as the "X400-Recipients" field is only given one address.

There is a problem if there are multiple RFC 822 recipients, and disclosure of recipients is prohibited. In this case, discard the per-recipient information.

If any MTS.ORName.directory-name is present, it shall be represented in an 822.comment.

If MTS.OtherMessageDeliveryFields.orignally-intended-recipient-name is present, then there has been redirection, or there has been distribution list expansion. Distribution list expansion is a per-message option, and the information associated with this is represented by the "DL-Expansion-History:" field described in Section 5.3.6. Other information is represented in an 822.comment associated with MTS.OtherMessageDeliveryFields.this-recipient-name, The message may be delivered to different RFC 822 recipients, and so several addresses in the "X400-Recipients:" field may have such comments. The non-commented recipient is the RFC 822 recipient. The EBNF of the comment is defined by redirect-comment.

```
redirect-comment  = redirect-first *( redirect-subsequent )

redirect-first = "Originally To:"  mailbox  "Redirected on"
   date-time "To:"  redirection-reason

redirect-subsequent = mailbox  "Redirected Again on"
   date-time "To:"  redirection-reason

redirection-history-item = "intended recipient" mailbox
   "redirected to"  redirection-reason
   "on" date-time

redirection-reason =
   "Recipient Assigned Alternate Recipient"
   / "Originator Requested Alternate Recipient"
   / "Recipient MD Assigned Alternate Recipient"
   / "Directory Look Up"
   / "Alias"
```

It is derived from MTA.PerRecipientMessageTransferFields.extension.redirection-history. The values are taken from the X.400(92) Implementor's guide (Version 13, July 1995). The first three values are in X.400(88). The fourth value is in X.400(92), but has the name "recipient-directory-substitution-alternate-recipient". An example of this with two redirects is:

```
X400-Recipients: postmaster@widget.com (Originally To:
          sales-manager@sales.widget.com
      Redirected on Thu, 30 May 91 14:39:40 +0100
          To: Originator Requested Alternate Recipient
          postmaster@sales.widget.com
      Redirected Again on Thu, 30 May 91 14:41:20 +0100
          To: Recipient MD Assigned Alternate Recipient)
```

In addition the following per-recipient services from MTS.OtherMessageDeliveryFields.extensions are represented in comments if they are used. None of these services can be provided on RFC 822 networks, and so in general these will be informative strings associated with other MTS recipients. In some cases, string values are defined. For the remainder, the string value shall be chosen by the implementor. If the parameter has a default value, then no comment shall be inserted when the parameter has that default value.

requested-delivery-method

physical-forwarding-prohibited
"(Physical Forwarding Prohibited)".

physical-forwarding-address-request
"(Physical Forwarding Address Requested)".

physical-delivery-modes

registered-mail-type

recipient-number-for-advice

physical-rendition-attributes

physical-delivery-report-request
"(Physical Delivery Report Requested)".

proof-of-delivery-request
"(Proof of Delivery Requested)".

4.6.2.3. Delivery Report Generation

If SMTP is used, the behaviour is specified in Appendix A. In other cases, if MTA.PerRecipientMessageTransferFields.per-recipient-indicators requires a positive delivery notification, this shall be generated by the gateway. Supplementary Information shall be set to indicate that the report is gateway generated. This information shall include the name of the gateway generating the report.

4.6.3. Message IDs (MTS)

A mapping from 822.msg-id to MTS.MTSIdentifier is defined. The reverse mapping is not needed, as MTS.MTSIdentifier is always mapped onto new RFC 822 fields. The value of MTS.MTSIdentifier.local-part will facilitate correlation of gateway errors.

To map from 822.msg-id, apply the standard mapping to 822.msg-id, in order to generate an MTS.ORAddress. The Country, ADMD, and PRMD components of this are used to generate MTS.MTSIdentifier.global-domain-identifier. MTS.MTSIdentifier.local-identifier is set to the 822.msg-id, including the braces "<" and ">". If this string is longer than MTS.ub-local-id-length (32), then it is truncated to this length.

The reverse mapping is not used in this specification. It would be applicable where MTS.MTSIdentifier.local-identifier is of syntax 822.msg-id, and it algorithmically identifies MTS.MTSIdentifier.

4.7. IPMS Mappings

All RFC 822 addresses are assumed to use the 822.mailbox syntax. This includes all 822.comments associated with the lexical tokens of the 822.mailbox. In the IPMS OR Names are encoded as MTS.ORName. This is used within the IPMS.ORDescriptor, IPMS.RecipientSpecifier, and IPMS.IPMIdentifier. An asymmetrical mapping is defined between these components.

4.7.1. RFC 822 -> X.400

To derive IPMS.ORDescriptor from an RFC 822 address.

1. Take the address, and extract an EBNF.822-address. Any source routing shall be removed. This can be derived trivially from either the 822.addr-spec or 822.route-addr syntax. This is mapped to MTS.ORName as described above, and used as IMPS.ORDescriptor.formal-name.

2. A string shall be built consisting of (if present):

- The 822.phrase component if the 822.address is an 822.phrase 822.route-addr construct.

- Any 822.comments, in order, retaining the parentheses.

This string is then encoded into T.61 using a human oriented mapping (as described in Section 3.5). If the string is not null, it is assigned to IPMS.ORDescriptor.free-form-name.

3. IPMS.ORDescriptor.telephone-number is omitted.

If IPMS.ORDescriptor is being used in IPMS.RecipientSpecifier, IPMS.RecipientSpecifier.reply-request and IPMS.RecipientSpecifier.notification-requests are set to default values (false and none).

If the 822.group construct is present, any included 822.mailbox is encoded as above to generate a separate IPMS.ORDescriptor. The 822.group is mapped to T.61 (as described in Section 3.5), and a IPMS.ORDescriptor with only an free-form-name component built from it.

4.7.2. X.400 -> RFC 822

Mapping from IPMS.ORDescriptor to RFC 822 address. In the basic case, where IPMS.ORDescriptor.formal-name is present, proceed as follows.

1. Encode IPMS.ORDescriptor.formal-name (MTS.ORName) as EBNF.822-address.

2a. If IPMS.ORDescriptor.free-form-name is present, convert it to ASCII or T.61 (Section 3.5), and use this as the 822.phrase component of 822.mailbox using the 822.phrase 822.route-addr construct.

2b. If IPMS.ORDescriptor.free-form-name is absent. If EBNF.822-address is parsed as 822.addr-spec use this as the encoding of 822.mailbox. If EBNF.822-address is parsed as 822.route 822.addr-spec, then an 822.phrase taken from 822.local-part is added.

3 If IPMS.ORDescriptor.telephone-number is present, this is placed in an 822.comment, with the string "Tel ". The normal international form of number is used. For example:

(Tel +44-181-333-7777)

4. If IPMS.ORDescriptor.formal-name.directory-name is present, then a text representation is placed in a trailing 822.comment.

5. If IPMS.RecipientSpecifier.report-request has any non-default values, then an 822.comment "(Receipt Notification Requested)", and/or "(Non Receipt Notification Requested)", and/or "(IPM Return Requested)" may be appended to the address. "(Receipt Notification Requested)" may be used to infer "(Non Receipt Notification Requested)". The effort of correlating P1 and P2 information is too great to justify the gateway sending Receipt Notifications.

In RFC 1327, inclusion of these comments was mandatory. Experience has shown that the clutter and confusion caused to RFC 822 users does not justify the information conveyed. Implementors are recommended to not include these comments. Unless an application is found where retention of these comments is desirable, they will be dropped from the next version.

6. If IPMS.RecipientSpecifier.reply-request is True, an 822.comment "(Reply requested)" is appended to the address.

If IPMS.ORDescriptor.formal-name is absent, IPMS.ORDescriptor.free-form-name is converted to ASCII (see section 3.5), and used as 822.phrase within the RFC 822 822.group syntax. For example:

Free Form Name ":" ";"

Steps 3-6 are then followed.

4.7.3. IP Message IDs

There is a need to map both ways between 822.msg-id and IPMS.IPMIdentifier. This allows for X.400 Receipt Notifications, Replies, and Cross References to reference an RFC 822 Message ID, which is preferable to a gateway generated ID. A reversible and symmetrical mapping is defined. This provides fully reversible mappings when messages pass multiple times across the X.400/RFC 822 boundary.

An important issue with messages identifiers is mapping to the exact form, as many systems use these ids as uninterpreted keys. The use of table driven mappings is not always symmetrical, particularly in the light of alternative domain names, and alternative management domains. For this reason, a purely algorithmic mapping is used. A mapping which is simpler than that for addresses can be used for two reasons:

- There is no major requirement to make message IDs "natural"

- There is no issue about being able to reply to message IDs. (For addresses, creating a return path which works is more important than being symmetrical).

The mapping works by defining a way in which message IDs generated on one side of the gateway can be represented on the other side in a systematic manner. The mapping is defined so that the possibility of clashes is low enough to be treated as impossible.

4.7.3.1. 822.msg-id represented in X.400

IPMS.IPMIdentifier.user is omitted. The IPMS.IPMIdentifier.user-relative-identifier is set to a printable string encoding of the 822.msg-id with the angle braces ("<" and ">") removed. The upper bound on this component is 64. The options for handling this are discussed in Section 5.1.3.

4.7.3.2. IPMS.IPMIdentifier represented in RFC 822

The 822.domain of 822.msg-id is set to the value "MHS". The 822.local-part of 822.msg-id is constructed by building a string of syntax EBNF.id-loc from IPMS.IPMIdentifier.

```
id-loc ::= [ printablestring ] "*"  [ std-or-address ]
```

EBNF.printablestring is the IPMS.IPMIdentifier.user-relative-identifier, and EBNF.std-or-address being an encoding of the IPMS.IPMIdentifier.user derived according to this specification. 822.local-part is derived from EBNF.id-loc, if necessary using the 822.quoted-string encoding. For example:

<"147*/S=Dietrich/O=Siemens/ADMD=DBP/C=DE/"@MHS>

4.7.3.3. 822.msg-id -> IPMS.IPMIdentifier

If the 822.local-part can be parsed as:

[printablestring] "*" [std-or-address]

and the 822.domain is "MHS", then this ID was X.400 generated. If EBNF.printablestring is present, the value is assigned to IPMS.IPMIdentifier.user-relative-identifier. If EBNF.std-or-address is present, the OR Address components derived from it are used to set IPMS.IPMIdentifier.user.

Otherwise, this is an RFC 822 generated ID. In this case, set IPMS.IPMIdentifier.user-relative-identifier to a printable string encoding of the 822.msg-id without the angle braces and omit IPMS.IPMID.user.

4.7.3.4. IPMS.IPMIdentifier -> 822.msg-id

If IPMS.IPMIdentifier.user is absent, and IPMS.IPMIdentifier.user-relative-identifier mapped to ASCII and angle braces added parses as 822.msg-id, then this is an RFC 822 generated ID.

Otherwise, the ID is X.400 generated. Use the IPMS.IPMIdentifier.user to generate an EBNF.std-or-address form string. Build the 822.local-part of the 822.msg-id with the syntax:

[printablestring] "*" [std-or-address]

The printablestring is taken from IPMS.IPMIdentifier.user-relative-identifier. Use 822.quoted-string if necessary. The 822.msg-id is generated with this 822.local-part, and "MHS" as the 822.domain.

4.7.3.5. Phrase form

In "In-Reply-To:" and "References:", the encoding 822.phrase may be used as an alternative to 822.msg-id. To map from 822.phrase to IPMS.IPMIdentifier, assign IPMS.IPMIdentifier.user-relative-identifier to the phrase. When mapping from IPMS.IPMIdentifier for "In-Reply-To:" and "References:", if IPMS.IPMIdentifier.user is

absent and IPMS.IPMIdentifier.user-relative-identifier does not parse as 822.msg-id, generate an 822.phrase rather than adding the domain MHS.

4.7.3.6. RFC 987 backwards compatibility

The mapping defined here is different to that used in RFC 987, as the RFC 987 mapping lead to changed message IDs in many cases. Fixing the problems is preferable to retaining backwards compatibility. An implementation of this standard may recognise message IDs generated by RFC 987. This is not recommended.

RFC 987 generated encodings may be recognised as follows. When mapping from X.400 to RFC 822, if the IPMS.IPMIdentifier.user-relative-identifier is "RFC-822" the id is RFC 987 generated. When mapping from RFC 822 to X.400, if the 822.domain is not "MHS", and the 822.local-part can be parsed as

[printablestring] "*" [std-or-address]

then it is RFC 987 generated. In each of these cases, it is recommended to follow the RFC 987 rules.

Chapter 5 - Detailed Mappings

This chapter specifies detailed mappings for the functions outlined in Chapters 1 and 2. It makes extensive use of the notations and mappings defined in Chapters 3 and 4.

5.1. RFC 822 -> X.400: Detailed Mappings

The mapping of RFC 822/MIME messages to X.400 InterPersonal Messages is described in Sections 5.1.1 to 5.1.7. Mapping of NOTARY format delivery status notifications, which are all messages of type multipart/report and subtype delivery-status-notifications to X.400 delivery reports is covered in Section 5.1.8.

5.1.1. Basic Approach

A single IP Message is generated from an RFC 822 message. The RFC 822 headers are used to generate the IPMS.Heading.

Some RFC 822 fields cannot be mapped onto a standard IPM Heading field, and so an extended field is defined in Section 5.1.2. This is then used for fields which cannot be mapped onto existing services.

The message is submitted to the MTS, and the services required can be defined by specifying MTS.MessageSubmissionEnvelope. A few parameters of the MTA Abstract service are also specified, which are not in principle available to the MTS User. Use of these services allows RFC 822 MTA level parameters to be carried in the analogous X.400 service elements. The advantages of this mapping far outweigh the layering violation.

5.1.2. X.400 Extension Field

An IPMS Extension is defined:

```
rfc-822-field HEADING-EXTENSION
           VALUE RFC822FieldList
           ::= id-rfc-822-field-list


RFC822FieldList ::= SEQUENCE OF RFC822Field

RFC822Field ::= IA5String
```

The Object Identifier id-rfc-822-field-list is defined in Appendix D.

To encode any RFC 822 Header using this extension, an RFC822Field element is built using the 822.field omitting the trailing CRLF (e.g., "Fruit-Of-The-Day: Kiwi Fruit"). All fields shall be unfolded. There shall be no space before the ":". The reverse mapping builds the RFC 822 field in a straightforward manner. This RFC822Field is appended to the RFC822FieldList, which is added to the IPM Heading as an extension field.

5.1.3. Generating the IPM

The IPM (IPMS Service Request) is generated according to the rules of this section. The IPMS.IPM.body is generated from the RFC 822 message body in the manner described in Section 5.1.5.

If no specific 1988 features are used, the IPM generated is encoded as content type 2. Otherwise, it is encoded as content type 22. The latter will always be the case if extension heading fields are generated.

When generating the IPM, the issue of upper bounds are handled as follows. Truncate fields to the upper bounds specified in X.400. This will prevent problems with UAs which enforce upper bounds, but will sometimes discard useful information. This approach will cause more problems for some fields than others (e.g., truncating an OR Address component that would be used to route a reply would be a more severe problem than truncating a Free Form Name). If the Free Form name is truncated, it shall be done so that it does not break RFC 822 comments and RFC 1522 encoding.

Note: This approach removes a choice of options given in RFC 1327, based on operational experience.

The rest of this section concerns IPMS.IPM.heading (IPMS.Heading). The only mandatory component of IPMS.Heading is the IPMS.Heading.this-IPM (IPMS.IPMIdentifier). A default is generated by the gateway. With the exception of "Received:", the values of multiple fields are merged (e.g., If there are two "To:" fields, then the mailboxes of both are merged to generate a single list which is used in the IPMS.Heading.primary-recipients. Information shall be generated from the standard RFC 822 Headers as follows:

Date:
: Ignore (Handled at MTS level)

Received:
: Ignore (Handled at MTA level)

Message-Id:
: Mapped to IPMS.Heading.this-IPM. For these, and all other fields containing 822.msg-id the mappings of Chapter 4 are used for each 822.msg-id.

From:
: If Sender: is present, this is mapped to IPMS.Heading.authorizing-users. If not, it is mapped to IPMS.Heading.originator. For this, and other components containing addresses, the mappings of Chapter 4 are used for each address.

Sender:
: Mapped to IPMS.Heading.originator. Because X.400 does not have the same From/Sender distinction as RFC 822, this mapping is not always natural and may lead to unexpected results in some cases.

Reply-To:
: Mapped to IPMS.Heading.reply-recipients.

To: Mapped to IPMS.Heading.primary-recipients

Cc: Mapped to IPMS.Heading.copy-recipients.

Bcc: Mapped to IPMS.Heading.blind-copy-recipients if there is at least one BCC: recipient. If there are no recipients in this field, it shall either be mapped to a zero length sequence or mapped to a single recipient that has a free from name "BCC" and no other addressing information. This alternate treatment is allowed because some X.400 systems cannot handle a zero lenght sequence of addresses.

In-Reply-To:
If there is one value, it is mapped to IPMS.Heading.replied-to-IPM, using the 822.phrase or 822.msg-id mapping as appropriate. If there are multiple values, this cannot be done as the X.400 heading is single valued. In this case no IPMS.Heading.replied-to-IPM is generated and the values are mapped to IPMS.Heading.related-IPMs, along with any values from a "References:" field.

References:
Mapped to IPMS.Heading.related-IPMs.

Keywords:
Mapped onto a heading extension.

Subject:
Mapped to IPMS.Heading.subject. The field-body uses the human oriented mapping referenced in Section 3.3.4.

Comments:
Mapped onto a heading extension.

This is a change from 1327, which specified to generate an IPMS.BodyPart of type IPMS.IA5TextBodyPart with IPMS.IA5TextBodyPart.parameters.repertoire set to the default (ia5), containing the value of the fields, preceded by the string "Comments: " and that this body part shall precede the other one. Experience has shown that this complexity is not justified. This text is retained to facilitate backwards compatibility.

Encrypted:
Mapped onto a heading extension.

Resent-*
Mapped onto a heading extension.

Note that it would be possible to use a ForwardedIPMessage for these fields, but the semantics are (arguably) slightly different, and it is probably not worth the effort.

Content-Language:
This field is defined in RFC 1766 [7]. Map the first two characters of each value given onto the IPM Languages extension. If any comments or values longer than two characters occur, a header extension shall also be generated.

Other Fields
In particular X-* fields, and "illegal" fields in common usage (e.g., "Fruit-of-the-day:") are mapped onto a heading extension, unless covered by another section or appendix of this specification. The same treatment is applied to RFC 822 fields where the content of the field does not conform to RFC 822 (e.g., a Date: field with unparseable syntax).

The mapping of the following headings is defined in RFC 2157.

MIME-Version: 5
Content-Transfer-Encoding:
Content-Type
Content-ID
Content-Description

5.1.4. Generating the IPM Body

Generation of the IPM Body is defined in RFC 2157.

5.1.5. Mappings to the MTS Abstract Service

The MTS.MessageSubmissionEnvelope comprises MTS.PerMessageSubmissionFields, and MTS.PerRecipientMessageSubmissionFields. The mandatory parameters are defaulted as follows.

MTS.PerMessageSubmissionFields.originator-name
This is always generated from SMTP, as defined in Chapter 4.

MTS.PerMessageSubmissionFields.content-type
Set to the value implied by the encoding of the IPM (2 or 22).

MTS.PerRecipientMessageSubmissionFields.recipient-name
These will always be supplied from SMTP, as defined in Chapter 4.

Optional components are omitted, and default components defaulted. This means that disclosure of recipients is prohibited and conversion is allowed. There are two exceptions to the defaulting. For MTS.PerMessageSubmissionFields.per-message-indicators, the following settings are made:

- Alternate recipient is allowed, as it seems desirable to maximise the opportunity for (reliable) delivery.

If SMTP is used, Appendix A shall be followed in setting these parameters.

The trace is set to indicate conversion (described below) and the encoded information types in the trace is derived from the message generated by the gateway, and shall reflect all body parts (including those in enclosed messages). In addition it shall include the Encoded Information Type "eit-mixer", which is defined in Appendix D. The presence of the EIT will indicate to the X.400 recipient that a MIXER conversion has occurred. MTS.PerMessageSubmissionFields.original-encoded-information-types will include all of the values used in the trace, unless specified otherwise in RFC 2157.

This type of conversion will prevent the normal loop detection from working in certain circumstances, and introduces the possiblity of gateway loops. MIXER gateways shall therefore count the number of MIXER conversions made. If this count exceeds five in one direction, the message shall be treated as if a loop has been detected.

The MTS.PerMessageSubmissionFields.content-correlator is encoded as IA5String, and contains the Subject:, Message-ID:, Date:, and To: fields (if present) in this order. This includes the strings "Subject:", "Date:", "To:", "Message-ID:", and appropriate folding to make the field appear readable. This shall be truncated to MTS.ub-content-correlator-length (512) characters. In addition, if there is a "Subject:" field, the MTS.PerMessageSubmissionFields.content-identifier, is set to a printable string representation of the contents of it. If the length of this string is greater than MTS.ub-content-id-length (16), it shall be truncated to 13 characters and the string "..." appended. Both are used, due to the much larger upper bound of the content correlator, and that the content id is available in X.400(1984).

5.1.6. Mappings to the MTA Abstract Service

There is a need to map directly onto some aspects of the MTA Abstract service, for the following reasons:

- So the MTS Message Identifier can be generated from the RFC 822 Message-ID:.
- So that the submission date can be generated from the 822.Date.
- To prevent loss of trace information
- To prevent RFC 822/X.400 looping caused by distribution lists or redirects

The following mappings are defined.

Message-Id:

If this is present and no Resent: fields are present, the MTA.PerMessageTransferFields.message-identifier may be generated from it, using the mappings described in Chapter 4.

This mapping arguably generates messages which do not conform to US GOSIP (1984 version only), which states:

6.7.e MPDU Identifier Validation

(1) Validation of the GlobalDomainIdentifier component of the MPDU Identifier is performed on reception of a message (i.e. the result of a TRANSFER.Indication).

(2) The country name should be known to the validating domain, and depending on the country name, validation of the

ADMD name may also be possible.

(3) Additional validation of the GlobalDomainIdentifier is performed against the corresponding first entry in the TraceInformation. If inconsistencies are found during the comparison, a non-delivery notice with the above defined reason and diagnostic code is generated.

(4) A request will be generated to the CCITT for a more meaningful diagnostic code (such as "InconsistentMPUTIdentifier").

This applies to ADMDs only, and is specified in the 1984 version and not the 1988 version. Conformance depends on the interpretation of "inconsistency". The specification makes the most sensible choice, and so is not being changed in the update from RFC 1327.

Date: (and Resent-Date:)

If one or more Resent-Date: fields is present, the most recent Resent-Date: field shall be used instead of the Date: field in the following description.

The Date: field is used to set the first component of MTA.PerMessageTransferFields.trace-information (MTA.TraceInformationElement). The SMTP originator is mapped into an MTS.ORAddress, and used to derive MTA.TraceInformationElement.global-domain-identifier. The optional components of MTA.TraceInformationElement.domain-supplied-information are omitted, and the mandatory components are set as follows:

MTA.DomainSuppliedInformation.arrival-time

This is set to the date derived from Date:

MTA.DomainSuppliedInformation.routing-action

Set to relayed.

The first element of MTA.PerMessageTransferFields.internal-trace-information is generated in an analogous manner, although this can be dropped later in certain circumstances (see the procedures for "Received:"). The MTA.InternalTraceInformationElement.mta-name is derived from the 822.domain in the 822 MTS Originator address.

Received:

All RFC 822 trace is used to derive MTA.PerMessageTransferFields.trace-information and MTA.PerMessageTransferFields.internal-trace-information. Processing of Received: lines follows processing of Date:, and is done from the bottom to the top of the RFC 822 header (i.e., in chronological order). When other trace elements (in particular X400-Received:) are processed the relative ordering (top to bottom of the header) shall be retained correctly.

The initial element of MTA.PerMessageTransferFields.trace-information shall be generated from Date: as described above, unless the message has previously been in X.400, when it will be derived from the X.400 trace information.

For each Received: field, the following processing shall be done. If the "by" part of the received is present and there is an available MCGAM which can map this domain, use it to derive an MTS.GlobalDomainIdentifier. Otherwise MTS.GlobalDomainIdentifier is set from local information. If this is different from the one in the last element of MTA.PerMessageTransferFields.trace-information (MTA.TraceInformationElement.global-domain-identifier) create a new MTA.TraceInformationElement, and optionally remove MTA.PerMessageTransferFields.internal-trace-information. Requirements on trace stripping are discussed below.

Then add a new element (MTA.InternalTraceInformationElement) to MTA.PerMessageTransferFields.internal-trace-information, creating this if needed. This shall be done, even if nter-MD trace is created. The MTA.InternalTraceInformationElement.global-domain-identifier is set to the value derived. The MTA.InternalTraceInformationElement.mta-supplied-information (MTA.MTASuppliedInformation) is set as follows:

MTA.MTASuppliedInformation.arrival-time
Derived from the date of the Received: line

MTA.MTASuppliedInformation.routing-action
Set to relayed

The MTA.InternalTraceInformationElement.mta-name is taken from the "by" component of the "Received:" field, truncated to MTS.ub-mta-name-length (32). For example:

Received: from computer-science.nottingham.ac.uk by
vs6.Cs.Ucl.AC.UK via Janet with NIFTP id aa03794;
28 Mar 89 16:38 GMT

Generates the string

vs6.Cs.Ucl.AC.UK

The gateway shall add in a single element of trace information, reflecting the gateway's local information and the time of conversion. The MTA.InternalTraceInformationElement.mta-supplied-information (MTA.MTASuppliedInformation) is set as follows:

MTA.DomainSuppliedInformation.arrival-time
Set to the time of conversion

MTA.DomainSuppliedInformation.routing-action
Set to relayed

MTA.AdditionalActions.converted-encoded-information-types Set to correct set of EITs for the message that is generated by the gateway. This trace element will thus reflect gateway operation as a conversion.

This trace generation will often lead to generation of substantial amounts of trace information, which does not reflect X.400 transfers. Stripping of some of this trace may be necessary in some operational environments. This stripping shall be considered a function of the associated X.400 MTA, and not of the MIXER gateway.

5.1.7. Mapping New Fields

This specification defines a number of new fields for Reports, Notifications and IP Messages. A gateway conforming to this specification shall map all of these fields to X.400, except as defined below.

The mapping of two extended fields is particularly important, in order to prevent looping. "DL-Expansion-History:" is mapped to MTA.PerMessageTransferFields.extensions.dl-expansion-history X400-Received: shall be mapped to MTA.PerMessageTransferFields.trace-information and MTA.PerMessageTransferFields.internal-trace-information. In cases where X400-Received: is present, the usual mapping of Date: to generate the first element of trace shall not be done. This is because the message has come from X.400, and so the first element of trace can be taken from the first X400-Received:.

The following fields shall not be mapped, and shall be

- Discarded-X400-MTS-Extensions:
- Message-Type:
- Discarded-X400-IPMS-Extensions:
- X400-Content-Type:
- X400-Originator:
- X400-Recipients:
- X400-MTS-Identifier: Mapping this field would be useful in some circumstances, but very dangerous in others (e.g., following an internet list expansion). Therefore it is not mapped.

5.1.8. Mapping Delivery Status Notifications to X.400

5.1.8.1. Basic Model

Internet Mail delivery status notifications (DSN) are mapped to X.400 delivery reports. With message mapping, information without a mapping is carried by an IPM Extension. This cannot be done for delivery reports. Two mechanisms are used for information where there is not a direct mapping.

The first mechanism is to define extensions, which allow all of the DSN information to be carried in the delivery report. This is not completely satisfactory for two reasons:

1. User defined extensions are supported by the ISO version of the standard, but not the CCITT one. Therefore, implementation support for these extensions will not be universal.

2. X.400 User Agent implementations will not in general recognise these extensions. Therefore, although the information will be present, it will often not be available to the user. This may be very problematic, as this information may be critical to diagnosing the reason for a failure.

Therefore a second mechanism is defined. This shall always be used when the DSN contains non-delivery information, and may be used in other cases. This mechanism is to map the whole DSN (as if it were an ordinary multipart) into the return of content. This will make the DSN information available as a text body part in the outer message, with the real returned content as an enclosed message. This mechanism will ensure that information is not lost at the gateway.

5.1.8.2. DSN Extensions

Two X.400 MTS extensions are defined as follows:

```
dsn-header-list EXTENSION
   RFC822FieldList
   ::= id-dsn-header-list

dsn-field-list EXTENSION
   RFC822FieldList
   ::= id-dsn-field-list
```

The Object Identifiers id-dsn-header-list and id-dsn-field-list are defined in Appendix D. Theses extensions are used in the same way as the IPM extension rfc-822-field, described in Section 5.1.2. These extensions may only be used with ISO-10021, and not X.400 (which does not allow user extensions at the MTS level).

5.1.8.3. DSN to Delivery Report Mapping

Some DSNs are mapped to Delivery Reports and some to IPMs, according to the value of the action field. The mapping to an IPM is exactly as for a normal IPM mapping. The choice of IPM and Delivery report is made for each reported recipient. If this choice is different for different reported recipients both a Delivery Report and an IPM shall be generated.

Reports are not be submitted in the X.400 model, and so the report submission is considered in terms of the MTA Abstract Service. An MTA.Report is constructed. The MTA.ReportTransferEnvelope.report-identifier is generated from the Message-Id of the DSN (if present) and otherwise generated as the MTA would generate one for a submitted message.

The DSN has an RFC 822 header. Trace is mapped in the same manner as for a message to MTA.ReportTransferEnvelope.trace-information. All other headers are used to create a dsn-header-list extension, which is added to MTA.PerReportTransferFields.extensions. The DSN will have a single SMTP recipient. This is mapped to the MTA.ReportTransferEnvelope.report-destination-name.

The DSN is then treated as a normal MIME message, and an X.400 IPM is generated. This IPM is used as MTA.PerReportTransferFields.returned-content, and its type is used to set MTA.PerReportTransferFields.content-type. The DSN body part is mapped as if it was IA5 text/plain.

The mandatory MTA.PerReportTransferFields.subject-identifier shall be generated from the DSN.per-message-field original-envelope-id, if this starts with the string "X400-MTS-Identifier: ", and derived from the rest of the field, which is encoded as EBNF.mts-msg-id. In other cases, this field shall be generated by the MIXER Gateway.

All other mappings are made from the DSN body part. A dsn-field-list extension is created and added to MTA.ReportTransferFields.extensions. This is referred to as the per report extension list. The DSN.per-message-fields are mapped as follows:

original-envelope-id-field
reporting-mta-field
dsn-gateway-field
received-from-mta-field
arrival-date-field
extension-field
other

All of these fields are added to the per report extension list. Currently there are no other mappings defined.

Each reported recipient is considered in turn, and a MTA.PerRecipientReportTransferFields created for each. The parameters of this are defaulted as follows:

originally-specified-recipient-number
In general, these are not available, and so are assigned incrementally.

last-trace-information
The arrival-time is generated from DSN.arrival-date if present, and if not from the Date: of the DSN. This is a strucutred field, and the Report element contains the key information on the recipient. For a DeliveryReport, the type-ofMTS-user is defaulted to public and the message-deliery-time is set to the same as the arrival-time. For a NonDeliveryReport, the code mappings are define in Section 5.1.8.4.

A dsn-field-list extension is created and added to MTA.PerRecipientTransferFields.extensions. This is referred to as the per recipient extension list. The DSN.per-recipient-fields are mapped as follows

original-recipient-field
Mapped to MTA.PerRecipientReportTransferFields.originally-intended-recipient-name.

final-recipient-field
Mapped to MTA.PerRecipientReportTransferFields.actual-recipient-name.

action-field
If this is set to "failed", a non-delivery report is generated. If this is set to "delivered" a delivery report is generated. Bit one or two of MTA.PerRecipientTransferFields.per-recipient-indicators is set accordingly. This also controls the encoding of MTA.PerRecipientTransferFields.last-trace-information, and the selection of the report type.

For other values of the action-field ("delayed", "relayed", "expanded"), an IPM is generated. This enables the status information to be communicated to the X.400 user, without the confusion of multiple delivery reports.

status-field
This is added to the per report extension list. For non-delivery, it is also used to generate the reason and diagnostic codes contained within MTA.PerRecipientReportTransferFields.last-trace. The mappings are defined below.

remote-mta-field

diagnostic-code-field

last-attempt-date-field

will-retry-until-field

extension-field

other
All of these fields are added to the per recipient extension list.

5.1.8.4. Status Value Mappings

Status values are mapped to X.400 reason and diagnostic codes as follows.

If a status value is found that is not in this table, the gateway may use the same mapping as for "X.n.0" (1/None or 0/None), or it may map to another, configurable code. Implementors are requested to forward new codes to the mixer list for inclusion in future versions of this standard. So for instance. "5.2.37", currently undefined, would map onto the same as "5.2.0", namely 1/None.

DSN code	Meaning	X400 code	Meaning
X.0.0	Other status	1/None	
X.1.0	Other Address Status	1/None	
X.1.1	Bad mailbox address	1/0	Unrecognized
X.1.2	Bad system address	1/0	Unrecognized
X.1.3	Bad mailbox address syntax	1/0	Unrecognized
X.1.4	Mailbox address ambiguous	1/1	
X.1.5	Only used for positive reports, not applicable		
X.1.6	Destination mailbox has moved	1/43	New addr unknown
X.1.7	Bad sender’s mailbox address syntax	1/11	Invalid arguments
X.1.8	Bad sender’s system address	1/11	Invalid arguments
X.2.0	Other or undefined mailbox status	1/None	
X.2.1	Mailbox disabled, not accepting	1/4	Recipient unavail
X.2.2	Mailbox full	1/4	
X.2.3	Message length exceeds admin limit.	1/7	Content too long
X.2.4	Mailing list expansion problem	1/30	DL expansion fail
X.3.0	Other or undefined system status	0/None	
X.3.1	System full	1/2	MTS congestion
X.3.2	System not accepting network messages	1/2	MTS congestion
X.3.3	System not capable of selected feat	1/18	Unsupp crit func
X.3.4	Message too big for system	1/7	
X.3.5	System incorrectly configured	1/None	
X.4.0	Other or undefined network or routing	0/None	
X.4.1	No answer from host	0/None	
X.4.2	Bad connection	0/None	
X.4.3	Routing server failure	6/None	Dir op unsucc.
X.4.4.	Unable to route	0/None	
X.4.5	Network congestion	1/2	MTS congest.
X.4.6	Routing loop detected	1/3	
X.4.7	Delivery time expired	1/5	
X.5.0	Other or undefined protocol status	1/None	
X.5.1	Invalid command	1/14	Protocol viol.
X.5.2	Syntax error	1/14	
X.5.3	Too many recipients	1/16	
X.5.4	Invalid command arguments	1/14	
X.5.5	Wrong protocol version	1/18	Unsupp.crit.func

X.6.0	Other or undefined media error	2/None	Conv. not perf
X.6.1	Media not supported	1/6	EIT unsupp.
X.6.2	Conversion required and prohibited	1/9	
X.6.3	Conversion required but not supported	2/8	
X.6.4	Conversion with loss performed	POSITIVE only	
X.6.5	Conversion failed	2/47	Unable to downgrade
X.7.0	Other or undefined security status	1/46	
X.7.1	Delivery not authorized, message ref	1/29	No DL submit perm
X.7.2	Mailing list expansion prohibited	1/28	
X.7.3	Security conversion req but not poss	1/46	Secure mess. error
X.7.4	Security features not supported	1/46	
X.7.5	Cryptographic failure	1/46	
X.7.6	Cryptographic algorithm not supported	1/46	
X.7.7	Message integrity failure	1/46	

5.1.8.5. DSNs that originated in X.400

The mapping of X.400 delivery reports to DSNs will in general provide sufficient information to make a useful reverse mapping. Messages will often be mapped multiple times, commonly due to forwarding messages and to distribution lists. Multiple mappings for delivery reports will be a good deal less common. For this reason, the reverse mapping of the X.400 DSN extensions defined in MIXER is optional.

5.2. Return of Contents

RFC 1327 offered two approaches for return of content, as this service is optional in X.400 and expected in RFC 822. MIXER simply requires that a gateway requests the return of content service from X.400.

5.3. X.400 -> RFC 822: Detailed Mappings

5.3.1. Basic Approach

A single RFC 822 message is generated from the incoming IP Message, Report, or IP Notification. All IPMS.BodyParts are mapped onto a single RFC 822 body. Other services are mapped onto RFC 822 header fields. Where there is no appropriate existing field, new fields are defined for IPMS, MTS and MTA services.

The gateway mechanisms will correspond to MTS Delivery. As with submission, there are aspects where the MTA (transfer) services are also used. In particular, there is an optimisation to allow for multiple SMTP recipients.

5.3.2. RFC 822 Settings

An RFC 822 Message has a number of mandatory fields in the RFC 822 Header. Some SMTP services mandate specification of an SMTP Originator. Even in cases where this is optional, it is usually desirable to specify a value. The following defaults are defined, which shall be used if the mappings specified do not derive a value:

SMTP Originator
: If this is not generated by the mapping (e.g., for a Delivery Report), a value pointing at a gateway administrator shall be assigned.

Date:
: A value will always be generated

From:
: If this is not generated by the mapping, it is assigned equal to the SMTP Originator. If this is gateway generated, an appropriate 822.phrase shall be added.

At least one recipient field
: If no recipient fields are generated, a field "To: list:;", shall be added.

This will ensure minimal RFC 822 compliance. When generating RFC 822 headers, folding may be used. It is recommended to do this, following the guidelines of RFC 822.

5.3.3. Basic Mappings

5.3.3.1. Encoded Information Types

This mapping from MTS.EncodedInformationTypes is needed in several disconnected places. EBNF is defined as follows:

```
encoded-info    = 1#encoded-type

encoded-type    = built-in-eit / object-identifier
```

```
      built-in-eit    = "Undefined"         ; undefined (0)
                      / "Telex"             ; tLX (1)
                      / "IA5-Text"          ; iA5Text (2)
                      / "G3-Fax"            ; g3Fax (3)
                      / "TIF0"              ; tIF0 (4)
                      / "Teletex"           ; tTX (5)
                      / "Videotex"          ; videotex (6)
                      / "Voice"             ; voice (7)
                      / "SFD"               ; sFD (8)
                      / "TIF1"              ; tIF1 (9)
```

MTS.EncodedInformationTypes is mapped onto EBNF.encoded-info. MTS.EncodedInformationTypes.non-basic-parameters is ignored. Built in types are mapped onto fixed strings (compatible with X.400(1984) and RFC 987), and other types are mapped onto EBNF.object-identifier.

5.3.3.2. Global Domain Identifier

The following simple EBNF is used to represent MTS.GlobalDomainIdentifier:

```
   global-id = std-or-address
```

This is encoded using the std-or-address syntax, for the attributes within the Global Domain Identifier.

5.3.4. Mappings from the IP Message

Consider that an IPM has to be mapped to RFC 822. The IPMS.IPM comprises an IPMS.IPM.heading and IPMS.IPM.body. The heading is considered first. Some EBNF for new fields is defined:

```
ipms-field = "Supersedes" ":" 1*msg-id
           / "Expires" ":" date-time
           / "Reply-By" ":" date-time
           / "Importance" ":" importance
           / "Sensitivity" ":" sensitivity
           / "Autoforwarded" ":" boolean
           / "Incomplete-Copy" ":"
           / "Content-Language" ":" 1#language
           / "Message-Type" ":" message-type
           / "Discarded-X400-IPMS-Extensions" ":" 1#object-identifier
           / "Autosubmitted" ":" autosubmitted


importance      = "low" / "normal" / "high"
```

```
sensitivity     = "Personal" / "Private" /
                  "Company-Confidential"

language        = 2*ALPHA [ "(" language-description ")" ]
     language-description = printable-string


message-type    = "Delivery Report"
                / "InterPersonal Notification"
                / "Multiple Part"

autosubmitted   = "not-auto-submitted"
                / "auto-generated"
                / "auto-replied"
                / "auto-forwarded"
```

The mappings and actions for the IPMS.Heading are now specified for each element. Addresses and Message Identifiers are mapped according to Chapter 4. Other mappings are explained, or are straightforward (algorithmic). If a field with addresses contains zero elements, it shall be discarded, except for IPMS.Heading.blind-copy-recipients, which can be mapped onto BCC: (the only RFC 822 field which allows zero recipients).

IPMS.Heading.this-IPM
Mapped to "Message-ID:".

IPMS.Heading.originator
If IPMS.Heading.authorizing-users is present this is mapped to Sender:, if not to "From:".

IPMS.Heading.authorizing-users
Mapped to "From:".

IPMS.Heading.primary-recipients
Mapped to "To:".

IPMS.Heading.copy-recipients
Mapped to "Cc:".

IPMS.Heading.blind-copy-recipients
Mapped to "Bcc:".

IPMS.Heading.replied-to-ipm
Mapped to "In-Reply-To:".

IPMS.Heading.obsoleted-IPMs
Mapped to the extended RFC 822 field "Supersedes:". The replaces the RFC 1327 field "Obsoletes:". Reverse mapping of the RFC 1327 field may be supported.

IPMS.Heading.related-IPMs
Mapped to "References:".

IPMS.Heading.subject
Mapped to "Subject:". The contents are converted to ASCII or T.61 (as defined in Section 3.5). CRLF will not be present in a valid X.400 field. Any CRLF present are not mapped, but are used as points at which the subject field shall be folded, unless an RFC 1522 encoding is used.

IPMS.Heading.expiry-time
Mapped to the extended RFC 822 field "Expires:". The replaces the RFC 1327 field "Expiry-Date:". Reverse mapping of the RFC 1327 field may be supported.

IPMS.Heading.reply-time
Mapped to the extended RFC 822 field "Reply-By:".

IPMS.Heading.reply-recipients
Mapped to "Reply-To:".

IPMS.Heading.importance
Mapped to the extended RFC 822 field "Importance:".

IPMS.Heading.sensitivity
Mapped to the extended RFC 822 field "Sensitivity:".

IPMS.Heading.autoforwarded
Mapped to the extended RFC 822 field "Autoforwarded:".

The standard extensions (Annex H of X.420 / ISO 10021-7) are mapped as follows:

incomplete-copy
Mapped to the extended RFC 822 field "Incomplete-Copy:".

language
Mapped to the RFC 822 field "Content-Language:", defined in RFC 1766 [7]. This mapping may be made without loss of information.

auto-submitted
Map to the extended RFC 822 field "Autosubmitted:".

If the RFC 822 extended header is found, this shall be mapped onto an RFC 822 header, as described in Section 5.1.2.

If a non-standard extension is found, it shall be discarded, unless the gateway understands the extension and can perform an appropriate mapping onto an RFC 822 header field. If extensions are discarded, the list is indicated in the extended RFC 822 field "Discarded-X400-IPMS-Extensions:".

5.3.4.1. Mapping the IPMS Body

The mapping of the IPMS Body is defined in RFC 2157.

5.3.4.2. Example Message

An example message, illustrating a number of aspects is given below.

```
Received: from mhs-relay.ac.uk by bells.cs.ucl.ac.uk via JANET with
          NIFTP id <7906-0@bells.cs.ucl.ac.uk>;
          Thu, 30 May 1991 18:24:55 +0100
X400-Received: by mta "mhs-relay.ac.uk" in /PRMD=uk.ac/ADMD= /C=gb/;
               Relayed; Thu, 30 May 1991 18:23:26 +0100
X400-Received: by /PRMD=HMG/ADMD=GOLD 400/C=GB/; Relayed;
               Thu, 30 May 1991 18:20:27 +0100
Message-Type: Multiple Part
Date: Thu, 30 May 1991 18:20:27 +0100
X400-Originator: Stephen.Harrison@gosip-uk.hmg.gold-400.gb
X400-MTS-Identifier:
     [/PRMD=HMG/ADMD=GOLD 400/C=GB/;PC1000-910530172027-57D8]
Original-Encoded-Information-Types: ia5
X400-Content-Type: P2-1984 (2)
X400-Content-Identifier: Email Problems
From: Stephen.Harrison@gosip-uk.hmg.gold-400.gb (Tel +44 71 217 3487)
Message-ID: <PC1000-910530172027-57D8*@MHS>
To: Jim Craigie <NTIN36@gec-b.rutherford.ac.uk>,
    Tony Bates <tony@ean-relay.ac.uk>,
    Steve Kille <S.Kille@cs.ucl.ac.uk>
Subject: Email Problems
Sender: Stephen.Harrison@gosip-uk.hmg.gold-400.gb
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=boundary-1

--boundary-1
Content-Type: text/plain; charset=US-ASCII

Hope you gentlemen.......
```

```
Regards,

Stephen Harrison
UK GOSIP Project

--boundary-1
Content-Type: message/rfc822

From: Urs Eppenberger <Eppenberger@verw.switch.ch>
Message-ID:
<562*/S=Eppenberger/OU=verw/O=switch/PRMD=SWITCH/ADMD=ARCOM/C=CH/@MHS>
To: "Stephen.Harrison" <Stephen.Harrison@gosip-uk.hmg.gold-400.gb>
Cc: kimura@bsdarc.bsd.fc.nec.co.jp
Subject: Response to Email link
Content-Type: multipart/mixed; boundary=boundary-2


--boundary-2

Dear Mr Harrison......


--boundary-2--

--boundary-1--
```

5.3.5. Mappings from an IP Notification

Because of the service setting, IP Notifications will not usually need to be mapped by a MIXER gateway. A message is generated, with the following fields:

From:
Set to the IPMS.IPN.ipn-originator.

To: Set to the recipient from MTS.MessageSubmissionEnvelope. If there have been redirects, the original address shall be used.

Subject:
Set to the string "X.400 Inter-Personal Notification" for a receipt notification and to "X.400 Inter-Personal Notification (failure)" for a non-receipt notification.

Message-Type:
Set to "InterPersonal Notification"

References:
Set to IPMS.IPN.subject-ipm

Discarded-X400-IPMS-Extensions:
Used for any discarded IPN extensions.

The following EBNF is defined for the body of the Message. This format is defined to ensure that all information from an interpersonal notification is available to the end user in a uniform manner.

```
ipn-body-format = ipn-description <CRLF>
                  [ ipn-extra-information <CRLF> ]
                  [ ipn-content-return ]

ipn-description = ipn-receipt / ipn-non-receipt

ipn-receipt = "Your message to:" preferred-recipient <CRLF>
          "was received at" receipt-time <CRLF> <CRLF>
          "This notification was generated"
           acknowledgement-mode <CRLF>
          "The following extra information was given:" <CRLF>
           ipn-suppl <CRLF>

ipn-non-receipt = "Your message to:"
              preferred-recipient <CRLF>
              ipn-reason

ipn-reason = ipn-discarded / ipn-auto-forwarded

ipn-discarded = "was discarded for the following reason:"
                 discard-reason <CRLF>

ipn-auto-forwarded = "was automatically forwarded." <CRLF>
                     [ "The following comment was made:"
                     auto-comment ]


ipn-extra-information =
       "The following information types were converted:"
        encoded-info

ipn-content-return = "The Original Message is not available"
                     / "The Original Message follows:"

preferred-recipient = mailbox
receipt-time        = date-time
auto-comment        = printablestring
ipn-suppl           = printablestring
```

```
      discard-reason     = "Expired" / "Obsoleted" /
                  "User Subscription Terminated" / "IPM Deleted"

      acknowledgement-mode = "Manually" / "Automatically"
```

The mappings for elements of the common fields of IPMS.IPN (IPMS.CommonFields) onto this structure and the message header are:

subject-ipm
Mapped to "References:"

ipn-originator
Mapped to "From:".

ipn-preferred-recipient
Mapped to EBNF.preferred-recipient

conversion-eits
Mapped to EBNF.encoded-info in EBNF.ipn-extra-information

The mappings for elements of IPMS.IPN.non-receipt-fields (IPMS.NonReceiptFields) are:

non-receipt-reason
Used to select between EBNF.ipn-discarded and EBNF.ipn-auto-forwarded

discard-reason
Mapped to EBNF.discard-reason

auto-forward-comment
Mapped to EBNF.auto-comment

returned-ipm
This applies only to non-receipt notifications. EBNF.ipn-content-return shall always be omitted for receipt notifications, and always be present in non-receipt notifications. If present, the second option of EBNF.ipn-content-return is chosen, and the message is included. In this case, the message is formatted as multipart/mixed, and the returned message included as message/rfc822 after the text body part. Otherwise the first option is chosen.

The mappings for elements of IPMS.IPN.receipt-fields (IPMS.ReceiptFields) are:

receipt-time
Mapped to EBNF.receipt-time

acknowledgement-mode
Mapped to EBNF.acknowledgement-mode

suppl-receipt-info
Mapped to EBNF.ipn-suppl

An example notification is:

```
From: Steve Kille <steve@cs.ucl.ac.uk>
To: Julian Onions <jpo@computer-science.nottingham.ac.uk>
Subject: X.400 Inter-personal Notification
Message-Type: InterPersonal Notification
References: <1229.614418325@UK.AC.NOTT.CS>
Date: Wed, 21 Jun 89 08:45:25 +0100

Your message to: Steve Kille <steve@cs.ucl.ac.uk>
was automatically forwarded.
The following comment was made:
   Sent on to a random destination

The following information types were converted: g3fax
```

5.3.6. Mappings from the MTS Abstract Service

This section describes the MTS mappings for User Messages (IPM and IPN). This mapping is defined by specifying the mapping of MTS.MessageDeliveryEnvelope. The following extensions to RFC 822 are defined to support this mapping:

```
mts-field = "X400-MTS-Identifier" ":" mts-msg-id
          / "X400-Originator" ":" mailbox
          / "X400-Recipients" ":" 1#mailbox
          / "Original-Encoded-Information-Types" ":"
            encoded-info
          / "X400-Content-Type" ":" mts-content-type
          / "X400-Content-Identifier" ":" printablestring
          / "Priority" ":" priority
          / "Originator-Return-Address" ":" 1#mailbox
          / "DL-Expansion-History" ":" mailbox ";" date-time
            ";"
          / "Conversion" ":" prohibition
          / "Conversion-With-Loss" ":" prohibition
          / "Delivery-Date" ":" date-time
          / "Discarded-X400-MTS-Extensions" ":"
            1#( object-identifier / labelled-integer )


prohibition     = "Prohibited" / "Allowed"
```

```
mts-msg-id       = "[" global-id ";" *text "]"

mts-content-type = "P2" /  labelled-integer
                 / object-identifier

priority        = "normal" / "non-urgent" / "urgent"
```

The mappings for each element of MTS.MessageDeliveryEnvelope can now be considered. Where the specified action does not result in an extended element being mapped, the criticality associated with this element shall be considered. If the element is marked as critical for transfer or for delivery, the message shall be non delivered by the gateway because a critical extension cannot be correctly handled.

MTS.MessageDeliveryEnvelope.message-delivery-identifier
Mapped to the extended RFC 822 field "X400-MTS-Identifier:".

MTS.MessageDeliveryEnvelope.message-delivery-time
Discarded, as this time will be represented in an appropriate trace element.

The mappings for elements of MTS.MessageDeliveryEnvelope.other-fields (MTS.OtherMessageDeliveryFields) are:

content-type
Mapped to the extended RFC 822 field "X400-Content-Type:". The string "P2" is retained for backwards compatibility with RFC 987. This shall not be generated, and either the EBNF.labelled-integer or EBNF.object-identifier encoding used.

originator-name
Mapped to the SMTP originator, and to the extended RFC 822 field "X400-Originator:". This is described in Section 4.6.2.

original-encoded-information-types
Mapped to the extended RFC 822 field "Original-Encoded-Information-Types:".

priority
Mapped to the extended RFC 822 field "Priority:".

delivery-flags
If the conversion-prohibited bit is set, add an extended RFC 822 field "Conversion:".

this-recipient-name and other-recipient-names
The handling of these elements is described in Section 4.6.2.

originally-intended-recipient-name
The handling of this element is described in Section 4.6.2.

converted-encoded-information-types
Discarded. This information will be mapped in the trace.

message-submission-time
Mapped to Date:.

content-identifier
Mapped to the extended RFC 822 field "X400-Content-Identifier:". In RFC 1327, this was "Content-Identifier:". This has been changed to avoid confusion with MIME defined fields. Gateways which reverse map, may support the old field.

If any extensions (MTS.MessageDeliveryEnvelope.other-fields.extensions) are present, and they are marked as critical for transfer or delivery, then the message shall be rejected. The extensions (MTS.MessageDeliveryEnvelope.other-fields.extensions) are mapped as follows.

conversion-with-loss-prohibited
If set to MTS.ConversionWithLossProhibited.conversion-with-loss-prohibited, then add the extended RFC 822 field "Conversion-With-Loss:".

requested-delivery-method
Mapped to a comment, as described in Section 4.6.2.2.

originator-return-address
Mapped to the extended RFC 822 field "Originator-Return-Address:".

physical-forwarding-address-request
physical-delivery-modes
registered-mail-type
recipient-number-for-advice
physical-rendition-attributes
physical-delivery-report-request
physical-forwarding-prohibited

These elements are only appropriate for physical delivery. They are represented as comments in the "X400-Recipients:" field, as described in Section 4.6.2.2.

originator-certificate
message-token
content-confidentiality-algorithm-identifier

content-integrity-check
message-origin-authentication-check
message-security-label
proof-of-delivery-request

These elements imply use of security services not available in the RFC 822 environment. If they are marked as critical for transfer or delivery, then the message shall be rejected. Otherwise they are discarded.

redirection-history
This is described in Section 4.6.2.

dl-expansion-history
Each element is mapped to an extended RFC 822 field "DL-Expansion-History:". These fileds shall be ordered in the message header, so that the most recent expansion comes first (same order as trace).

If any MTS (or MTA) Extensions not specified in X.400 are present, and they are marked as critical for transfer or delivery, then the message shall be rejected. If they are not so marked, they can safely be discarded. The list of discarded fields shall be indicated in the extended header "Discarded-X400-MTS-Extensions:".

5.3.7. Mappings from the MTA Abstract Service

There are some mappings at the MTA Abstract Service level which are done for IPM and IPN. These can be derived from MTA.MessageTransferEnvelope. The reasons for the mappings at this level, and the violation of layering are:

- Allowing for multiple recipients to share a single RFC 822 message

- Making the X.400 trace information available on the RFC 822 side

- Making any information on deferred delivery available

The SMTP recipients are calculated from the full list of X.400 recipients. This is all of the members of MTA.MessageTransferEnvelope.per-recipient-fields being passed through the gateway, where the responsibility bit is set. In some cases, a different RFC 822 message would be calculated for each recipient, due to differing service requests for each recipient. As discussed in 4.6.2.2, this specification allows either for multiple messages to be generated, or for the per-recipient information to be discarded.

The following EBNF is defined for extended RFC 822 headers:

```
mta-field       = "X400-Received" ":" x400-trace
                / "Deferred-Delivery" ":" date-time
                / "Latest-Delivery-Time" ":" date-time

x400-trace       = "by" md-and-mta ";"
                   [ "deferred until" date-time ";" ]
                   [ "converted" "(" encoded-info ")" ";" ]
                   [ "attempted" md-or-mta ";"  ]
                      action-list
                   ";" arrival-time

md-and-mta       = [ "mta" mta "in" ]  global-id
mta              = word
arrival-time     = date-time

md-or-mta        = "MD" global-id
                 / "MTA" mta

Action-list      = 1#action
action           = "Redirected"
                 / "Expanded"
                 / "Relayed"
                 / "Rerouted"
```

Note the EBNF.mta is encoded as 822.word. If the character set does not allow encoding as 822.atom, the 822.quoted-string encoding is used.

If MTA.PerMessageTransferFields.deferred-delivery-time is present, it is used to generate a Deferred-Delivery: field. X.400 does not make this information available at the MTS level on delivery, because it requires that this service is provided by the first MTA. In the event that the first MTA does not provide this service, the function may optionally be implemented by the gateway: that is, the gateway may hold the message until the time specified in the protocol element. Thus, the value of this element will usually be in the past. For this reason, the extended RFC 822 field is primarily for information.

If MTA.PerMessageTransferFields.extensions.dl-expansion-prohibited is present and set to dl-expansion-probited, the gateway may reject that message on the basis that it is unable to control distribution list expansion beyond the gateway. The service relating to this is described in Section 2.3.1.2. This approach was not specified in RFC 1327. If it is found to be useful, it may be made mandatory in future versions of MIXER.

If MTA.PerMessageTransferFields.extensions.recipient-reassignment-prohibited is present and set to recipeint-reassignment-probited, the gateway may reject that message on the basis that it is unable to control distribution list expansion beyond the gateway. The service relating to this is described in Section 2.3.1.2. This approach was not specified in RFC 1327. If it is found to be useful, it may be made mandatory in future versions of MIXER.

Merge MTA.PerMessageTransferFields.trace-information, and MTA.PerMessageTransferFields.internal-trace-information to produce a single ordered trace list. If Internal trace from other management domains has not been stripped, this may require complex interleaving. Where an element of internal trace and external trace are identical, except for the MTA in the internal trace, only the internal trace element shall be presented. Use this to generate a sequence of "X400-Received:" fields. The only difference between external trace and internal trace will be the extra MTA information in internal trace elements.

When generating an RFC 822 message all trace fields (X400-Received and Received) shall be at the beginning of the header, before any other fields. Trace shall be in chronological order, with the most recent element at the front of the message. This ordering is determined from the order of the fields, not from timestamps in the trace, as there is no guarantee of clock synchronisation. A simple example trace (external) is:

```
X400-Received: by /PRMD=UK.AC/ADMD=Gold 400/C=GB/ ; Relayed ;
        Tue, 20 Jun 89 19:25:11 +0100
```

A more complex example (internal):

```
X400-Received: by mta "UK.AC.UCL.CS" in
       /PRMD=UK.AC/ADMD=Gold 400/C=GB/ ;
       deferred until  Tue, 20 Jun 89 14:24:22 +0100 ;
        converted (undefined, g3fax) ; attempted MD /ADMD=Foo/C=GB/ ;
        Relayed, Expanded, Redirected ; Tue, 20 Jun 89 19:25:11 +0100
```

The gateway itself shall add a single line of trace information, indicating MIXER conversion by use of a comment. For example:

```
Received: from isode.com by isode.com
       (MIXER Conversion following RFC 1327);
       Thu, 2 Jan 1997 14:46:03 +0000
```

If SMTP is being used, Appendix A shall also be followed, which includes optional mappings to extension parameters.

5.3.8. Mappings from Report Delivery

that only reports destined for the MTS user will be mapped. Some additional services are also taken from the MTA service. X.400 Delivery Reports are Mapped onto Delivery Status Notifications, as defined by NOTARY [28].

5.3.8.1. MTS Mappings

A Delivery Report service will be represented as MTS.ReportDeliveryEnvelope, which comprises of per-report-fields (MTS.PerReportDeliveryFields) and per-recipient-fields.

The enclosing message is a MIME message of content type multipart/report, with report-type=delivery-status, which is generated with the following fields:

From:
An administrator at the gateway system.

To: A mapping of the MTA.ReportTransferEnvelope.report-destination-name. This is also the SMTP recipient.

Message-Type:
Set to "Delivery Report". This is strictly redundant, but retained for backwards compatibility with RFC 1327.

Subject:
The EBNF for the subject line is:

```
subject-line  = "Delivery-Report" "(" status ")"
                [ "for" destination ]

status        = "success" / "failure" / "success and failures"

destination   = mailbox / "MTA" word
```

The subject is intended to give a clear indication as to the nature of the message, and summarise its contents. EBNF.status is set according to whether the recipients reported on are all successes, all failures, or a mixture. It is common for a report to reference a single recipient, in which case a subject line giving using all of the options of EBNF.status can be used. This gives useful information to the recipient. Where information varies between reported recpients, the options cannot be used. The EBNF.destination is used to indicate the addresses in the reports. If the report is for a single address, EBNF.mailbox is used to give the RFC 822

representation of the address. If all of the reported recpients reference the same MTA this is included in EBNF.word. The MTA is determined from the delivery report's trace.

The format of the body of the message follows the NOTARY delivery status notification format, and is defined to ensure that all information is conveyed to the RFC 822 user in a consistent manner. The format is structured as if it was a message coming from the gateway, with three body parts. The first body part is ASCII text structured as follows:

1. A few lines giving keywords to indicate the original message.

2. A human summary of the status of each recipient being reported on.

The second (mandatory) body part is the NOTARY delivery status notification, which contains detailed information extracted from the report. This information may be critical to diagnosing an obscure problem.

The third (optional) body part contains the returned message (return of content). This structure is useful to the RFC 822 recipient, as it enables the original message to be extracted. For negative reports it shall be included if the original message is available. For positive reports headers from the message shall be included if the original message is available.

The first body part containing the user oriented description is of type text/plain. The format of this body part is defined below as EBNF.dr-user-info.

```
dr-user-info = dr-summary <CRLF>
               dr-recipients <CRLF>
               dr-content-return

dr-content-return = "The Original Message is not available"
     / "The Original Message follows:"

dr-summary = "This report relates to your message:" <CRLF>
               content-correlator <CRLF> <CRLF>
             "of" date-time <CRLF> <CRLF>

dr-recipients = *(dr-recipient <CRLF> <CRLF>)

dr-recipient = dr-recip-success / dr-recip-failure
```

```
dr-recip-success =
                "Your message was successfully delivered to:"
                 mailbox "at" date-time


dr-recip-failure = "Your message was not delivered to:"
                          mailbox <CRLF>
                "for the following reason:" *word report-point
= [ "mta" mta-name "in" ] global-id content-correlator = *word
mta-name = word
```

EBNF.dr-summary
The EBNF.content-correlator is taken from the content correlator (or content identifier if there is no content correlator) and the EBNF.date-time from the trace, as described in Section 5.3.8.3. LWSP may be added to improve the layout of the body part.

EBNF.dr-recipients
There is an element for each recipient in the delivery report. In each case, EBNF.mailbox is taken from the RFC 822 form of the originally specified recipient, which is taken from the originally specified recipient element if present or from the actual recipient. When reporting success, the message delivery time is used to derive EBNF.date-time. When reporting failure, the information includes a human readable interpretation of the X.400 diagnostic and reason codes, and the supplementary information.

EBNF.dr-content-return
This is set according to whether or not the content is being returned.

The EBNF of this body part is designed for english-speaking users. The language of the strings in the EBNF may be altered.

The EBNF used in the delivery status notification is:

```
dr-per-message-fields =
   / "X400-Conversion-Date" ":" date-time
   / "X400-Subject-Submision-Identifier" ":"
                              mts-msg-id
   / "X400-Content-Identifier" ":" printablestring
   / "X400-Content-Type" ":" mts-content-type
   / "X400-Original-Encoded-Information-Types" ":"
                         encoded-info
   / "X400-Originator-and-DL-Expansion-History" ":"
                         mailbox ";" date-time ";"
   / "X400-Reporting-DL-Name" ":" mailbox
   / "X400-Content-Correlator" ":" content-correlator
   / "X400-Recipient-Info" ":" recipient-info
   / "X400-Subject-Intermediate-Trace-Information" ":"
                         x400-trace
   / dr-extensions

dr-per-recipient-fields =
   / "X400-Redirect-Recipient" ":" "x400" ";" std-or
   / "X400-Mapped-Redirect-Recipient" ":" "rfc822" ";" mailbox
   / "X400-Converted-EITs" ":" encoded-info ";"
   / "X400-Delivery-Time" ":" date-time
   / "X400-Type-of-MTS-User" ":" labelled-integer
   / "X400-Last-Trace" ":" [ encoded-info ] date-time
   / "X400-Supplementary-Info" ":"
        <"> printablestring <"> ";"
   / "X400-Redirection-History" ":" redirect-history-item
   / "X400-Physical-Forwarding-Address" ":" mailbox
   / "X400-Originally-Specified-Recipient-Number" ":"
        integer
   / dr-extensions

dr-extensions = "X400-Discarded-DR-Extensions" ":"
                   1# (object-identifier / labelled-integer)

dr-diagnostic = "Reason" labelled-integer-2
                   [ ";" "Diagnostic" labelled-integer-2 ]
```

A body part of type delivery status, as defined by NOTARY, is generated. MIXER extends this delivery status notification (DSN) specification, by defining additional per message fields in EBNF.dr-per-message-fields and additional per recipient fields in EBNF.dr-per-recipient-fields. These are used as extensions to DSN.per-message-fields and DSN.per-recipient-fields. MIXER also defines a new NOTARY address type "x400", with encoding of EBNF.std-or. A directory name may be inluded as an RFC 822 comment.

The following DSN.per-message-fields are always generated:

DSN.reporting-mta-field
The DSN.mta-name-type is set to "x400", and this string is reserved by MIXER. The DSN.mta-name has its syntax specified by EBNF.report-point, with the information derived from the first element of the DR's trace.

DSN.arrival-date-field
This is derived from the date of the MTA.PerRecipientReportTransferFields.last-trace-info.arrival-time of the first recipient in the report.

The following two EBNF.per-message-fields are generated by the MIXER gateway:

DSN.dsn-gateway-field
The type is set to "dns" and the domain set to the local domain of the gateway.

X400-Conversion-Date:
The EBNF.date-time is set to the time of the MIXER conversion.

The elements of MTS.ReportDeliveryEnvelope.per-report-fields are mapped as follows onto the DSN per message fields as follows:

subject-submission-identifier
Mapped to DSN.original-envelope-id-field. The encoding of this MTS Identifier follows the format EBNF.mts-msg-id.

content-identifier
Mapped to X400-Content-Identifier:

content-type
Mapped to X400-Content-Type:

original-encoded-information-types
Mapped to X400-Encoded-Info:

The extensions from MTS.ReportDeliveryEnvelope.per-report-fields.extensions are mapped as follows:

originator-and-DL-expansion-history
Each element is mapped to an "X400-Originator-and-DL-Expansion-History:" They shall be ordered so that the most recent expansion comes first in the header (same order as trace).

reporting-DL-name
Mapped to X400-Reporting-DL-Name:

content-correlator
If the content correlator starts with the string "SMTP/NOTARY ENVID: ", then the remainder of the content correlator is mapped to the DSN original-envelope-id field. If this is not the case, the content correlator is mapped to X400-Content-Correlator:, provided that the encoding is IA5String (this will always be the case).

message-security-label
reporting-MTA-certificate
report-origin-authentication-check

These security parameters will not be present unless there is an error in a remote MTA. If they are present, they shall be discarded in preference to discarding the whole report. They shall be listed in the X400-Discarded-DR-Extensions: field.

If there are any other DR extensions, they shall also be discarded and listed in the X400-Discarded-DR-Extensions: field.

For each element of MTS.ReportDeliveryEnvelope.per-recipient-fields, a set of DSN.per-recipient-fields is generated. The fields are filled in as follows:

actual-recipient-name
If originally-intended-recipient-name is not present, generate a DSN.original-recipient-field fields, with DSN.address-type of "rfc822", and with an RFC 822 mailbox generated from the address encoded as specified by NOTARY. Also generate a DSN.final-recipient-field field, which holds the X.400 representation of the same address. If the directory name is present, it shall be added as a trailing comment in the X.400 form.

If originally-intended-recipient-name is present, generate an "X400-Mapped-Redirect-Recipient:" field, with DSN.address-type of "rfc822", and with an RFC 822 mailbox generated from the address encoded as specified by NOTARY. Also generate an "X400-Redirect-Recipient:" field, which holds the X.400 representation of the same address. If the directory name is present, it shall be added as a trailing comment in the X.400 form.

report
If it is MTS.Report.delivery, then set DSN.action-field to "delivered", and set "X400-Delivery-Time:" and "X400-Type-of-MTS-User:" from the information in the report. DSN.status field is set to "2.0.0".

If it is MTS.Report.non-delivery, then set DSN.action-field to "failed". DSN.diagnostic-code-field is encoded according to the syntax EBNF.dr-diagnostic, with the labelled integers set from the reason and diagnostic codes. DSN.status-field is derived from the reason and diagnostic codes, as described below.

converted-encoded-information-types
Set X400-Converted-EITs:

originally-intended-recipient
Generate a DSN.final-recipient-field field, with DSN.address-type of "rfc822", and with an RFC 822 mailbox generated from the address encoded as specified by NOTARY. Also generate a DSN.original-recipient-field field, which holds the X.400 representation of the same address. If the directory name is present, it shall be added as a trailing comment in the X.400 form.

supplementary-info
Set X400-Supplementary-Info:

redirection-history
Generate an "X400-Redirection-History:" field for each redirect history element. The fields are ordered with the earliest redirect first.

physical-forwarding-address
Set X400-Physical-Forwarding-Address as a mailbox, with directory name in comment if present.

recipient-certificate
Discard

proof-of-delivery
Discard

Any unknown extensions shall be discarded, irrespective of criticality. All discarded extensions shall be included in a "X400-Discarded-DR-Extensions:" field.

The number from the MTA.PerRecipientReportTransferFields.originally-specified-recipient-number shall be mapped to "X400-Originally-Specified-Recipient-Number:", in order to facilitate reverse mapping of delivery reports.

The original message shall be included in the delivery status notification if it is available. The original message will usually be available at the gateway, as discussed in Section 5.2. If the original message is available, but is not a legal message format, a dump of the ASN.1 may be included, encoded as application/octet-string. This is recommended, but not required.

Where the original message is included, it shall be encoded according to the MIME specifications as content type message/rfc822.

5.3.8.2. Status Code Mappings

This section defines the mappings from X.400 diagnostic and status codes to the NOTARY Status field.

C/D	X400 meaning	DSN code	Means
0/Any	Transfer failure (may be temporary)	4.4.0	Other net/route
1/Any	Unable to transfer	5.0.0	Other, unknown
2/Any	Conversion not performed	5.6.3	Conv not supported
3/Any	Physical rendition not performed	5.6.0	Other media error
4/Any	Physical delivery not performed	5.1.0	Other address status
5/Any	Restricted delivery	5.7.1	
6/Any	Directory operation unsuccessful	5.4.3	Routing server failure
7/Any	Deferred delivery not performed	5.3.3	Not capable
1/0	Unrecognized OR name	5.1.1	
1/1	Ambiguous OR name	5.1.4	
1/2	MTS congestion	4.3.1	
1/3	Loop detected	5.4.6	
1/4	Recipient unavailable	4.2.1	
1/5	Delivery time expired	4.4.7	
1/6	Encoded information types unsupported	5.6.1	Media unsupp.
1/7	Content too long	5.2.3	
2/8	Conversion impractical	5.6.3	
2/9	Conversion prohibited	5.6.3	
1/10	Implicit conversion not subscribed	5.6.3	
1/11	Invalid arguments	5.5.2	
1/12	Content syntax error	5.5.2	
1/13	Size constraint violation	5.5.2	
1/14	Protocol violation	5.5.0	

1/15	Content type not supported	5.6.1	Media unsupp.
1/16	Too many recipients	5.5.3	
1/17	No bilateral agreement	5.4.4	
1/18	Unsupported critical function	5.3.3	System not capable
2/19	Conversion with loss prohibited	5.6.2	
2/20	Line too long	5.6.0	
2/21	Page split	5.6.0	
2/22	Pictorial symbol loss	5.6.2	
2/23	Punctuation symbol loss	5.6.2	
2/24	Alphabetic character loss	5.6.2	
2/25	Multiple information loss	5.6.2	
1/26	Recipient reassignment prohibited	5.4.0	Undefined net/route
1/27	Redirection loop detected	5.4.6	
1/28	DL expansion prohibited	5.7.2	
1/29	No DL submit permission	5.7.1	Delivery not authorized
1/30	DL expansion failure	4.2.4	
4/31	Physical rendition attrs not supported	5.6.0	Undefined media error
4/32-45	Various physical mail stuff	5.1.0	Other address status
1/43	New address unknown	5.1.6	Destination mbox moved
1/46	Secure messaging error	5.7.0	Other security status
2/47	Unable to downgrade	5.3.3	System not capable
0/48	Unable to complete transfer	5.3.4	Message too big
0/49	Transfer attempts limit reached	4.4.7	Delivery time expired

5.3.8.3. MTA Mappings

The single SMTP recipient is constructed from MTA.ReportTransferEnvelope.report-destination-name, using the mappings of Chapter 4. Unlike with a user message, this information is not available at the MTS level.

The following additional mappings are made, which results in fields in the outer header of the DSN.

MTA.ReportTransferEnvelope.report-destination-name
: This is used to generate the To: field.

MTA.ReportTransferEnvelope.identifier
: Mapped to the extended RFC 822 field "X400-MTS-Identifier:". It may also be used to derive a "Message-Id:" field.

MTA.ReportTransferEnvelope.trace-information
and
MTA.ReportTransferEnvelope.internal-trace-information

Mapped onto the extended RFC 822 field "X400-Received:", as described in Section 5.3.7. Date: is generated from the first element of trace.

The following additional mappings are made, which result in per message fields in the DSN body part:

MTA.PerRecipientReportTransferFields.last-trace-information
Mapped to X400-Last-Trace:".

MTA.PerReportTransferFields.subject-intermediate-trace-information Mapped to "X400-Subject-Intermediate-Trace-Information:". These fields are ordered so that the most recent trace element comes first.

5.3.8.4. Example Delivery Reports

This section contains sample delivery reports. These are the same examples used in RFC 1327, and so they also illustrate the changes between RFC 1327 and this document. Example Delivery Report 1:

Received: from cs.ucl.ac.uk by bells.cs.ucl.ac.uk
via Delivery Reports Channel id <27699-0@bells.cs.ucl.ac.uk>;
Thu, 7 Feb 1991 15:48:39 +0000 From: UCL-CS MTA
<postmaster@cs.ucl.ac.uk> To: S.Kille@cs.ucl.ac.uk Subject: Delivery
Report (failure) for H.Hildegard@bbn.com Message-Type: Delivery
Report Date: Thu, 7 Feb 1991 15:48:39 +0000 Message-ID:
<"bells.cs.u.694:07.01.91.15.48.34"@cs.ucl.ac.uk> X400-Content-
Identifier: Greetings. MIME-Version: 1.0 Content-Type:
multipart/report; report-type=delivery-status;
boundary=boundary-1

--boundary-1

This report relates to your message:
Greetings.

of Thu, 7 Feb 1991 15:48:20 +0000

Your message was not delivered to
H.Hildegard@bbn.com for the following reason:
Bad Address
MTA 'bbn.com' gives error message (USER) Unknown user name
in

```
        "H.Hildegard@bbn.com"

   The Original Message follows:


   --boundary-1 content-type: message/delivery-status

   Reporting-MTA: x400;  bells.cs.ucl.ac.uk in /PRMD=uk.ac/ADMD=gold
   400/C=gb/ Arrival-Date: Thu, 7 Feb 1991 15:48:34 +0000 DSN-Gateway:
   dns;  bells.cs.ucl.ac.uk X400-Conversion-Date: Thu, 7 Feb 1991
   15:48:40 +0000 Original-Envelope-Id:
            [/PRMD=uk.ac/ADMD=gold
   400/C=gb/;<1803.665941698@UK.AC.UCL.CS>] X400-Content-Identifier:
   Greetings.  X400-Subject-Intermediate-Trace-Information:
   /PRMD=uk.ac/ADMD=gold 400/C=gb/;
            arrival Thu, 7 Feb 1991 15:48:20 +0000 action Relayed X400-
   Subject-Intermediate-Trace-Information:  /PRMD=uk.ac/ADMD=gold
   400/C=gb/;
            arrival Thu, 7 Feb 1991 15:48:18 +0000 action Relayed



   Original-Recipient: rfc822; H.Hildegard@bbn.com Final-Recipient:
   x400;
     /RFC-822=H.Hildegard(a)bbn.com/OU=cs/O=ucl/PRMD=uk.ac/ADMD=gold
   400/C=gb/; Action: failure Status: 5.1.1 Diagnostic-Code: x400;
   Reason 1 (Unable-To-Transfer);
        Diagnostic 0 (Unrecognised-ORName) X400-Last-Trace: (ia5) Thu, 7
   Feb 1991 15:48:18 +0000; X400-Originally-Specified-Recipient-Number:
   1 X400-Supplementary-Info: "MTA 'bbn.com' gives error message  (USER)
       Unknown user name in "H.Hildegard@bbn.com"";


   --boundary-1 Content-Type: message/rfc822

   Received: from glenlivet.cs.ucl.ac.uk by bells.cs.ucl.ac.uk
     with SMTP inbound id <27689-0@bells.cs.ucl.ac.uk>;
     Thu, 7 Feb 1991 15:48:21 +0000 To: H.Hildegard@bbn.com Subject:
   Greetings.  Phone: +44-71-380-7294 Date: Thu, 07 Feb 91 15:48:18
   +0000 Message-ID: <1803.665941698@UK.AC.UCL.CS> From: Steve Kille
   <S.Kille@cs.ucl.ac.uk>


   Steve

   --boundary-1--
```

Example Delivery Report 2:

```
Received: from cs.ucl.ac.uk by bells.cs.ucl.ac.uk
  via Delivery Reports Channel id <27718-0@bells.cs.ucl.ac.uk>;
  Thu, 7 Feb 1991 15:49:11 +0000
X400-Received: by mta "bells.cs.ucl.ac.uk" in
  /PRMD=uk.ac/ADMD=gold 400/C=gb/;
  Relayed; Thu, 7 Feb 1991 15:49:08 +0000
X400-Received: by /PRMD=DGC/ADMD=GOLD 400/C=GB/; Relayed;
  Thu, 7 Feb 1991 15:48:40 +0000
From: UCL-CS MTA <postmaster@cs.ucl.ac.uk>
To: S.Kille@cs.ucl.ac.uk
Subject: Delivery Report (failure) for
         j.nosuchuser@dle.cambridge.DGC.gold-400.gb
Message-Type: Delivery Report
Date: Thu, 7 Feb 1991 15:46:11 +0000
Message-ID: <"DLE/910207154840Z/000"@cs.ucl.ac.uk>
X400-Content-Identifier: A useful mess...
MIME-Version: 1.0
Content-Type: multipart/report; report-type=delivery-status;
    boundary=boundary-1

--boundary-1

This report relates to your message:
        A useful mess...

        of Thu, 7 Feb 1991 15:43:20 +0000


Your message was not delivered to
        j.nosuchuser@dle.cambridge.DGC.gold-400.gb
        for the following reason:
        Bad Address
        DG 21187: (CEO POA) Unknown addressee.

The Original Message is not available


--boundary-1
content-type: message/delivery-status


Reporting-MTA: x400; /PRMD=DGC/ADMD=GOLD 400/C=GB/
Arrival-Date: Thu, 7 Feb 1991 15:48:40 +0000
DSN-Gateway: dns;  bells.cs.ucl.ac.uk
X400-Conversion-Date: Thu, 7 Feb 1991 15:49:12 +0000
Original-Envelope-Id:
```

```
    [/PRMD=uk.ac/ADMD=gold 400/C=gb/;<1796.665941626@UK.AC.UCL.CS>]
  X400-Content-Identifier: A useful mess...


  Original-Recipient: rfc822; j.nosuchuser@dle.cambridge.DGC.gold-400.gb
  Final-Recipient: x400;
    /I=j/S=nosuchuser/OU=dle/O=cambridge/PRMD=DGC/ADMD=GOLD 400/C=GB/
  Action: failure
  Status: 5.1.1
  Diagnostic-Code: x400; Reason 1 (Unable-To-Transfer);
      Diagnostic 0 (Unrecognised-ORName)
  X400-Supplementary-Info: "DG 21187: (CEO POA) Unknown addressee."
  X400-Originally-Specified-Recipient-Number: 1

  --boundary-1--
```

5.3.9. Probe

This is an MTS internal issue. Any probe shall be serviced by the gateway, as there is no equivalent RFC 822 functionality. The value of the reply is dependent on whether the gateway could service an MTS Message with the values specified in the probe. The reply shall make use of MTS.SupplementaryInformation to indicate that the probe was serviced by the gateway.

Appendix A - Mappings Specific to SMTP

This Appendix is specific to the Simple Mail Transfer Protocol (RFC 821). It describes specific changes in the context of this protocol. When MIXER is used with SMTP, conformance to this appendix is mandatory.

1. Probes

When servicing a probe, as described in section 5.3.9, use may be made of the SMTP VRFY command to increase the accuracy of information contained in the delivery report.

2. Long Lines

SMTP is a text oriented protocol, and is required to support a line length of at least 1000 characters. Some implementations do not support line lengths greater than 1000 characters. This can cause problems. Where body parts have long lines, it is recommended to use a MIME encoding that folds lines (quoted printable).

3. SMTP Extensions

There are several RFCs that specify extensions to SMTP. Most of these are not relevant to MIXER. The NOTARY work to support delivery report defines extensions which are relevant [29]. Use of these extensions by a MIXER gateway is optional. If these extensions are used, they shall be used in the manner described below.

3.1. SMTP Extension mapping to X.400

Mappings are defined for the following extensions:

NOTIFY
: This is used to set the report and non-delivery bits of MTA.PerRecipientMessageTransferFields.per-recipient-indicators. If the value is NEVER, both bits are zero. If SUCCESS is present, the report bit is set. Otherwise, the non-delivery-report bit is set. If the gateway uses the NOTIFY command, it shall perform this mapping in all cases.

ORCPT
: If the address type of the original recipient is "x400" or "rfc822", this may be used at the MTS level, to generate an element of redirection history, with the redirection date being the date of conversion and the reason set to "alias".

ENVID
If present, this may be used to generate a content correlator. This is used rather than the MTS Identifier, as the ENVID is unique for the UA only and is likely to be too large to map to an MTS identifier. The content correlator is encoded as an IA5 String containing the ENVID and prefixed by the string:

"SMTP/NOTARY ENVID: "

If the ENVID starts with the string "X400-MTS-Identifier: ", then this ENVID was generated from an X.400 MTS Identifier. The reverse mapping defined in Section 3.2 of Appendix A shall not be used, as this may cause problems in certain situations (e.g., where the message was expanded by an Internet mailing list).

3.2. X.400 Mapping to SMTP Extensions

The following extensions may be used as a part of the MIXER mapping:

NOTIFY
The originator-report and originator-non-delivery-report bits of MTA.PerRecipientMessageTransferFields.per-recipient-indicators determine how this is used. If both bits are zero, the parameter is NEVER. If the report bit is set, SUCCESS is used. Otherwise, FAILURE is used. If this is done, the gateway shall not generate a delivery report for this recipient, unless this is needed in the case where the originating MTA service report requirements differ from the user requirements. Additional originating MTA requrirements are satisfied by the gateway.

ORCPT
If the MTS.perRecipientDeliveryFields.originally-intended-recipient-name is present, the ORCPT command may be used to carry this value, using the "x400" syntax.

ENVID
This may be generated, with the value taken from the MTS.MessageDeliveryEnvelope.message-delivery-identifer. If this is done, it shall be encoded as EBNF.mts-msg-id, preceded by the string "X400-MTS-Identifier: ".

RET
If MTA.PerMessageTransferFields.per-message-indicators.content-return-request is set to FALSE, the parameter RET may be set to HDRS, to specify return of headers only.

Appendix B - Mapping with X.400(1984)

This appendix defines modifications to the mapping for use with X.400(1984).

The X.400(1984) protocols are a proper subset of X.400(1988). When mapping from X.400(1984) to RFC 822, no changes to this specification are needed.

When mapping from RFC 822 to X.400(1984), no use can be made of 1988 specific features. No use of such features is made at the MTS level. The heading extension feature is used at the IPMS level, and this shall be replaced by the RFC 987 approach. All header information which would usually be mapped into the rfc-822-heading-list extension is mapped into a single IA5 body part, which is the first body part in the message. This body part will start with the string "RFC-822-Headers:" as the first line. The headers then follow this line. This specification requires correct reverse mapping of this format, either from 1988 or 1984. RFC 822 extended headers which could be mapped into X.400(1988) elements, are also mapped to the body part.

In an environment where RFC 822 is of major importance, it may be desirable for downgrading to consider the case where the message was originated in an RFC 822 system, and mapped according to this specification. The rfc-822-heading-list extension may be mapped according to this appendix.

When parsing std-or, the following restrictions shall be observed:

- Only the 84/88 attributes identified in the table in Section 4.2 are present.

- No teletex encoding is allowed.

If an address violates this, it shall be treated as an RFC 822 address, which will usually lead to encoding as a DDA "RFC-822".

It is possible that attributes of zero length may be present in an OR Address. This is not legal in 1988, except for ADMD where the case is explicitly described in Section 4.3.5. Attributes of zero length are deprecated (the attribute shall be omitted), and will therefore be unusual. However, some systems generate them and rely on them. Therefore, any null attribute shall be enoded using the std-or encoding (e.g., /O=/).

If a non-Teletex Common Name (CN) is present, it shall be mapped onto a Domain Defined Attribute "Common". This is in line with RFC 1328 on X.400 1988 to 1984 downgrading [22].

This specification defines a mapping of the Internet message framework to X.400. Body part mappings are defined in RFC 2157 [6], which relies on X.400(88) features. Downgrading to X.400(84) for body parts is defined in RFC 1496 (HARPOON), which shall be followed in the context of this appendix [5].

Appendix C - RFC 822 Extensions for X.400 access

This appendix defines a number of optional mappings which may be provided to give access from RFC 822 to a number of X.400 services. These mappings are beyond the basic scope of this specification. There has been a definite demand to use extended RFC 822 as a mechanism to access X.400, and these extensions provide access to certain features. If this functionality is provided, this appendix shall be followed. The following headings are defined:

```
extended-heading =
    "Prevent-NonDelivery-Report" ":"
    / "Generate-Delivery-Report" ":"
    / "Alternate-Recipient" ":" prohibition
    / "Disclose-Recipients" ":"  prohibition
    / "X400-Content-Return" ":" prohibition
```

Prevent-NonDelivery-Report and Generate-Delivery-Report allow setting of MTS.PerRecipientSubmissionFields.originator-report-request. The setting will be the same for all recipients.

Alternate-Recipient, Disclose-Recipients, and X400-Content-Return allow for override of the default settings for MTS.PerMessageIndicators.

Use of NOTARY mechanisms is a preferred meachanism for controlling these parameters.

Appendix D - Object Identifier Assignment

The following Object Identifiers shall be used.

```
internet ::= OBJECT IDENTIFIER  { iso org(3) dod(6) 1 } -- from RFC
1155

mail OBJECT IDENTIFIER ::= { internet 7 }  -- IANA assigned

mixer OBJECT IDENTIFIER ::= { mail mixer(1) } -- inherited from RFC
1495
mixer-core OBJECT IDENTIFIER ::= { mixer core(3) }

id-rfc-822-field-list OBJECT IDENTIFIER ::= {mixer-core 2}
id-dsn-header-list OBJECT IDENTIFIER ::= {mixer-core 3}
id-dsn-field-list OBJECT IDENTIFIER ::= {mixer-core 4}

eit-mixer OBJECT IDENTIFIER ::= {mixer-core 5}
               -- the MIXER pseudo-EIT
```

This object identifier for id-rfc-822-field-list is different to the one assigned in RFC 1327, which was erroneous.

Appendix E - BNF Summary

```
   boolean = "TRUE" / "FALSE"


   numericstring = *(DIGIT / " ")


   printablestring  = *( ps-char )
   ps-restricted-char      = 1DIGIT /  1ALPHA / " " / "'" / "+"
                           / "," / "-" / "." / "/" / ":" / "=" / "?"
   ps-delim         = "(" / ")"
   ps-char          = ps-delim / ps-restricted-char


   ps-encoded       = *( ps-restricted-char / ps-encoded-char )
   ps-encoded-char  = "(a)"               ; (@)
                    / "(p)"                 ; (%)
                    / "(b)"                 ; (!)
                    / "(q)"                 ; (")
                    / "(u)"                 ; (_)
                    / "(l)"                 ; "("
                    / "(r)"                 ; ")"
                    / "(" 3DIGIT ")"

   teletex-string   = *( ps-char / t61-encoded )
   t61-encoded      = "{" 1* t61-encoded-char "}"
   t61-encoded-char = 3DIGIT


   teletex-and-or-ps = [ printablestring ] [ "*" teletex-string ]


   labelled-integer ::= [ key-string ] "(" numericstring ")"

   labelled-integer-2 ::= [ numericstring ] "(" key-string ")"

   key-string      = *key-char
   key-char        = <a-z, A-Z, 0-9, and "-">

   object-identifier  ::= oid-comp object-identifier
                        | oid-comp

   oid-comp ::= [ key-string ] "(" numericstring ")"


   encoded-info    = 1#encoded-type
```

```
   encoded-type    = built-in-eit / object-identifier

   built-in-eit    = "Undefined"          ; undefined (0)
                   / "Telex"              ; tLX (1)
                   / "IA5-Text"           ; iA5Text (2)
                   / "G3-Fax"             ; g3Fax (3)
                   / "TIF0"               ; tIF0 (4)
                   / "Teletex"            ; tTX (5)
                   / "Videotex"           ; videotex (6)
                   / "Voice"              ; voice (7)
                   / "SFD"                ; sFD (8)
                   / "TIF1"               ; tIF1 (9)


   encoded-pn      = [ given "." ] *( initial "." ) surname

   given           = 2*<ps-char not including ".">

   initial         = ALPHA

   surname         = printablestring


   std-or-address  = 1*( "/" attribute "=" value ) "/"
   attribute       = standard-type
                   / "RFC-822"
                   / dd-key "." std-printablestring

   std-or-address-input =  [ sep pair ] sep  pair *( sep pair )
                           sep  [ pair sep ]

   sep             = "/" / ";"
   pair            = input-attribute "=" value
   input-attribute = attribute
                   / dd-key ":" std-printablestring


   standard-type   = key-string

   dd-key          = key-string

   value           = std-printablestring

   std-printablestring
                   = *( std-char / std-pair )

   std-char        = <"{", "}", "*", and any ps-char
```

```
                                                    except "/" and "=" >
   std-pair        = "$" ps-char


   global-id = std-or-address



   mta-field       = "X400-Received" ":" x400-trace
                   / "Deferred-Delivery" ":" date-time
                   / "Latest-Delivery-Time" ":" date-time



   x400-trace       = "by" md-and-mta ";"
                    [ "deferred until" date-time ";" ]
                    [ "converted" "(" encoded-info ")" ";" ]
                    [ "attempted" md-or-mta ";"  ]
                       action-list
                       ";" arrival-time


   md-and-mta       = [ "mta" mta "in" ]  global-id
   mta              = word
   arrival-time     = date-time

   md-or-mta        = "MD" global-id
                    / "MTA" mta

   Action-list      = 1#action
   action           = "Redirected"
                    / "Expanded"
                    / "Relayed"
                    / "Rerouted"

   dr-user-info = dr-summary <CRLF>
                  dr-recipients <CRLF>
                  dr-content-return


   dr-content-return = "The Original Message is not available"
        / "The Original Message follows:"

   dr-summary = "This report relates to your message:" <CRLF>
                  content-correlator <CRLF> <CRLF>
                "of" date-time <CRLF> <CRLF>

   dr-recipients = *(dr-recipient <CRLF> <CRLF>)
```

```
   dr-recipient = dr-recip-success / dr-recip-failure

   dr-recip-success =
                  "Your message was successfully delivered to:"
                   mailbox "at" date-time


   dr-recip-failure = "Your message was not delivered to:"
                           mailbox <CRLF>
                  "for the following reason:" *word

   report-point = [ "mta" mta-name "in" ] global-id
   content-correlator = *word
   mta-name = word

      dr-per-message-fields =
                / "X400-Conversion-Date" ":" date-time
                / "X400-Subject-Submision-Identifier" ":"
                                  mts-msg-id
                / "X400-Content-Identifier" ":" printablestring
                / "X400-Content-Type" ":" mts-content-type
                / "X400-Original-Encoded-Information-Types" ":"
                             encoded-info
                / "X400-Originator-and-DL-Expansion-History" ":"
                             mailbox ";" date-time ";"
                / "X400-Reporting-DL-Name" ":" mailbox
                / "X400-Content-Correlator" ":" content-correlator
                / "X400-Recipient-Info" ":" recipient-info
                / "X400-Subject-Intermediate-Trace-Information" ":"
                                    x400-trace
                / dr-extensions


      dr-per-recipient-fields =
                / "X400-Redirect-Recipient" ":" "x400" ";" std-or
                / "X400-Mapped-Redirect-Recipient" ":" "rfc822" ";"
                     mailbox
                / "X400-Converted-EITs" ":" encoded-info ";"
                / "X400-Delivery-Time" ":" date-time
                / "X400-Type-of-MTS-User" ":" labelled-integer
                / "X400-Last-Trace" ":" [ encoded-info ] date-time
                / "X400-Supplementary-Info" ":"
                     <"> printablestring <"> ";"
                / "X400-Redirection-History" ":" redirect-history-item
                / "X400-Physical-Forwarding-Address" ":" mailbox
                / "X400-Originally-Specified-Recipient-Number" ":"
                     integer
                / dr-extensions
```

```
      dr-extensions = "X400-Discarded-DR-Extensions" ":"
                       1# (object-identifier / labelled-integer)


      dr-diagnostic = "Reason" labelled-integer-2
                      [ ";" "Diagnostic" labelled-integer-2 ]

      mts-field = "X400-MTS-Identifier" ":" mts-msg-id
                / "X400-Originator" ":" mailbox
                / "X400-Recipients" ":" 1#mailbox
                / "Original-Encoded-Information-Types" ":"
                                encoded-info
                / "X400-Content-Type" ":" mts-content-type
                / "X400-Content-Identifier" ":" printablestring
                / "Priority" ":" priority
                / "Originator-Return-Address" ":" 1#mailbox
                / "DL-Expansion-History" ":" mailbox ";" date-time ";"
                / "Conversion" ":" prohibition
                / "Conversion-With-Loss" ":" prohibition
                / "Delivery-Date" ":" date-time
                / "Discarded-X400-MTS-Extensions" ":"
                             1#( object-identifier / labelled-integer )

      prohibition     = "Prohibited" / "Allowed"

      mts-msg-id       = "[" global-id ";" *text "]"

      mts-content-type = "P2" /  labelled-integer
                      / object-identifier

      priority        = "normal" / "non-urgent" / "urgent"

      ipn-body-format = ipn-description <CRLF>
                      [ ipn-extra-information <CRLF> ]
                      [ ipn-content-return ]

      ipn-description = ipn-receipt / ipn-non-receipt

      ipn-receipt = "Your message to:" preferred-recipient <CRLF>
                "was received at" receipt-time <CRLF> <CRLF>
                "This notification was generated"
                 acknowledgement-mode <CRLF>
                "The following extra information was given:" <CRLF>
                 ipn-suppl <CRLF>

      ipn-non-receipt = "Your message to:"
                      preferred-recipient <CRLF>
                      ipn-reason
```

```
ipn-reason = ipn-discarded / ipn-auto-forwarded

ipn-discarded = "was discarded for the following reason:"
                discard-reason <CRLF>

ipn-auto-forwarded = "was automatically forwarded." <CRLF>
                [ "The following comment was made:"
                        auto-comment ]


ipn-extra-information =
         "The following information types were converted:"
         encoded-info

ipn-content-return = "The Original Message is not available"
                / "The Original Message follows:"

preferred-recipient = mailbox
receipt-time        = date-time
auto-comment        = printablestring
ipn-suppl           = printablestring


discard-reason     = "Expired" / "Obsoleted" /
            "User Subscription Terminated" / "IPM Deleted"

acknowledgement-mode = "Manually" / "Automatically"


ipms-field = "Supersedes" ":" 1*msg-id
           / "Expires" ":" date-time
           / "Reply-By" ":" date-time
           / "Importance" ":" importance
           / "Sensitivity" ":" sensitivity
           / "Autoforwarded" ":" boolean
           / "Incomplete-Copy" ":"
           / "Content-Language" ":" 1#language
           / "Message-Type" ":" message-type
           / "Discarded-X400-IPMS-Extensions" ":"
                1#object-identifier
           / "Autosubmitted" ":" autosubmitted

importance      = "low" / "normal" / "high"

sensitivity     = "Personal" / "Private" /
                       "Company-Confidential"

language        = 2*ALPHA [ "(" language-description ")" ]
```

```
      language-description = printable-string


      message-type    = "Delivery Report"
                      / "InterPersonal Notification"
                      / "Multiple Part"


      autosubmitted   = "not-auto-submitted"
                      / "auto-generated"
                      / "auto-replied"
                      / "auto-forwarded"

      redirect-comment  = redirect-first *( redirect-subsequent )

      redirect-first = "Originally To:"  mailbox  "Redirected on"
               date-time "To:"  redirection-reason

      redirect-subsequent = mailbox  "Redirected Again on"
               date-time "To:"  redirection-reason

      redirection-history-item = "intended recipient" mailbox
               "redirected to"  redirection-reason
               "on" date-time

      redirection-reason =
               "Recipient Assigned Alternate Recipient"
               / "Originator Requested Alternate Recipient"
               / "Recipient MD Assigned Alternate Recipient"
               / "Directory Look Up"
               / "Alias"

      subject-line  = "Delivery-Report" "(" status ")"
                      [ "for" destination ]

      status        = "success" / "failure" / "success and failures"

      destination   = mailbox / "MTA" word

      extended-heading =
          "Prevent-NonDelivery-Report" ":"
          / "Generate-Delivery-Report" ":"
          / "Alternate-Recipient" ":" prohibition
          / "Disclose-Recipients" ":"  prohibition
          / "X400-Content-Return" ":" prohibition
```

Appendix F - Text format for MCGAM distribution

1. Text Formats

This appendix defines text formats for exchange of four types of mapping.

1. Domain Name Space -> OR Address Space MCGAM

2. OR Address Space -> Domain Name Space MCGAM

3. Domain Name Space -> OR Address of preferred gateway

4. OR Address Space -> Domain Name of preferred gateway

2. Mechanisms to register and to distribute MCGAMs

There is a well known set of MCGAM tables.

The global coordination of the mapping rules is a part of the DANTE MailFLOW Project. New mapping rules may be defined by the authority responsible for the relevant name space. The rules need to be registered with a national mapping registration authority, which in turn passes them on to the central mapping registration authority. All the collected mapping rules are merged together into the globally coordinated mapping tables by the MailFLOW Project Team. The tables are available from the national mapping registration authorities.

To get a contact address of the mapping registration authority for the respective country or more information about the MailFLOW Project contact:

SWITCH
MailFLOW Project Team
Limmatquai 138
8001 Zuerich
Switzerland

email: mailflow@mailflow.dante.net
S=MailFLOW;O=MailFLOW;P=DANTE;A=mailnet;C=fi;

fax: +41 1 268 15 68
tel: +41 1 268 15 20

3. Syntax Definitions

An address syntax is defined, which is compatible with the syntax used for 822.domains. By representing the OR addresses as domains, all lookups can be mechanically implemented as domain -> domain mappings. This syntax defined is initially for use in table format, but the syntax is defined in a manner which makes it suitable to be adapted for use with the Domain Name Service. This syntax allows for a general representation of OR addresses, so that it can be used in other applications. Not all attributes are used in the table formats defined.

To allow the mapping where a level of the hierarchy is omitted, the pseudo-value "@" (not a printable string character) is used to indicate omission of a level in the hierarchy. This is distinct from the form including the element with no value, although a correct X.400 implementation will interpret both in the same manner.

This syntax is not intended to be handled by users.

```
dmn-or-address  = dmn-part *( "." dmn-part )
mn-part        = dmn-attribute "$" value
dmn-attribute  = standard-type
                 /  "~" dmn-printablestring
value           = dmn-printablestring
                 / "@"
dmn-printablestring =
                 = *( dmn-char / dmn-pair )
dmn-char        = <"{", "}", "*", and any ps-char
                                        except ".">
dmn-pair        = "\."
```

An example usage:

```
~ROLE$Big\.Chief.ADMD$ATT.C$US
PRMD$DEC.ADMD$@.C$US
```

The first example illustrates quoting of a "." and a domain define attribute (ROLE). The second example illustrates omission of the ADMD level. There shall be a strict ordering of all components in this table, with the most significant components on the RHS. This allows the encoding to be treated as a domain.

Various further restrictions are placed on the usage of dmn-or-address in the address space mapping tables.

a. Only C, ADMD, PRMD, O, and up to four OUs may be used.

b. No components shall be omitted from this hierarchy, although the hierarchy may terminate at any level. If the mapping is to an omitted component, the "@" syntax is used.

4. Table Lookups

When determining a match, there are aspects which apply to all lookups. Matches are always case independent. The key for all three tables is a domain. The longest possible match shall be obtained. Suppose the table has two entries with the following keys:

```
K.L
J.K.L
```

Domain "A.B.C" will not return any matches. Domain "I.J.K.L" will match the entry "J.K.L:.

5. Domain -> OR Address MCGAM format

The BNF is:

```
domain-syntax "#" dmn-or-address "#"
```

EBNF.domain-syntax is defined in Section 4.2. Note that the trailing "#" is used for clarity, as the dmn-or-address syntax might lead to values with trailing blanks. Lines starting with "#" are comments.

```
For example:
AC.UK#PRMD$UK\.AC.ADMD$GOLD 400.C$GB#
XEROX.COM#O$Xerox.ADMD$ATT.C$US#
GMD.DE#O$@.PRMD$GMD.ADMD$DBP.C$DE#
```

A domain is looked up to determine the top levels of an OR Address. Components of the domain which are not matched are used to build the remainder of the OR address, as described in Section 4.3.4.

6. OR Address -> Domain MCGAM format

The syntax of this table is:

```
dmn-or-address "#" domain-syntax "#"
```

For example:

```
#
# Mapping table
#
PRMD$UK\.AC.ADMD$GOLD 400.C$GB#AC.UK#
```

The OR Address is used to generate a domain key. It is important to order the components correctly, and to fill in missing components in the hierarchy. Use of this mapping is described in Section 4.3.2.

7. Domain -> OR Address of Preferred Gateway table

This uses the same format as the domain -> OR address MCGAM table. In this case, the restriction to only use C/ADMD/PRMD/O/OU does not apply. Use of this mapping is described in Section 4.3.4. A domain cannot appear in this table and in the domain to OR Address table.

8. OR Addresss -> domain of Preferred Gateway table

This uses the same format as the OR Address -> domain MCGAM table. Use of this mapping is described in Section 4.3.5. An OR Address cannot appear in this table and in the OR Address to domain table.

Appendix G - Conformance

This appendix defines a number of options, which a conforming gateway shall specify. Conformance to this specification shall not be claimed if any of the mandatory features are not implemented. A specification of conformance may list the service elements of Chapter 2, in order to be clear that full conformance is provied. In particular:

- Formats for all fields shall be followed.
- The gateway shall enable MCGAMs to be used.
- Formats for subject lines, delivery reports and IPNs shall be followed. A system which followed the syntax, but translated text into a language other than english would be conformant.
- RFC 1137 shall not be followed when mapping to SMTP.
- All mappings of trace shall be implemented.
- There shall be a mechanism to access all three global mappings.
- RFC 2157 shall be followed for mapping body parts.
- When it is specified that a MIME format message is generated, RFC 2045 shall be followed.

A gateway shall specify:

- Which Interent Message Transport (822-MTS) protocols are supported. If SMTP is supported, Appendex A of MIXER shall be used.
- Which X.400 versions are supported (84, 88, 92).
- Which mechanisms (table, X.500, DNS) are supported to access MCGAMs.
- The mechanism or mechanisms by which the global mapping information is accessed.

The following are optional parts of this specification. A conforming implementation shall specify which of these it supports.

- Support for the extension mappings of Appendix C.

- Support for returning illegal format content in a delivery report

- Which address interpretation heuristics are supported (4.3.4.1)

- If RFC 987 generated message ids are handled in a backwards compatible manner (4.7.3.6)

Appendix H - Change History: RFC 987, 1026, 1138, 1148

RFC 987 was the original document, and contained the key elements of this specification. It was specific to X.400(1984). RFC 1026 specified a small number of necessary changes to RFC 987.

RFC 1138 was based on the RFC 987 work. It contained an editorial error, and was reissued a few months later as RFC 1148. RFC 1148 will be referred to here, as it is the document which is widely referred to elsewhere. The major goal of RFC 1148 was to upgrade RFC 987 to X.400(1988). It did this, but did not obsolete RFC 987, which was recommended for use with X.400(1984). This appendix summarises the changes made in going from RFC 987 to RFC 1148.

RFC 1148 noted the following about its upgrade from RFC 987: Unnecessary change is usually a bad idea. Changes on the RFC 822 side are avoided as far as possible, so that RFC 822 users do not see arbitrary differences between systems conforming to this specification, and those following RFC 987. Changes on the X.400 side are minimised, but are more acceptable, due to the mapping onto a new set of services and protocols.

1. Introduction

The model has shifted from a protocol based mapping to a service based mapping. This has increased the generality of the specification, and improved the model. This change affects the entire document.

A restriction on scope has been added.

2. Service Elements

- The new service elements of X.400 are dealt with.
- A clear distinction is made between origination and reception

3. Basic Mappings

- Add teletex support
- Add object identifier support
- Add labelled integer support
- Make PrintableString <-> ASCII mapping reversible

- The printable string mapping is aligned to the NBS mapping derived from RFC 987.

4. Addressing

- Support for new addressing attributes

- The message ID mapping is changed to not be table driven

5. Detailed Mappings

- Define extended IPM Header, and use instead of second body part for RFC 822 extensions

- Realignment of element names

- New syntax for reports, simplifying the header and introducing a mandatory body format (the RFC 987 header format was unusable)

- Drop complex autoforwarded mapping

- Add full mapping for IP Notifications, defining a body format

- Adopt an MTS Identifier syntax in line with the OR Address syntax

- A new format for X400 Trace representation on the RFC 822 side

6. Appendices

- Move Appendix on restricted 822 mappings to a separate RFC

- Delete Phonenet and SMTP Appendixes

Appendix I - Change History: RFC 1148 to RFC 1327

1. General

- The scope of the document was changed to cover X.400(1984), and so obsolete RFC 987.

- Changes were made to allow usage to connect RFC 822 networks using X.400

- Text was tightened to be clear about optional and mandatory aspects

- A good deal of clarification

- A number of minor EBNF errors

- Better examples are given

- Further X.400 upper bounds are handled correctly

2. Basic Mappings

- The encoding of object identifier is changed slightly

3. Addressing

- A global mapping of domain to preferred gateway is introduced.

- An overflow mechanism is defined for RFC 822 addresses of greater than 128 bytes

- Changes were made to improve compatibility with the PDAM on writing OR Addresses.

+ The PD and Terminal Type keywords were aligned to the PDAM. It is believed that minimal use has been made of the RFC 1148 keywords.

+ P and A are allowed as alternate keys for PRMD and ADMD

+ Where keywords are different, the PDAM keywords are alternatives on input. This is mandatory.

4. Detailed Mappings

- The format of the Subject: lines is defined.

- Illegal use (repetition) of the heading EXTENSION is corrected, and a new object identifier assigned.

- The Delivery Report format is extensively revised in light of operational experience.

- The handling of redirects is significantly changed, as the previous mechanism did not work.

5. Appendices

- An SMTP appendix is added, allowing optional use of the VRFY command to improve probe information.

- Handling of JNT Mail Acknowledge-To is changed slightly.

- A DDA JNT-MAIL is allowed on input.

- The format definitions of Appendix F are explained further, and a third table definition added.

- An appendix on use with X.400(1984) is added.

- Optional extensions are defined to give RFC 822 access to further X.400 facilities.

- An appendix on conformance is added.

Appendix J - Change History: RFC 1327 to this Document

1. General

This update is primarily for stability, and to fold in compatibility for MIME and to add support for the new NOTARY delivery status notifications. Other general changes:

- Various editorial updates
- Minor EBNF errors
- Reference to mapping table support by DNS and X.500.
- Alignment to X.400(92)
- Assignment of a new object identifier
- Removal of specification relating to body mapping, which is now defined in RFC 2157.

2. Service Elements

- Support of Auto-Submitted service

3. Basic Mappings

- Comments shall not be used in new headers, to remove parsing ambiguity
- RFC 1522 encoding may be used as an alternative to X.408 downgrade, where appropriate.
- Correct handling of RFC 822 four year dates.

4. Addressing

- Replaced the mandatory global address mapping with MCGAMs.
- Add codes and add a heuristic to align to the standard X.400 form of writing OR Addresses.
- Improved text on ordering heuristic
- Leading "/" interpretation added

- All bar one of the address mapping heuristics made mandatory.

- Interpretation of domain defined attribute "RFC-822" made mandatory in all cases

- Make report request comments optional

5. Detailed Mappings

- Comments no longer maps to separate body part

- Allow Languages to be multi-valued

- Change Content-Identifier to X400-Content-Identifier, in order to avoid confusion with MIME.

- Reverse mapping of MIXER defined fields made mandatory

- "Expiry-Date:" changed to "Expires:".

- "Obsoletes:" changed to "Supersedes:".

- Define correct handling when "Resent-Date:" is present.

6. Appendices

- Change "Content-Return" to "X400-Content-Return" in Appendix C.

- Relaxation of restrictions on mapping 3 in Appendix F.

- Add linkage to HARPOON in Appendix B.

- RFC 2157 added to the conformance statement of Appendix G.

- Added Appendix L, with ASN. Summary.

Appendix L - ASN.1 Summary

```
   MIXER Definitions { iso org(3) dod(6) internet(1) mail(7)
         mixer(1) mixer-core(3) definitions(1) }

   DEFINITIONS IMPLICIT TAGS ::=

   BEGIN

   -- exports everything

   IMPORTS

   EXTENSION FROM
     MTSAbstractService {join-iso-ccit mhs-motis(6) mts(3)
           modules(0) mts-abstract-service(1) }

     HEADING-EXTENSION FROM
       IPMSAbstractService {join-iso-ccit mhs-motis(6) ipms(1)
             modules(0) abstract-service(3) }



   rfc-822-field HEADING-EXTENSION
           VALUE RFC822FieldList
           ::= id-rfc-822-field-list


   RFC822FieldList ::= SEQUENCE OF RFC822Field

   RFC822Field ::= IA5String



   dsn-header-list EXTENSION
      RFC822FieldList
      ::= id-dsn-header-list

   dsn-field-list EXTENSION
      RFC822FieldList
      ::= id-dsn-field-list

   internet ::= OBJECT IDENTIFIER  { iso org(3) dod(6) 1 } -- from RFC
   1155

   mail OBJECT IDENTIFIER ::= { internet 7 }  -- IANA assigned
```

```
   mixer OBJECT IDENTIFIER ::= { mail mixer(1) } -- inherited from RFC
   1495
   mixer-core OBJECT IDENTIFIER ::= { mixer core(3) }

   id-rfc-822-field-list OBJECT IDENTIFIER ::= {mixer-core 2}
   id-dsn-header-list OBJECT IDENTIFIER ::= {mixer-core 3}
   id-dsn-field-list OBJECT IDENTIFIER ::= {mixer-core 4}

   eit-mixer OBJECT IDENTIFIER ::= {mixer-core 5}
                  -- the MIXER pseudo-EIT


   END -- MIXER ASN.1
```

SECURITY CONSIDERATIONS

Security issues are not discussed in this memo.

AUTHOR'S ADDRESS

Steve Kille
Isode Ltd
The Dome
The Square
Richmond
TW9 1DT
England

Phone: +44-181-332-9091
Internet EMail: S.Kille@ISODE.COM

X.400 Email: I=S; S=Kille; P=Isode; A=Mailnet; C=FI;

UFN: S.Kille, Isode, GB

References

1. CCITT , "Recommendations X.400", Message Handling Systems: System Model - Service Elements, October 1984.

2. Allocchio, C., "MaXIM11 - Mapping between X.400 / Internet Mail and Mail-11 mail", RFC 2162, January 1998.

3. Allocchio, C., "Using the Internet DNS to Distribute MIXER Conformant Global Address Mapping (MCGAM)", RFC 2163, January 1998.

4. Alvestrand, H., Kille, S., Miles, R., Rose, M., and S. Thompson, "Mapping between X.400 and RFC-822 Message Bodies", RFC 1495, August 1993.

5. Alvestrand, H., Romaguera, J., and K. Jordan, "Rules for Downgrading Messages for X.400(88) to X.400(84) When MIME Content-Types are Present in the Messages (Harpoon)", RFC 1496, August 1993.

6. Alvestrand, H., and S. Thompson, "Equivalences between X.400 and RFC-822 Message Bodies", RFC 1494, August 1993.

7. Alvestrand, H., "Tags for the Identification of Languages", RFC 1766, March 1995.

8. Alvestrand, H., "Mapping between X.400 and RFC-822/MIME Message Bodies", RFC 2157, January 1998.

9. Freed, N., and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

10. Braden, R., "Requirements for Internet Hosts -- Application and Support", STD 3, RFC 1123, October 1989.

11. CCITT/ISO, "CCITT Recommendations X.420/ ISO/IEC 10021-7," Message Handling Systems: Interpersonal Messaging System, Dec 1988.

12. CCITT/ISO, "CCITT Recommendations X.411/ ISO/IEC 10021-4," Message Handling Systems: Message Transfer System: Abstract Service Definition and Procedures, Dec 1988.

13. CCITT/ISO, "CCITT Recommendations X.400/ ISO/IEC 10021-1," Message Handling: System and Service Overview , Dec 1988.

14. CCITT/ISO, "Specification of Abstract Syntax Notation One (ASN.1)," CCITT Recommendation X.208 / ISO/IEC 8824, Dec 1988.

15. CCITT/ISO, "CCITT Recommendations X.400/ ISO/IEC 10021-1," Message Handling: System and Service Overview , Dec 1992.

16. Crocker, D., "Standard of the Format of ARPA Internet Text Messages", STD 11, RFC 822, August 1982.

17. Kille, S., "Mapping Between X.400 and RFC 822", UK Academic Community Report (MG.19) / RFC 987, June 1986.

18. Kille, S., "Addendum to RFC 987", UK Academic Community Report (MG.23) / RFC 1026, August 1987.

19. Kille, S., "Mapping Between X.400(1988) / ISO 10021 and RFC 822", RFC 1138, October 1989.

20. Kille, S., "Mapping Between X.400(1988) / ISO 10021 and RFC 822", RFC 1148, March 1990.

21. Kille, S., "Mapping Between X.400(1988) / ISO 10021 and RFC 822", RFC 1327, May 1992.

22. Kille, S., "X.400 1988 to 1984 downgrading", RFC 1328, May 1992.

23. Kille, S., "A String Encoding of Presentation Address", RFC 1278, November 1992.

24. Kille, S., "A String Representation of Distinguished Name", RFC 1485, January 1992.

25. Kille, S., "Using the OSI Directory to achieve User Friendly Naming", RFC 1484, January 1992.

26. Kille, S., "Use of an X.500/LDAP directory to support MIXER address mapping", RFC 2164, January 1998.

27. Koorland, N., "Message Attachmment Work Group (MAWG): MAWG Feasibility Project Guide," EMA Report, Version 1.5, Nov 1995.

28. Moore, K., and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", RFC 1894, January 1996.

29. Moore, K., "SMTP Service Extensions for Delivery Status Notifications", RFC 1891, Januaty 1996.

30. Postel, J., "SIMPLE MAIL TRANSFER PROTOCOL", STD 10, RFC 821, August 1982.

Full Copyright Statement