INTERNATIONAL TELECOMMUNICATION UNION

CCITT

X.420

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

(09/92)

DATA COMMUNICATION NETWORKS

MESSAGE HANDLING SYSTEMS – INTERPERSONAL MESSAGING SYSTEM

Recommendation X.420

FOREWORD

The CCITT (the International Telegraph and Telephone Consultative Committee) is a permanent organ of the International Telecommunication Union (ITU). CCITT is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The Plenary Assembly of CCITT which meets every four years, establishes the topics for study and approves Recommendations prepared by its Study Groups. The approval of Recommendations by the members of CCITT between Plenary Assemblies is covered by the procedure laid down in CCITT Resolution No. 2 (Melbourne, 1988).

Recommendation X.420 was revised by Study Group VII and was approved under the Resolution No. 2 procedure on the 10th of September 1992.

CCITT NOTES

1) In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized private operating agency.

2) A list of abbreviations used in this Recommendation can be found in Annex Q.

© ITU 1993

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

INTRODUCTION

This Recommendation is one of a set of Recommendations for Message Handling. The entire set provides a comprehensive blueprint for a Message Handling System (MHS) realized by any number of cooperating open systems.

The purpose of an MHS is to enable users to exchange messages on a store-and-forward basis. A message submitted on behalf of one user, the originator, is conveyed by the Message Transfer System (MTS) and subsequently delivered to the agents of one or more additional users, the recipients. Access units (AUs) link the MTS to communication systems of other kinds (e.g. postal systems). A user is assisted in the preparation, storage, and display of messages by a user agent (UA). Optionally, it is assisted in the storage of messages by a message store (MS). The MTS comprises a number of message transfer agents (MTAs) which collectively perform the store-and-forward message transfer function.

This Recommendation defines the Message Handling application called *Interpersonal Messaging*, specifying in the process the message content type and associated procedures known as *P2*.

The text of this Recommendation is the subject of joint CCITT-ISO agreement. The corresponding ISO/IEC International Standard is ISO/IEC 10021-7:1990 as modified by Technical Corrigenda 1, 2, 3 and 4 and draft Amendment 1.

Recommendation X.420

MESSAGE HANDLING SYSTEMS – INTERPERSONAL MESSAGING SYSTEM

(revised 1992)

SECTION 1 – INTRODUCTION

1 Scope

This Recommendation defines **Interpersonal Messaging**, a form of Message Handling tailored for ordinary interpersonal business or private correspondence.

This Recommendation is one of a series on Message Handling. CCITT Rec. X.402 | ISO/IEC 10021-2 constitutes the introduction to the series and identifies the other documents in it.

The architectural basis and foundation for Message Handling are defined in still other Recommendations | International Standards. CCITT Rec. X.402 | ISO/IEC 10021-2 identifies those documents as well.

This Recommendation is structured as follows. Section 1 is this introduction. Section 2 defines the kinds of information objects exchanged in Interpersonal Messaging. Section 3 defines the associated abstract service. Section 4 specifies how it is provided. Annexes provide important supplemental information.

The requirements for conformance to this Recommendation are given in clause 22.

2 Normative references

The following CCITT Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation are encouraged to investigate the possibility of applying the most recent editions of the Recommendations and Standards listed below. Members of ISO and IEC maintain registers of currently valid International Standards. The CCITT Secretariat maintains a list of currently valid CCITT Recommendations.

2.1 *Open Systems Interconnection*

This Recommendation cites the following OSI specification:

– CCITT Recommendation X.208 (1988), *Specification of Abstract Syntax Notation One (ASN.1)*.

  ISO/IEC 8824:1990, *Information processing systems – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1)*.

– CCITT Recommendation X.209 (1988), *Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)*.

  ISO/IEC 8825:1990, *Information processing systems – Open Systems Interconnection – Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)*.

2.2 *Message Handling Systems*

This Recommendation cites the following Message Handling System specifications:

– CCITT Recommendation X.400 (1992), *Message handling: Service and system overview*.

ISO/IEC 10021-1:1990, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 1: Service and system overview*.

ISO/IEC 10021-1: 1990/Cor. 1: 1991, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 1: Service and system overview – Technical Corrigendum 1*.

ISO/IEC 10021-1:1990/Cor. 2: 1991, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 1: Service and system overview – Technical Corrigendum 2*.

ISO/IEC 10021-1:1990/Cor. 3: 1992, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 1: Service and system overview – Technical Corrigendum 3*.

ISO/IEC 10021-1:1990/Cor. 4: 1992, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 1: Service and system overview – Technical Corrigendum 4*.

– CCITT Recommendation X.402 (1992), *Message handling: Overall architecture*

ISO/IEC 10021-2:1990, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) - Part 2: Overall architecture*.

ISO/IEC 10021-2:1990/Cor. 1: 1991, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 2: Overall architecture – Technical Corrigendum 1*.

ISO/IEC 10021-2:1990/Cor. 2: 1991, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 2: Overall architecture – Technical Corrigendum 2*.

ISO/IEC 10021-2:1990/Cor. 3: 1992, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 2: Overall architecture – Technical Corrigendum 3*.

ISO/IEC 10021-2:1990/Cor. 4: 1992, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 2: Overall architecture – Technical Corrigendum 4*.

ISO/IEC 10021-2:1990/Amd. 1: 1993, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 2: Overall architecture – Amendment 1: Representation of O/R Addresses for Human Usage*.

ISO/IEC 10021-2:1990/Amd. 2: 1993, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 2: Overall architecture – Amendment 2: Minor Enhancements*.

ISO/IEC 10021-3:1990, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 3: Abstract service definition conventions*.

ISO/IEC 10021-3:1990/Cor. 1: 1992, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 3: Abstract service definition conventions – Technical Corrigendum 1*.

– CCITT Recommendation X.408 (1988), *Message handling systems: Encoded information type conversion rules*.

– CCITT Recommendation X.411 (1992), *Message handling: Message transfer system: Abstract service definition and procedures*.

ISO/IEC 10021-4:1990, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 4: Message transfer system: Abstract service definition and procedures*.

ISO/IEC 10021-4:1990/Cor. 1: 1991, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 4: Message transfer system: Abstract service definition and procedures – Technical Corrigendum 1*.

ISO/IEC 10021-4:1990/Cor. 2: 1991, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 4: Message transfer system: Abstract service definition and procedures – Technical Corrigendum 2*.

ISO/IEC 10021-4:1990/Cor. 3: 1992, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 4: Message transfer system: Abstract service definition and procedures – Technical Corrigendum 3*.

ISO/IEC 10021-4:1990/Cor. 4: 1992, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 4: Message transfer system: Abstract service definition and procedures – Technical Corrigendum 4*.

ISO/IEC 10021-4:1990/Amd. 1: 1993, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 4: Message transfer system: Abstract service definition and procedures – Amendment 1: Minor Enhancements*.

– CCITT Recommendation X.413 (1992), *Message handling: Message store: Abstract service definition*.

ISO/IEC 10021-5:1990, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 5: Message store: Abstract service definition*.

ISO/IEC 10021-5:1990/Cor. 1: 1991, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 5: Message store: Abstract service definition – Technical Corrigendum 1*.

ISO/IEC 10021-5: 990/Cor. 2: 1991, *Information technology - Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 5: Message store: Abstract service definition – Technical Corrigendum 2*.

ISO/IEC 10021-5:1990/Cor. 3: 1992, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 5: Message store: Abstract service definition – Technical Corrigendum 3*.

ISO/IEC 10021-5:1990/Cor. 4: 1992, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 5: Message store: Abstract service definition – Technical Corrigendum 4*.

– CCITT Recommendation X.419 (1992), *Message handling: Protocol specifications*.

ISO/IEC 10021-6:1990, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 6: Protocol specifications*.

ISO/IEC 10021-6:1990/Cor. 1: 1991, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 6: Protocol specifications – Technical Corrigendum 1*.

ISO/IEC 10021-6:1990/Cor. 2: 1991, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 6: Protocol specifications – Technical Corrigendum 2*.

ISO/IEC 10021-6:1990/Cor. 3: 1992, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 6: Protocol specifications - Technical Corrigendum 3*.

ISO/IEC 10021-6:1990/Cor. 4: 1992, *Information technology – Text communication – Message-Oriented Text Interchange Systems (MOTIS) – Part 6: Protocol specifications – Technical Corrigendum 4*.

– CCITT Recommendation X.408 (1988), *Message handling systems: Encoded information type conversion rules*.

– CCITT Recommendation X.420 (1984), Message handling systems: Interpersonal messaging user agent layer.

2.3 *Directory Systems*

This Recommendation cites the following Directory System specification:

– CCITT Recommendation X.501 (1988), *The Directory – Models.*

ISO/IEC 9594-2:1990, *Information technology – Open Systems Interconnection – The Directory – Part 2: Models*.

2.4 *Language Code*

This Recommendation cites the following Language Code specification:

– ISO 639:1988, *Code for the representation of names of languages*.

2.5 *Character Sets*

This Recommendation cites the following Character Set specifications:

– ISO 2375:1985, *Data processing – Procedure for registration of escape sequences*.

– ISO 8859-1:1987, *Information processing – 8-bit single-byte coded graphic character sets – Part 1: Latin Alphabet No. 1*.

– CCITT Recommendation T.61 (1988), *Character repertoire and coded character sets for the international teletex service*.

2.6 *Telematic Services*

This Recommendation cites the following Telematic Service specifications:

– CCITT Recommendation T.4 (1988), *Standardization of group 3 facsimile apparatus for document transmission*.

– CCITT Recommendation T.30 (1988), *Procedures for document facsimile transmission in the general switched telephone network*.

– CCITT Recommendation T.100 (1988), *International information exchange for interactive videotex*.

– CCITT Recommendation T.101 (1988), *International interworking for videotex services*.

– CCITT Recommendation T.330 (1988), *Telematic access to IPMS*.

2.7 *File Transfer*

This Recommendation cites the following File Transfer specifications:

– ISO 8571-1:1988, *Information processing systems – Open Systems Interconnection – File Transfer, Access and Management – Part 1: General Introduction. Technical Corrigendum 1:1990 – Amendment 1:1992 File Store Management, Amendment 2:1992, Overlapped access.*

– ISO 8571-2:1988, *Information processing systems – Open Systems Interconnection – File Transfer, Access and Management – Part 2: Virtual Filestore Definition.*

– ISO 8571-4:1988, *Information processing systems – Open Systems Interconnection – File Transfer, Access and Management – Part 4: File Protocol Specification – Technical Corrigendum 1:1990, Technical Corrigendum 2:1992.*

2.8 *Open Document Architecture*

This Recommendation cites the following Open Document Architecture specifications:

– CCITT Recommendation T.415 (1988), *Open document architecture (ODA) and interchange format – Open document interchange format (ODIF).*

– ISO 8613-5:1989, *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 5: Office Document Interchange Format (ODIF).*

3 Definitions

For the purposes of this Recommendation, the definitions given in CCITT Rec. X.402 | ISO/IEC 10021-2 apply.

4 Abbreviations

For the purposes of this Recommendation, the abbreviations given in CCITT Rec. X.402 | ISO/IEC 10021-2 apply.

5 Conventions

This Recommendation uses the descriptive conventions identified below.

5.1 *ASN.1*

This Recommendation uses for the indicated purposes the following ASN.1-based descriptive conventions:

a) To define the information objects of Interpersonal Messaging, and other data types and values of all kinds, ASN.1 itself.

b) To define the functional objects of Interpersonal Messaging, the OBJECT and REFINE macros of CCITT Rec. X.407 | ISO/IEC 10021-3.

c) To define the abstract service of Interpersonal Messaging, the PORT and ABSTRACT-OPERATION and -ERROR macros of CCITT Rec. X.407 | ISO/IEC 10021-3.

d) To define the *heading extensions*, the IPMS-EXTENSION macro of 7.2.17.

e) To define *extended body part types*, the EXTENDED-BODY-PART-TYPE macro of 7.3.12.

f) To define MS attributes, the ATTRIBUTE macro of CCITT Rec. X.500 | ISO/IEC 9594-2.

The various uses of the ASN.1 notation are summarized in Table 1/X.420. With the two exceptions evident from the table, whenever ASN.1 is employed, it appears both in the body of the Recommendation to aid the exposition, and again, largely redundantly, in an annex for reference.

TABLE 1/X.420

Uses of the ASN.1 notation

Subject matter	Exposition	Reference
Object identifiers	–	Annex D
Abstract information objects	Section 2	Annex E
Functional objects	Clauses 10, 11, 16	Annex F
Abstract service	Clauses 12 and 13	Annex G
Heading extensions	Annex A	Annex H
Extended body part types	Annex B	Annex I
Message store attributes	Annex C	Annex J
Upper bounds	–	Annex K

If differences are found between the ASN.1 used in the exposition and that supplied for reference, a specification error is indicated.

ASN.1 tags are implicit throughout the ASN.1 module the annex defines; the module is definitive in that respect.

Note 1 – The use of ASN.1 to describe a class or piece of information does not in itself imply that that information is transported between open systems. The fact that the information, by virtue of its description in ASN.1 and of ASN.1's Basic Encoding Rules, has a concrete transfer syntax may be immaterial. Information actually conveyed between systems is designated as such by its inclusion in an application protocol.

Note 2 – The use of the ABSTRACT-OPERATION and -ERROR macros, derived from the correspondingly named macros of Remote Operations, does not imply that the abstract operations and errors are invoked and reported across the boundary between open systems. The fact that the abstract operations and errors, by virtue of their description using these macros and with minimal additional specification, actually could be invoked via ROS is immaterial in the present context.

Note 3 – Underline used in ASN.1 indicates "new" for 1992.

5.2 *Grade*

This Recommendation uses the concept of grade as developed in CCITT Rec. X.402 | ISO/IEC 10021-2.

5.3 *Terms*

Throughout this Recommendation, terms are rendered in **bold** when defined, in *italic* when referenced prior to their definitions, without emphasis upon all other occasions.

Terms that are proper nouns are capitalized, generic terms are not.

SECTION 2 – ABSTRACT INFORMATION OBJECTS

6 Overview

This section abstractly describes the information objects that users exchange in Interpersonal Messaging. They are of two kinds, *interpersonal messages* (*IPMs*) and *interpersonal notifications* (*IPNs*). One of the latter acknowledges a user's receipt of one of the former.

```
InformationObject ::= CHOICE {
        ipm [0] IPM,
        ipn [1] IPN }
```

This section covers the following topics:

a) interpersonal messages;

b) interpersonal notifications.

Note 1 – The use, throughout this section, of words such as "originator" and "recipient" anticipates the fact that *IPMs* and *IPNs* are conveyed between users as the contents of messages (see clause 20). These words, therefore, refer to the roles users and DLs play in such transmittals.

Note 2 – An *IPM* may appear (see § 7.3.8) in the *Body* of another *IPM* which itself is conveyed as the content of a message. The words "originator" and "recipient" shall be understood in the context of an *IPM*'s conveyance as the (entire) content of a message, not as a component of the *Body* of another *IPM* so conveyed.

Note 3 – An *IPM* or *IPN* makes various assertions about its own transmittal (e.g. who originates the message containing it). Furthermore, an *IPN* makes assertions about the transmittal of the *IPM* to which it responds. All of these assertions are unverified.

7 Interpersonal messages

An **interpersonal message** (**IPM**) is a member of the primary class of information object conveyed between users in Interpersonal Messaging.

```
IPM ::= SEQUENCE {
        heading Heading,
        body    Body }
```

It has the following components:

a) **Heading**: A Set of **heading fields** (or **fields**), each an information item that gives a characteristic of the IPM (e.g. its importance). See 7.2 for the syntax definition.

b) **Body**: A Sequence of **body parts**, each an information object that the **IPM** is intended to convey between users (e.g. a document).

```
Body ::= SEQUENCE OF BodyPart
```

The structure of an IPM is depicted in Figure 1/X.420.

This clause defines and describes the most prominent Heading field component types and the defined Heading fields and body part types.

Note – An IPM may be likened to a business memo. In fact, the terms "Heading" and "Body" appeal to that analogy.

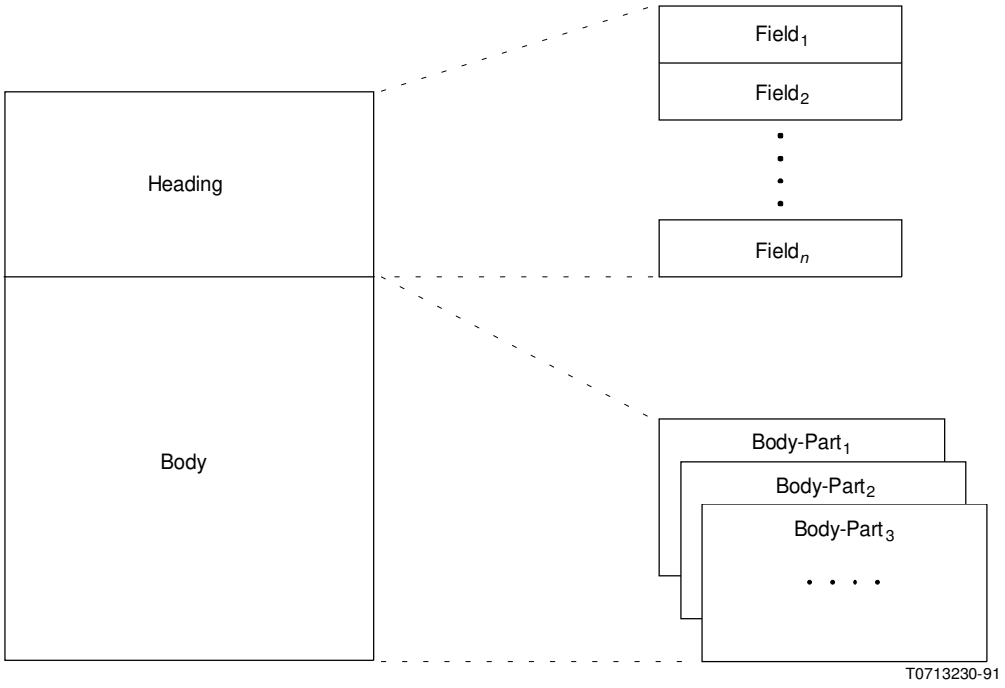

FIGURE 1/X.420

An interpersonal message

7.1 *Heading field component types*

Information items of several kinds appear throughout the Heading. These Heading field component types --*IPM identifier*, *recipient specifier*, and *O/R descriptor*--are defined and described below.

7.1.1 *IPM identifier*

An **IPM identifier** is an information item that unambiguously and uniquely identifies an IPM, distinguishing it from all other IPMs ever conveyed by any user.

```
IPMIdentifier ::= [APPLICATION 11] SET {
        user                    ORName OPTIONAL,
        user-relative-identifier  LocalIPMIdentifier }
```

An IPM identifier has the following components:

a) **User** (O): Identifies the user who originates the IPM. One of the user's O/R names. This component's omission is discouraged.

b) **User-relative-identifier** (M): Uniquely and unambiguously identifies the IPM, distinguishing it from all other IPMs that the user who is identified by the User component originates. A Printable String of from zero to a prescribed number of characters (see annex K). A length of zero is discouraged.

```
LocalIPMIdentifier ::= PrintableString
        (SIZE (0..ub-local-ipm-identifier) )
```

Note – The "11" in IPMIdentifier is the only ASN.1 application-wide tag this Recommendation assigns.

7.1.2 *Recipient specifier*

A **recipient specifier** is an information item that identifies a (preferred) recipient of an IPM and that may make certain requests of him.

```
RecipientSpecifier ::= SET {
    recipient              [0]  ORDescriptor,
    notification-requests  [1]  NotificationRequests DEFAULT { },
    reply-requested        [2]  BOOLEAN DEFAULT FALSE,
    recipient-extensions   [3]  RecipientExtensionsField OPTIONAL }
```

A recipient specifier has the following components:

a) **Recipient** (M): Identifies the preferred recipient in question. An *O/R descriptor*.

If the *Notification-requests* or *Reply-requested* component makes a request of the preferred recipient, the *Formal-name* component of the *O/R descriptor* above shall be present.

b) **Notification-requests** (D no values): May make certain requests of the preferred recipient denoted by the Recipient component.

```
NotificationRequests ::= BIT STRING {
    rn          (0),
    nrn         (1),
    ipm-return  (2) }
```

This component may assume any of the following values simultaneously, except that the value *rn* shall not be selected unless the value *nrn* is selected:

i) *rn:* A *receipt notification* is requested in the circumstances prescribed in clause 8.

ii) *nrn:* A *non-receipt notification* is requested in the circumstances prescribed in clause 8.

iii) *ipm-return:* It is requested that the IPM be returned in any *non-receipt notification*.

c) **Reply-requested** (D *false*): Indicates whether a reply is requested of the preferred recipient denoted by the Recipient component. A Boolean.

A **reply** is one IPM sent in response to another. A user may reply to an IPM even though no reply is requested of him and, indeed, even if he is not among the IPM's preferred recipients. Furthermore, a user of whom a reply is requested may refrain from replying.

d) **Recipient-extensions** (O) contains extensions to the recipient specifier subfield.

```
RecipientExtensionsField ::= SET OF IPMSExtension
```

There are no recipient extensions defined in this Recommendation.

7.1.3 *O/R descriptor*

An **O/R descriptor** is an information item that identifies a user or DL.

```
ORDescriptor ::= SET {
    formal-name              ORName OPTIONAL,
    free-form-name      [0]  FreeFormName OPTIONAL,
    telephone-number    [1]  TelephoneNumber OPTIONAL }
```

An O/R descriptor has the following components:

a) **Formal-name** (C): Identifies the user or DL in question. One of its O/R names.

This conditional component shall be present if (but not only if) one or more of the following criteria are satisfied:

i) The *Free-form-name* component is absent.

ii) The O/R descriptor appears in the *Reply Recipients* heading field.

iii) The O/R descriptor is the Recipient component of a recipient specifier and the conditions stated in item a) of 7.1.2 are satisfied.

b) **Free-form-name** (O): Identifies the user or DL in question. A Teletex String of from zero to a prescribed number of characters (see Annex K), chosen from the graphic subset of the Teletex String character set. A length of zero is discouraged.

```
FreeFormName ::= TeletexString (SIZE (0..ub-free-form-name))
```

c) **Telephone-number** (O): Provides the telephone number of the user or DL in question. A Printable String of from zero to a prescribed number of characters (see Annex K), chosen from the graphical subset of the Printable String character set. A length of zero is discouraged.

```
TelephoneNumber ::= PrintableString [SIZE (0..ub-telephone-number)]
```

Note – One or more O/R descriptors may appear in each of the following heading fields: Originator, Authorizing Users, Primary Recipients, Copy Recipients, Blind Copy Recipients, and Reply Recipients. In addition, an O/R descriptor may appear in the following notification fields (see clause 8): IPN Originator and IPM Preferred Recipient.

7.2 *Heading fields*

The fields that may appear in the Heading of an IPM are defined and described below.

```
Heading ::= SET {
    this-IPM                    ThisIPMField,
    originator                  [0] OriginatorField OPTIONAL,
    authorizing-users           [1] AuthorizingUsersField OPTIONAL,
    primary-recipients          [2] PrimaryRecipientsField DEFAULT { },
    copy-recipients             [3] CopyRecipientsField DEFAULT { },
    blind-copy-recipients       [4] BlindCopyRecipientsField OPTIONAL,
    replied-to-IPM              [5] RepliedToIPMField OPTIONAL,
    obsoleted-IPMs              [6] ObsoletedIPMsField DEFAULT { },
    related-IPMs                [7] RelatedIPMsField DEFAULT { },
    subject                     [8] EXPLICIT SubjectField OPTIONAL,
    expiry-time                 [9] ExpiryTimeField OPTIONAL,
    reply-time                 [10] ReplyTimeField OPTIONAL,
    reply-recipients           [11] ReplyRecipientsField OPTIONAL,
    importance                 [12] ImportanceField DEFAULT normal,
    sensitivity                [13] SensitivityField OPTIONAL,
    auto-forwarded             [14] AutoForwardedField DEFAULT FALSE,
    extensions                 [15] ExtensionsField DEFAULT { } }
```

Some fields have components and thus are composite, rather than indivisible. A field component is called a **sub-field**.

7.2.1 *This IPM*

The **This IPM** heading field (M) identifies the IPM. It comprises an IPM identifier.

```
ThisIPMField ::= IPMIdentifier
```

7.2.2 *Originator*

The **Originator** heading field (O) identifies the IPM's originator. It comprises an O/R descriptor.

```
OriginatorField ::= ORDescriptor
```

7.2.3 *Authorizing Users*

The **Authorizing Users** heading field (C) identifies the zero or more users who are the IPM's *authorizing users*. It comprises a Sequence of sub-fields, each an O/R descriptor, one for each such user.

AuthorizingUsersField ::= SEQUENCE OF AuthorizingUsersSubfield

AuthorizingUsersSubfield ::= ORDescriptor

An **authorizing user** is a user who, either individually or in concert with others, authorizes the origination of an IPM. The word "authorizes" above is not precisely defined by this Recommendation; it is given meaning by users.

This conditional field shall be present if, and only if, the authorizing users are other than the IPM's originator alone.

Note – Suppose, e.g. that a manager instructs his secretary to originate an IPM on his behalf. In this case, the secretary, the IPM's originator, might consider the manager the authorizing user.

7.2.4 *Primary Recipients*

The **Primary Recipients** heading field [D no subfields (i.e. elements)] identifies the zero or more users and DLs who are the "primary recipients" of the IPM. It also identifies the responses the authorizing users ask of each of those users and of each member of those DLs. It comprises a Sequence of sub-fields, each a recipient specifier, one for each primary recipient.

PrimaryRecipientsField ::= SEQUENCE OF PrimaryRecipientsSubfield

PrimaryRecipientsSubfield ::= RecipientSpecifier

The phrase "primary recipients" above is not precisely defined by this Recommendation; it is given meaning by users.

Note – The primary recipients, e.g. might be those users and those DLs whose members are expected to act upon the IPM.

7.2.5 *Copy Recipients*

The **Copy Recipients** heading field [D no subfields (i.e. elements)] identifies the zero or more users and DLs who are the "copy recipients" of the IPM. It also identifies the responses the authorizing users ask of each of those users and of each member of those DLs. It comprises a Sequence of sub-fields, each a recipient specifier, one for each copy recipient.

CopyRecipientsField ::= SEQUENCE OF CopyRecipientsSubfield

CopyRecipientsSubfield ::= RecipientSpecifier

The phrase "copy recipients" above is not precisely defined by this Recommendation, it is given meaning by users.

Note – The copy recipients, e.g. might be those users to whom, and those DLs to whose members the IPM is conveyed for information.

7.2.6 Blind Copy Recipients

The **Blind Copy Recipients** heading field (C) identifies zero or more users and DLs who are intended *blind* copy "recipients" of the IPM. It also identifies the responses the authorizing users ask of each of those users and of each member of those DLs. It comprises a Sequence of sub-fields, each a recipient specifier, one for each *blind* copy recipient.

BlindCopyRecipientsField ::= SEQUENCE OF BlindCopyRecipientsSubfield

BlindCopyRecipientsSubfield ::= RecipientSpecifier

The phrase "copy recipients" above has the same meaning as in 7.2.5. A **blind** copy recipient is one whose role as such is disclosed to neither primary nor copy recipients.

In the instance of an IPM intended for a blind copy recipient, this conditional field shall be present and identify that user or DL. Whether it shall also identify the other blind copy recipients is a local matter. In the instance of the IPM intended for a primary or copy recipient, the field shall be absent or identify no users or DLs.

7.2.7 *Replied-to IPM*

The **Replied-to IPM** heading field (C) identifies the IPM to which the present IPM is a reply. It comprises an IPM identifier.

```
RepliedToIPMField ::= IPMIdentifier
```

This conditional field shall be present if, and only if, the IPM is a reply.

Note – In the context of *forwarding*, care should be taken to distinguish between the *forwarding IPM* and the *forwarded IPM*. This field should identify whichever of these two IPMs to which the reply responds.

7.2.8 *Obsoleted IPMs*

The **Obsoleted IPMs** heading field [D no subfields (i.e. elements)] identifies zero or more IPMs that the authorizing users of the present IPM consider it to obsolete. It comprises a Sequence of sub-fields, each an IPM identifier, one for each IPM.

```
ObsoletedIPMsField      ::= SEQUENCE OF ObsoletedIPMsSubfield

ObsoletedIPMsSubfield ::= IPMIdentifier
```

Note – In the context of *forwarding*, care should be taken to distinguish between the *forwarding IPM* and the *forwarded IPM*. This field should identify whichever of these two IPMs the present IPM obsoletes.

7.2.9 *Related IPMs*

The **Related IPMs** heading field [D no subfields (i.e. elements)] identifies zero or more IPMs that the authorizing users of the present IPM consider related to it. It comprises a Sequence of sub-fields, each an IPM identifier, one for each IPM.

```
RelatedIPMsField      ::= SEQUENCE OF RelatedIPMsSubfield

RelatedIPMsSubfield   ::= IPMIdentifier
```

The word "related" above is not precisely defined by this Recommendation; it is given meaning by users.

Note 1 – A related IPM, e.g. might be one discussed in the Body of the present IPM.

Note 2 – In the context of *forwarding*, care should be taken to distinguish between the *forwarding IPM* and the *forwarded IPM*. This field should identify whichever of these two IPMs is related to the present IPM.

7.2.10 *Subject*

The **Subject** heading field (O) identifies the subject of the IPM. It comprises a Teletex String of from zero to a prescribed number of characters (see Annex K), chosen from the graphic subset of the Teletex String character set. A length of zero is discouraged.

```
SubjectField ::= TeletexString (SIZE (0..ub-subject-field))
```

7.2.11 *Expiry Time*

The **Expiry Time** heading field (O) identifies when the authorizing users consider the IPM to lose its validity. It comprises a date and time.

```
ExpiryTimeField ::= Time
```

7.2.12 *Reply Time*

The **Reply Time** heading field (O) identifies by when the authorizing users request (but do not demand) that any replies to the present IPM be originated. It comprises a date and time.

```
ReplyTimeField ::= Time
```

7.2.13 *Reply Recipients*

The **Reply Recipients** heading field (C) identifies zero or more users and DLs whom the authorizing users request (but do not demand) be among the preferred recipients of any replies to the present IPM. It comprises a Sequence of sub-fields, each an O/R descriptor, one for each user or DL.

```
ReplyRecipientsField      ::= SEQUENCE OF ReplyRecipientsSubfield

ReplyRecipientsSubfield   ::= ORDescriptor
```

This conditional field shall be present if, and only if, the desired reply recipients are other than the originator of the present IPM alone.

Note – If this field is present and identifies several users and DLs, the originator may include himself among them. If he elects not to do so, he will not be considered among the desired reply recipients.

7.2.14 *Importance*

The **Importance** heading field (D *normal*) identifies the importance that the authorizing users attach to the IPM. It may assume any one of the following values: *low*, *normal*, or *high*.

```
ImportanceField ::= ENUMERATED {
        low      (0),
        normal   (1),
        high     (2) }
```

The values above are not defined by this Recommendation; they are given meaning by users.

7.2.15 *Sensitivity*

The **Sensitivity** heading field (C) identifies the sensitivity that the authorizing users attribute to the IPM.

```
SensitivityField ::= ENUMERATED {
        personal              (1),
        private               (2),
        company-confidential  (3) }
```

This field may assume any one of the following values:

a) *personal:* The IPM is conveyed to its preferred recipients as individuals, rather than in their professional capacities.

b) *private:* The IPM should be conveyed to no one other than its preferred recipients.

c) *company-confidential:* The IPM contains information that should be handled according to company-specific procedures.

This conditional field shall be present if, and only if, the IPM is sensitive.

7.2.16 *Auto-forwarded*

The **Auto-forwarded** heading field (D *false*) indicates whether the IPM is the result of *auto-forwarding*. It is a Boolean.

```
AutoForwardedField  ::= BOOLEAN
```

7.2.17 *Extensions*

The **Extensions** heading field [D no *extensions* (i.e. members)] conveys information accommodated by no other heading field. It comprises a Set of zero or more **IPMS extensions** (or **extensions**), each conveying one such information item.

```
ExtensionsField  ::= SET OF IPMSExtension

IPMSExtension  ::= SEQUENCE {
        type    OBJECT IDENTIFIER,
        value   ANY DEFINED BY type DEFAULT NULL NULL }
```

Each extension has the following components:

a) **Type** (M): Identifies the semantics and restricts the abstract syntax of the *Value* component. An Object Identifier.

b) **Value** (D null): An information item whose abstract syntax is restricted only by the Type component. An Any.

The Type components of all the extensions in the Extensions field shall differ. Not every defined extension need appear in the field.

All heading extensions defined in this Recommendation are contained in annex A. An extension whose Type component is not understood may be ignored.

Every extension is defined by means of the following macro.

```
IPMS-EXTENSION MACRO  ::=
BEGIN
        TYPE  NOTATION    ::= "VALUE" type | empty
        VALUE NOTATION    ::= value (VALUE OBJECT IDENTIFIER)
END
```

An instance of the macro's type notation identifies the data type to which the extension's Value component shall be restricted. If no type is identified explicitly, Null is implied.

An instance of the macro's value notation identifies the Object Identifier that shall appear as the extension's Type component.

Note 1 – Future versions of this Recommendation may define additional extensions. Furthermore, future versions are likely to add information to the Heading only by means of this field.

Note 2 – Whenever an IPMS Extension is defined, it should also be considered whether new MS attributes should also be defined (see Annex C), and whether the UA operation needs to be modified (see clause 18). As any extension may be ignored, privately-defined extensions may not mandate support of the new MS attributes or UA operation.

7.3 *Body part types*

The types of body parts that may appear in the Body of an IPM are defined and described below.

```
BodyPart ::= CHOICE {
        ia5-text              [0]   IA5TextBodyPart,
        g3-facsimile          [3]   G3FacsimileBodyPart,
        g4-class1             [4]   G4Class1BodyPart,
        teletex               [5]   TeletexBodyPart,
        videotex              [6]   VideotexBodyPart,
        encrypted             [8]   EncryptedBodyPart,
        message               [9]   MessageBodyPart,
        mixed-mode           [11]   MixedModeBodyPart,
        bilaterally-defined  [14]   BilaterallyDefinedBodyPart,
        nationally-defined    [7]   NationallyDefinedBodyPart,
        externally-defined   [15]   ExternallyDefinedBodyPart }
```

Body parts of some of the types defined below have two components, *Parameters* and *Data*. The **Parameters** component (M) comprises a Sequence of information items that describe the information object the body part represents and that typically are format and control parameters.

The **Data** component (M) is the information object itself.

Note 1 – In CCITT Recommendation X.420 (1984), context-specific tags 1 and 10 denote Telex and Simple Formattable Document body parts, respectively, which are no longer defined. In ISO DP 9065, context-specific tags 12 and 13 denote ODA and ISO 6937 Text body parts, respectively, which are no longer defined. In CCITT Recommendation X.420 (1984), CCITT Rec. X.420 (1988) and ISO/IEC 10021-7:1990, context-specific tag 2 denotes the Voice *basic* body part which is no longer defined. These tags, therefore, are avoided in BodyPart.

Note 2 – Under some circumstances, an IPM may be subjected to conversion while in transit between users. Such a transmittal event may alter a body part's type.

7.3.1 *IA5 Text*

An **IA5 Text** body part represents IA5 text. It has Parameters and Data components.

```
IA5TextBodyPart ::= SEQUENCE {
        parameters      IA5TextParameters,
        data            IA5TextData }

IA5TextParameters ::= SET {
        repertoire      [0] Repertoire DEFAULT ia5 }

IA5TextData ::= IA5String
```

The Parameters component comprises the following parameters:

– **Repertoire** (D *IA5*): Identifies the character set to which the Data component is constrained.

```
Repertoire ::= ENUMERATED {
        ita2  (2),
        ia5   (5) }
```

This parameter may assume any one of the following values:

i) *ITA2:* The Data component shall be limited to the ITA2 (i.e. telex) character set.

ii) *IA5* The Data component may draw upon the full IA5 character set.:

The Data component is the text, an IA5 String. It may contain lines of any length. Whenever the component is rendered (e.g. displayed to or printed for a user), all (rather than only a part) of the text must be communicated (e.g. lines may be folded but shall not be truncated).

Note – Many terminals have a maximum line length of 80 characters. Therefore, lines that do not exceed that length are most likely to be satisfactorily rendered (e.g. are most likely to avoid being folded).

7.3.2 *Voice*

Note – The original definition of this body part has been replaced by the definition in B.4.

A **Voice** body part represents speech. It has Parameters and Data components.

```
VoiceBodyPart ::= SEQUENCE {
        parameters      VoiceParameters,
        data            VoiceData }

VoiceParameters ::= SET -- for further study

VoiceData ::= BIT STRING -- for further study
```

The parameters of such a body part, and the digitized speech encoding technique that those parameters might identify and parameterize, are for further study**.**

The Data component is the speech, a Bit String.

7.3.3 *G3 Facsimile*

A **G3 Facsimile** body part represents Group 3 facsimile images. It has Parameters and Data components.

```
G3FacsimileBodyPart ::= SEQUENCE {
        parameters      G3FacsimileParameters,
        data            G3FacsimileData }

G3FacsimileParameters ::= SET {
        number-of-pages         [0] INTEGER OPTIONAL,
        non-basic-parameters    [1] G3FacsimileNonBasicParameters OPTIONAL }

G3FacsimileData ::= SEQUENCE OF BIT STRING
```

The Parameters component comprises the following parameters:

a) **Number-of-pages** (O): Identifies the number of pages of Group 3 facsimile data present in the Data component. A non-negative Integer.

b) **Non-basic-parameters** (C): Identifies the non-basic parameters (NBPs) for Group 3 facsimile that characterize the Data component. A G3 NBPs descriptor.

   This conditional parameter may be absent if the data component is of basic G3 Facsimile type. If the data component is of a non-basic type, it shall be present if (but not only if) the Body contains two or more G3 Facsimile body parts. The absence of this parameter when the data component is of a non-basic type is discouraged.

   Note – Its absence in these conditions provides compatibility with Recommendation X.420 (1984).

The Data component is the facsimile images, a Sequence of Bit Strings, each encoding a single page of Group 3 facsimile data as dictated by Recommendation T.4, but filled out to a multiple of 8 bits with additional zero bits, and with each group of 8 bits reversed such that, for each page of the T.4 data:

– the 1st bit in the T.4 data becomes the 8th bit in the G3FacsimileData bit string,

– the 8th bit in the T.4 data becomes the 1st bit in the G3FacsimileData bit string,

– the 9th bit in the T.4 data becomes the 16th bit in the G3FacsimileData bit string,

– the 16th bit in the T.4 data becomes the 9th bit in the G3FacsimileData bit string, etc.

The Return-To-Control signal (defined in Recommendation T.4) shall be present at the end of each page of T.4 data.

Note 1 – The Number-of-pages component identifies the number of elements in the Sequence that constitutes the Data component and is thus redundant.

Note 2 – If the Body comprises a single such body part, its NBPs may (but need not) be conveyed by means of the envelope of the message that contains the IPM.

Note 3 – Where the body part has been received from a facsimile terminal, the Return-To-Control signal may be encoded in the form that it was received from the terminal. Where received over an unreliable network, the Return-To-Control signal may be subject to error.

7.3.4 *G4 Class 1*

A **G4 Class 1** body part represents a final-form document of the sort that is processable by Group 4 Class 1 facsimile terminals. It comprises a Sequence of interchange data elements, defined in Recommendation T.415, which describe the document's layout structure.

```
G4Class1BodyPart ::= SEQUENCE OF Interchange-Data-Element
```

7.3.5 *Teletex*

A **Teletex** body part represents a Teletex document. It has Parameters and Data components.

```
TeletexBodyPart ::= SEQUENCE {
        parameters   TeletexParameters,
        data         TeletexData }

TeletexParameters ::= SET {
        number-of-pages       [0] INTEGER OPTIONAL,
        telex-compatible      [1] BOOLEAN DEFAULT FALSE,
        non-basic-parameters  [2] TeletexNonBasicParameters OPTIONAL }

TeletexData ::= SEQUENCE OF TeletexString
```

The Parameters component comprises the following parameters:

a) **Number-of-pages** (O): Identifies the number of pages of Teletex text present in the Data component. A non-negative Integer.

b) **Telex-compatible** (D *false*): Indicates whether the document in the Data component is telex-compatible. A Boolean.

   If this parameter has the value *true*, every Teletex String in the Data component shall be restricted to the ITA2 character set. No line shall exceed 69 characters in length.

c) **Non-basic-parameters** (C): Identifies the NBPs for Teletex that characterize the Data component. A Teletex NBPs descriptor.

   This conditional parameter may be absent if the data component is of basic Teletex type. If the data component is of a non-basic type, it shall be present if (but not only if) the Body contains two or more Teletex body parts. The absence of this parameter when the data component is of a non-basic type is discouraged.

   Note – Its absence in these conditions provides compatibility with Recommendation X.420 (1984).

The Data component is the document, a Sequence of Teletex Strings, each of which encodes one of its pages. The text of every page (including the first page of the document) shall be introduced either by Form Feed and Carriage Return or by Carriage Return and Form Feed. This sequence shall be preceded by Identify Graphic Subrepertoire under the conditions stated in Recommendation T.61.

Note 1 – The Number-of-pages component identifies the number of elements in the Sequence that constitutes the Data component, and is thus redundant.

Note 2 – If the Body comprises a single such body part, its NBPs may (but need not) be conveyed by means of the envelope of the message that contains the IPM.

Note 3 – The initial Form Feed and Carriage Return on each page of a Teletex document is required by Recommendation T.61.

7.3.6 *Videotex*

A **Videotex** body part represents Videotex data. It has Parameters and Data components.

```
VideotexBodyPart ::= SEQUENCE {
        parameters   VideotexParameters,
        data         VideotexData }

VideotexParameters ::= SET {
        syntax  [0]  VideotexSyntax OPTIONAL }

VideotexData ::= VideotexString
```

The Parameters component comprises the following parameters:

– **Syntax** (O): Identifies the syntax of the Data component. In the parameter's absence, the syntax shall be considered unspecified.

```
VideotexSyntax ::= INTEGER {
        ids           (0),
        data-syntax1  (1),
        data-syntax2  (2),
        data-syntax3  (3) }
```

This parameter may assume any one of the following values, each of which denotes as follows one of the Videotex syntaxes defined in Recommendations T.100 and T.101:

i) *ids:* The IDS syntax.

ii) *data-syntax1:* Data Syntax 1.

iii) *data-syntax2:* Data Syntax 2.

iv) *data-syntax3:* Data Syntax 3.

The Data component is the Videotex data, a Videotex String. It shall conform to the Videotex syntax denoted by the Syntax parameter.

7.3.7 *Encrypted*

An **Encrypted** body part represents the result of encrypting a body part of a type defined by this Recommendation. It has Parameters and Data components.

```
EncryptedBodyPart ::= SEQUENCE {
        parameters  EncryptedParameters,
        data        EncryptedData }

EncryptedParameters ::= SET OF ANY -- for further study

EncryptedData ::= BIT STRING -- for further study
```

The parameters of such a body part, and the encryption technique that those parameters might identify and parameterize, are for further study.

The Data component is the encrypted body part, a Bit String. The bits of the Bit String shall encrypt a data value of (ASN.1) type BodyPart encoded in accordance with the Basic Encoding Rules of CCITT Rec. X.209 | ISO/IEC 8825.

7.3.8 *Message*

A **Message** body part represents an IPM and, optionally, its delivery envelope. It has Parameters and Data components.

```
MessageBodyPart ::= SEQUENCE {
        parameters  MessageParameters,
        data        MessageData }

MessageParameters ::= SET {
        delivery-time       [0] MessageDeliveryTime OPTIONAL,
        delivery-envelope   [1] OtherMessageDeliveryFields OPTIONAL }

MessageData ::= IPM
```

The Parameters component comprises the following parameters:

a) **Delivery-time** (O): The date and time the IPM was delivered. The presence of this component in the absence of the Delivery-envelope component is discouraged.

b) **Delivery-envelope** (O): The IPM's other message delivery fields. The presence of this component in the absence of the Delivery-time component is discouraged.

The Data component is the IPM.

Including one IPM in another as described in the present clause is called **forwarding** that IPM. The enclosing IPM is called the **forwarding IPM**, the enclosed IPM the **forwarded IPM**.

Note 1 – The possible future inclusion of a message identifier in the Parameters component is for further study. Its present omission provides compatibility with Recommendation X.420 (1984).

Note 2 – That the IPM and purported delivery envelope of a Message body part are, in any sense, genuine is unverified.

7.3.9 *Mixed-mode*

A **Mixed-mode** body part represents a final-form document of the sort that is processable by mixed-mode Teletex terminals and Group 4 Classes 2 and 3 facsimile terminals. It comprises a Sequence of interchange data elements, defined in Recommendation T.415, which describe the document's layout structure.

```
MixedModeBodyPart ::= SEQUENCE OF Interchange-Data-Element
```

7.3.10 *Bilaterally Defined*

A **Bilaterally Defined** body part represents an information object whose semantics and abstract syntax are bilaterally agreed by the IPM's originator and all of its potential recipients. It comprises an Octet String.

```
BilaterallyDefinedBodyPart ::= OCTET STRING
```

Note – The use of this body part type is discouraged. It predates the Externally Defined body part type and is retained for backward compatibility with Recommendation X.420 (1984). The Externally Defined body part type provides the same capabilities and more, and its use is preferred, e.g. because such use clearly distinguishes between the body parts defined by one community of users and those defined by another.

7.3.11 *Nationally Defined*

A **Nationally Defined** body part represents an information object whose semantics and abstract syntax are nationally defined by a country whose identity is bilaterally agreed by the IPM's originator and all of its potential recipients. It comprises an Any.

```
NationallyDefinedBodyPart ::= ANY
```

Note 1 – This body part type is intended for use in domestic communication where the country in question is implicitly that of the originator and all of the potential recipients.

Note 2 – The use of this body part type is discouraged. It predates the Externally Defined body part type and is retained for backward compatibility with Recommendation X.420 (1984). The Externally Defined body part type provides the same capabilities and more, and its use is preferred, e.g. because such use clearly distinguishes between the body parts defined by one country and those defined by another.

7.3.12 *Externally Defined*

An **Externally Defined** body part represents an information object whose semantics and abstract syntax are denoted by an Object Identifier which the body part carries. It has Parameters and Data components.

```
ExternallyDefinedBodyPart ::= SEQUENCE {
        parameters  [0]  ExternallyDefinedParameters OPTIONAL,
        data             ExternallyDefinedData }

ExternallyDefinedParameters ::= EXTERNAL

ExternallyDefinedData ::= EXTERNAL
```

The Parameters and Data components are Externals (see clause 32 of CCITT Rec. X.208 | ISO/IEC 8824). Their Direct-reference components shall be present, their Indirect-reference and Data-value-descriptor components absent.

On the basis of the Externally Defined body part type, all body part types are divided into two important classes as follows:

a) **basic**: Said of any body part type except Externally Defined. Denoted by an integer (an ASN.1 context-specific tag).

   All basic body part types are defined in I.1 of Annex I.

b) **extended**: Said of the Externally Defined body part type restricted to any one value of the Direct-reference component of the Data component of such a body part. Denoted by an Object Identifier.

   Some (but not necessarily all) extended body part types are defined in Annex B.

Every extended body part type this Recommendation defines is defined by means of the following macro. Every extended body part type defined elsewhere shall be so defined as well.

```
EXTENDED-BODY-PART-TYPE MACRO ::=
BEGIN

                              TYPE  NOTATION  ::= Parameters Data
                              VALUE NOTATION ::= value (VALUE OBJECT IDENTIFIER)

Parameters    ::= "PARAMETERS" type "IDENTIFIED" "BY" value (OBJECT IDENTIFIER) |
                  empty
Data          ::= "DATA" type
END
```

An instance of the macro's type notation defines, by means of its PARAMETERS clause, the type of the data value that is represented by the Parameters component of such an (Externally Defined) body part (an External), and the Object Identifier that appears in the Direct-reference component of this Parameters component. The presence of the PARAMETERS clause implies the presence of the Parameters component in every instance of the extended body part type; its omission implies the absence of the Parameters component in every instance. An instance of the type notation also defines, by means of its DATA clause, the type of the data value that is represented by the Data component of such a body part (an External).

An instance of the macro's value notation defines the Object Identifier that appears as the Direct-reference component of the Data component of such an (Externally Defined) body part. The Object Identifier identifies the encoding rules for the body part. Those body parts whose types this Recommendation defines shall be encoded using ASN.1's basic encoding rules.

Note 1 – This body part type enables the exchange of information objects of all kinds, each unambiguously and uniquely identified. This identification relies upon the Direct-reference component mentioned above, which is an Object Identifier. Object Identifiers are easily obtained, e.g. by national bodies and private organizations.

Note 2 – If an Externally Defined body part has a Parameters component, the Object Identifier in its Direct-reference component is allocated at the same time and by the same naming authority as that in the Direct-reference component of the Data component.

Note 3 – When a new extended body-part type and a new EIT are defined and have a one to one relationship, then the same Object Identifier may be used for the data component and the EIT.

Note 4 – Like body parts of other types, an Externally Defined body part may be subjected to conversion. However, specification of the conversion algorithms may be outside the scope of Recommendation X.408.

Note 5 – The basic body part types exist for purely historical reasons, predating the Externally Defined body part type.

8 Interpersonal notifications

An **interpersonal notification** (**IPN**) is a member of a secondary class of information object conveyed between users in Interpersonal Messaging.

```
IPN ::= SET {
        -- common-fields  -- COMPONENTS OF CommonFields,
        choice [0] CHOICE {
                non-receipt-fields              [0]  NonReceiptFields,
                receipt-fields                  [1]  ReceiptFields,
                other-notification-type-fields  [2]  OtherNotificationTypeFields } }
```

An IPN may take any of the following forms:

a) **non-receipt notification** (**NRN**): An IPN that reports its originator's failure to receive, to accept, or his delay in receiving, an IPM.

NRN ::= IPN *-- with non-receipt-fields chosen*

b) **receipt notification** (**RN**): An IPN that reports its originator's receipt, or his expected and arranged future receipt, of an IPM.

RN ::= IPN *-- with receipt-fields chosen*

c) **other notification** (**ON**): An IPN that reports some other event concerning an IPM.

ON ::= IPN *-- with other-notification-type-fields chosen*

There are no ONs defined in this Recommendataion. Specific uses of ON may be defined in future versions of this Recommendation to support extended semantics of an IPN, such as secure notifications.

The IPM to which an IPN refers is called the **subject IPM**. Only a UA to which the subject IPM is actually delivered shall originate an IPN relating to it, and it shall originate at most one such IPN which shall be conveyed to the subject IPM's originator alone.

An actual recipient shall originate an IPN only in accordance with the Notification-requests component of the *subject recipient specifier*. The **subject recipient specifier** is that recipient specifier in the subject IPM's Heading as a result of which the subject IPM is delivered to that user.

The subject recipient specifier is determined by examining the Sequences of recipient specifiers that constitute the subject IPM's Primary, Copy, and Blind Copy Recipients heading fields. The fields are examined in the order in which they are mentioned in the preceding sentence. Within each field, the specifiers are examined in the order in which they appear there. The subject recipient specifier is the first one found whose Recipient component has as its value an O/R descriptor whose Formal-name component is present and has as its value an O/R name of the preferred recipient as a result of which the subject IPM was delivered to the user on whose behalf the examination is performed.

An IPN comprises a Set of information items called **notification fields** (or **fields**), each of which is of one of the following classes:

a) **common field**: A notification field applicable to both NRNs and RNs.

b) **non-receipt field**: A notification field applicable to NRNs alone.

c) **receipt field**: A notification field applicable to RNs alone.

d) **other receipt notification type fields**: A notification field applicable to ONs alone.

The structure of an IPN is depicted in Figure 2/X.420.

The fields, in each of the above classes, that may appear in an IPN are defined and described below.

8.1 *Common fields*

The common fields are defined and described below.

```
CommonFields ::= SET {
        subject-ipm                 SubjectIPMField,
        ipn-originator          [1] IPNOriginatorField OPTIONAL,
        ipm-preferred-recipient [2] IPMPreferredRecipientField OPTIONAL,
        conversion-eits             ConversionEITsField OPTIONAL,
        notification-extensions [3] NotificationExtensionsField OPTIONAL }
```

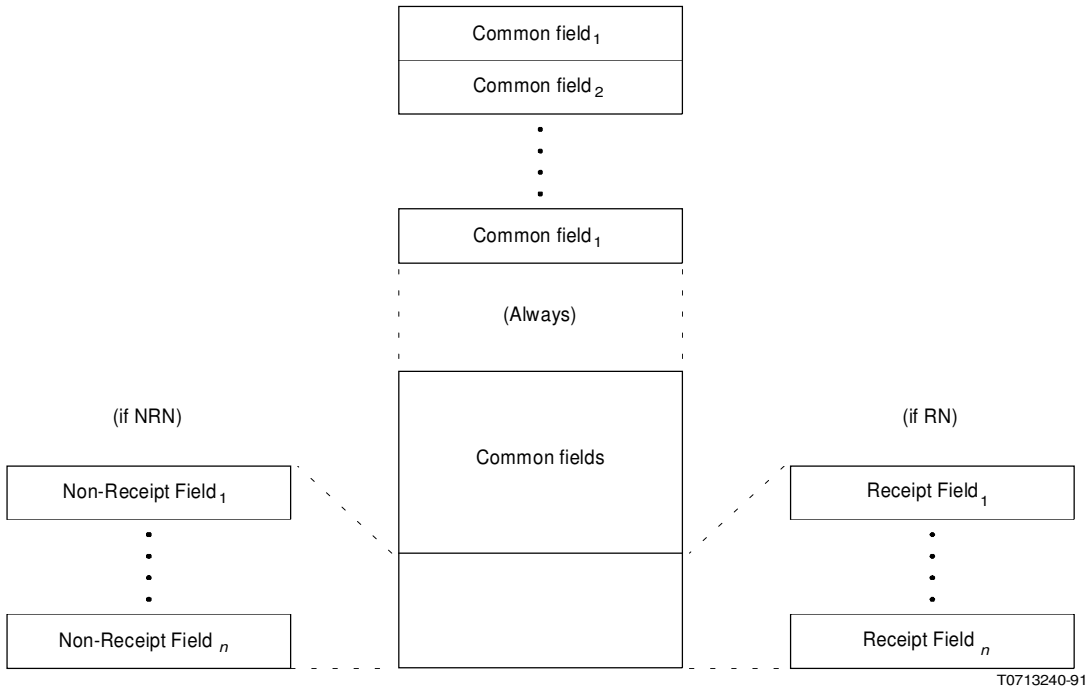

FIGURE 2/X.420
An interpersonal notification

8.1.1 *Subject IPM*

The **Subject IPM** common field (M) identifies the subject IPM. It comprises an IPM identifier.

SubjectIPMField ::= IPMIdentifier

8.1.2 *IPN Originator*

The **IPN Originator** common field (O) identifies the IPN's originator. It comprises an O/R descriptor.

IPNOriginatorField ::= ORDescriptor

If the IPN's originator is a preferred recipient of the subject IPM, the O/R descriptor above shall be precisely that which is the value of the Recipient component of the subject recipient specifier.

8.1.3 *IPM Preferred Recipient*

The **IPM Preferred Recipient** common field (C) identifies the preferred recipient of the subject IPM who gives rise to its delivery to the IPN's originator (an alternate, (**DL**) member, or substitute recipient). It comprises an O/R descriptor.

IPMPreferredRecipientField ::= ORDescriptor

The O/R descriptor above shall be precisely that which is the value of the Recipient component of the subject recipient specifier.

This conditional field shall be present if, and only if, it would identify a user other than the IPN's originator or a DL.

8.1.4 *Conversion EITs*

The **Conversion EITs** common field (C) identifies the EITs of the subject IPM upon delivery to the IPN's originator. It comprises an EITs descriptor.

```
ConversionEITsField ::= EncodedInformationTypes
```

This conditional field shall be present if, and only if, the IPM was subjected to conversion for delivery to the IPN's originator.

8.1.5 *Notification Extensions*

The **Notification Extensions** common field (O) allows for future extensions to the IPN.

```
NotificationExtensionsField ::= SET OF IPMSExtension
```

There are no Notification Extensions defined in this Recommendation.

8.2 *Non-receipt fields*

The non-receipt fields are defined and described below.

```
NonReceiptFields ::= SET {
        non-receipt-reason      [0]  NonReceiptReasonField,
        discard-reason          [1]  DiscardReasonField OPTIONAL,
        auto-forward-comment    [2]  AutoForwardCommentField OPTIONAL,
        returned-ipm            [3]  ReturnedIPMField OPTIONAL,
        nrn-extensions          [4]  NRNExtensionsField OPTIONAL }
```

8.2.1 *Non-receipt Reason*

The **Non-receipt Reason** non-receipt field (M) indicates why the NRN's originator has not received the subject IPM (even though it was delivered to him).

```
NonReceiptReasonField ::= ENUMERATED {
        ipm-discarded       (0),
        ipm-auto-forwarded  (1) }
```

This field may assume any one of the following values:

a) *ipm-discarded:* The IPM was discarded. This case is further illumined by the *Discard Reason* field.

b) *ipm-auto-forwarded:* The IPM was auto-forwarded. This case is further illumined by the *Auto-forward Comment* field.

8.2.2 *Discard Reason*

The **Discard Reason** non-receipt field (C) indicates why the subject IPM was discarded (subsequent to its delivery to the NRN's originator and prior to its receipt).

```
DiscardReasonField ::= ENUMERATED {
        ipm-expired                 (0),
        ipm-obsoleted               (1),
        user-subscription-terminated (2) }
```

This field may assume any one of the following values:

a) *ipm-expired:* Auto-discard was in effect, expired IPMs were being discarded, and the time identified by the subject IPM's Expiry Time heading field had arrived.

b) *ipm-obsoleted: Auto-discard* was in effect, obsolete IPMs were being discarded, and the Obsoleted IPMs heading field of another IPM, delivered to the NRN's originator, identified the subject IPM.

c) *user-subscription-terminated:* The Interpersonal Messaging subscription of the NRN's originator was terminated.

This conditional field shall be present only if the Non-receipt Reason field has the value *ipm-discarded*. In the absence of this field the reason for discarding is not specified.

8.2.3 *Auto-forward Comment*

The **Auto-forward Comment** non-receipt field (C) is information pre-supplied for this purpose by the NRN's originator. It comprises a Printable String of from zero to a prescribed number of characters (see Annex K), chosen from the Printable String character set. A length of zero is discouraged.

```
AutoForwardCommentField ::= AutoForwardComment

AutoForwardComment      ::= PrintableString
        (SIZE (0..ub-auto-forward-comment) )
```

The value of this field shall be precisely the auto-forward-comment argument of the *Change Auto-forwarding* abstract operation as a result of which the subject IPM was auto-forwarded.

This conditional field shall be present if, and only if, the Non-receipt Reason field has the value *ipm-auto-forwarded* and the auto-forward-comment argument above was supplied.

8.2.4 *Returned IPM*

The **Returned IPM** non-receipt field (C) is precisely the subject IPM.

```
ReturnedIPMField ::= IPM
```

This conditional field shall be present if, and only if, *ipm-returned* is among the values of the Notification-requests component of the subject recipient specifier and the subject IPM was not subjected to conversion for delivery to the NRN's originator.

8.2.5 *NRN Extensions*

The **NRN Extensions** field (O) allows for future extensions to the structure of an NRN.

```
NRNExtensionsField ::= SET OF IPMSExtension
```

There are no NRN Extensions defined in this Recommendation.

8.3 *Receipt fields*

The receipt fields are defined and described below.

```
ReceiptFields ::= SET {
        receipt-time            [0]  ReceiptTimeField,
        acknowledgment-mode     [1]  AcknowledgmentModeField DEFAULT manual,
        suppl-receipt-info      [2]  SupplReceiptInfoField OPTIONAL,
        rn-extensions           [3]  RNExtensionsField OPTIONAL }
```

8.3.1 *Receipt Time*

The **Receipt Time** receipt field (M) identifies when the RN's originator received the subject IPM. It comprises a date and time.

```
ReceiptTimeField ::= Time
```

8.3.2 *Acknowledgment Mode*

The **Acknowledgment Mode** receipt field (D *manual*) identifies the manner in which the RN was originated.

```
AcknowledgmentModeField ::= ENUMERATED {
        manual      (0),
        automatic   (1) }
```

This field may assume any one of the following values:

a) *manual:* The RN was originated by means of the *Originate RN* abstract operation.

b) *automatic:* The RN was originated as a result of *auto-acknowledgment*.

8.3.3 *Suppl Receipt Info*

The **Suppl Receipt Info** receipt field (O) gives supplementary information about the receipt of the subject IPM by the RN's originator. It comprises a Printable String of from zero to a prescribed number of characters (see Recommendation X.411), chosen from the Printable String character set.

```
SupplReceiptInfoField ::= SupplementaryInformation
```

8.3.4 *RN Extensions*

The **RN Extensions** field (O) allows for future extensions to the structure of an RN.

```
RNExtensionsField ::= SET OF IPMSExtension
```

There are no RN Extensions defined in this Recommendation.

8.4 *Other notification type fields*

The other notification type fields relate to an ON. There are no ONs defined in this Recommendation.

```
OtherNotificationTypeFields ::= SET OF IPMSExtension
```

SECTION 3 – ABSTRACT SERVICE DEFINITION

9 Overview

This section defines the abstract service that characterizes Interpersonal Messaging, and describes the environment in which that service is supplied and consumed. It does both using the abstract service definition conventions of CCITT Rec. X.407 | ISO/IEC 10021-3.

This section covers the following topics:

a) Primary object types;

b) Primary port types;

c) Abstract operations;

d) Abstract errors;

e) Other capabilities.

10 Primary Object types

The environment in which Interpersonal Messaging takes place can be modelled as an abstract object which is hereafter referred to as the **Interpersonal Messaging Environment** (**IPME**).

```
ipme OBJECT
        ::= id-ot-ipme
```

When refined (i.e. functionally decomposed), the IPME can be seen to comprise lesser objects which interact by means of ports.

```
ipme-refinement REFINE ipme AS
        ipms

                origination     [S]   PAIRED WITH ipms-user
                reception       [S]   PAIRED WITH ipms-user
                management      [S]   PAIRED WITH ipms-user
        ipms-user RECURRING
        ::= id-ref-primary
```

The lesser objects are referred to as the **primary objects** of Interpersonal Messaging. They include a single, central object, the *Interpersonal Messaging System* (*IPMS*), and numerous peripheral objects called *Interpersonal Messaging System users* (*IPMS users*).

The structure of the IPME is depicted in Figure 3/X.420.

The primary object types are defined and described below. The types of ports by means of which they interact are discussed in clause 11.

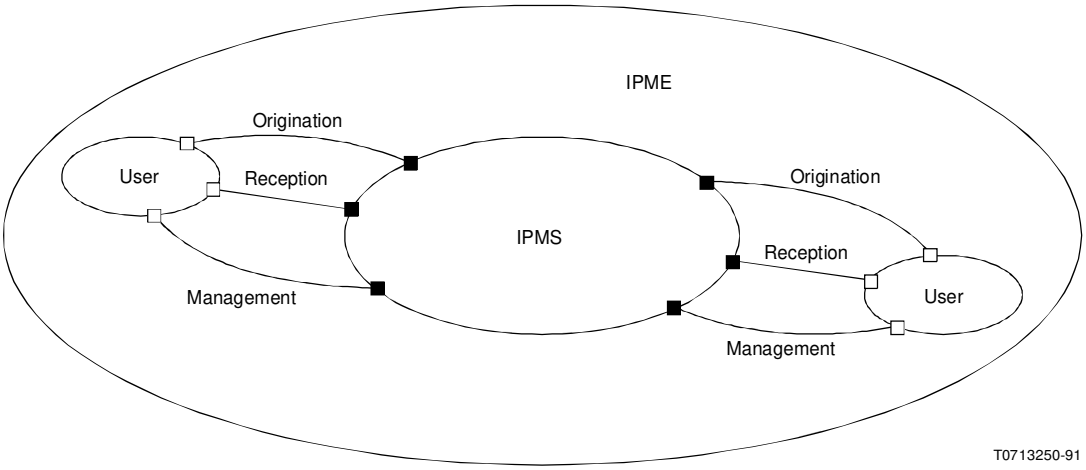

FIGURE 3/X.420
The Interpersonal Messaging Environment

10.1 *Interpersonal Messaging System user*

An **Interpersonal Messaging System user** (**IPMS user**) is a user that engages in Interpersonal Messaging. An IPMS user originates, receives, or both originates and receives information objects of the types defined in Section 2.

```
ipms-user OBJECT
    PORTS {
        origination    [C],
        reception      [C],
        management     [C] }
    ::= id-ot-ipms-user
```

The IPME comprises any number of IPMS users.

Note 1 – As its name suggests, Interpersonal Messaging is typically an activity of people. Often, therefore, this Recommendation uses personal pronouns (e.g. "he") to refer to IPMS users. This practice, however, is not intended to preclude other, atypical uses of Interpersonal Messaging in which IPMS users are not people.

Note 2 – For brevity, the term "user" is used throughout the rest of this Recommendation with the meaning of "IPMS user".

10.2 *Interpersonal Messaging System*

The **Interpersonal Messaging System** (**IPMS**) is the object by means of which all users communicate with one another in Interpersonal Messaging.

```
ipms OBJECT
        PORTS {

                origination     [S],
                reception       [S],
                management      [S] }
        ::= id-ot-ipms
```

The IPME comprises exactly one IPMS.

11 Primary Port types

The primary objects of Interpersonal Messaging are joined to and interact with one another by means of ports. These ports, which the IPMS supplies, are referred to as the **primary ports** of Interpersonal Messaging. They are of the three types defined below.

Note – In clause 16, the IPMS is decomposed into still lesser objects, among which is the MTS. This fact is anticipated in the present clause by the inclusion of certain MTS capabilities in the IPMS Abstract Service.

11.1 *Origination*

An **origination port** is the means by which a single user conveys to the IPMS messages containing information objects of the types defined in Section 2. Through such a port the user originates *interpersonal messages* and *receipt notifications*. In addition, the user may originate probes through such a port.

The IPMS supplies one origination port to each user [with the exception of indirect users served by PDAUs (see 16.5)].

11.2 *Reception*

A **reception port** is the means by which the IPMS conveys to a single user messages containing information objects of the types defined in Section 2. Through such a port the user receives *interpersonal messages* and *interpersonal notifications*. In addition, the user may receive reports through such a port.

The IPMS supplies one reception port to each user.

11.3 *Management*

A **management port** is the means by which a single user changes information about himself on file with the IPMS. By means of such a port the user enables and disables *auto-discard*, *-acknowledgment*, and *-forwarding*.

The IPMS supplies one management port to each user [with the exception of indirect users served by PDAUs (see clause 16.5)].

12 Abstract operations

The **IPMS Abstract Service** is the set of capabilities that the IPMS provides to each user by means of one origination, one reception, and one management port. Those capabilities are modelled as abstract operations, which may encounter abstract errors when invoked.

The abstract operations available at origination, reception, and management ports, respectively, are defined and described below. The abstract errors they may provoke are the subject of clause 13.

Note 1– The IPMS Abstract Service involves neither abstract bind nor abstract unbind operations.

Note 2 – The IPMS authenticates (i.e. establishes the identity of) the typical user before offering the IPMS Abstract Service to him. By this means it can verify, e.g. that the user is an IPMS subscriber. Authentication, where required, is implicit (rather than explicit) in the definition of the IPMS Abstract Service.

Note 3 – The purpose of the IPMS Abstract Service definition is not to prescribe the user interfaces of implementations of portions of the IPMS, but rather to clarify the meaning and intended use of the information objects of Section 2. A user interface need not provide commands in one-to-one correspondence to the service's abstract operations, nor indeed even divide the labour between the user and the IPMS as the service does. Also, the IPMS Abstract Service definition does not model the facilities provided by a Message Store.

Note 4 – In clause 16 below, the IPMS is decomposed into objects among which is the MTS. The present clause reflects this fact by its inclusion of various MTS-defined information items in the IPMS Abstract Service.

12.1 *Origination abstract operations*

The abstract operations available at an origination port are invoked by the user and performed by the IPMS.

```
origination PORT
        CONSUMER INVOKES {
                OriginateProbe,
                OriginateIPM,
                OriginateRN,
                OriginateON }
        ::= id-pt-origination
```

12.1.1 *Originate Probe*

The **Originate Probe** abstract operation originates a probe concerning (a class of) messages whose contents are IPMs.

```
OriginateProbe ::= ABSTRACT-OPERATION
        ARGUMENT SET {
                envelope    [0]  ProbeSubmissionEnvelope,
                content     [1]  IPM }
        RESULT SET {
                submission-identifier   [0]  ProbeSubmissionIdentifier,
                submission-time         [1]  ProbeSubmissionTime }
        ERRORS {
                SubscriptionError,
                RecipientImproperlySpecified }
```

This abstract operation has the following arguments:

a) **Envelope** (M): A probe submission envelope, whose make-up the MTS Abstract Service defines. The UA supplies all but the following envelope components, which the user provides:

   i) The desired per-message options (i.e. per-message indicators and extensions).

   ii) The O/R names of the preferred recipients and the per-recipient options (i.e. originator report request, explicit conversion, and extensions) desired for each.

b) **Content** (M): An instance of the class of IPM whose deliverability is to be probed.

This abstract operation has the following results:

1) **Submission-identifier** (M): The probe submission identifier the MTS assigns to the probe.

2) **Submission-time** (M): The date and time the probe was directly submitted.

12.1.2 *Originate IPM*

The **Originate IPM** abstract operation originates a message whose content is an IPM.

```
OriginateIPM ::= ABSTRACT-OPERATION
        ARGUMENT SET {
                envelope    [0]  MessageSubmissionEnvelope,
                content     [1]  IPM }
        RESULT SET {
                submission-identifier   [0]  MessageSubmissionIdentifier,
                submission-time         [1]  MessageSubmissionTime }
        ERRORS {
                SubscriptionError,
                RecipientImproperlySpecified }
```

This abstract operation has the following arguments:

a) **Envelope** (M): A message submission envelope, whose make-up the MTS Abstract Service defines. The UA supplies all but the following envelope components, which the user provides:

   i) The desired per-message options (i.e. priority, per-message indicators, deferred delivery time, and extensions).

   ii) The O/R names of the preferred recipients and the per-recipient options (i.e, originator report request, explicit conversion, and extensions) desired for each.

b) **Content** (M): The IPM being originated. Its Auto-forwarded heading field shall be absent or have the value *false*.

This abstract operation has the following results:

1) **Submission-identifier** (M): The message submission identifier the MTS assigns to the submission.

2) **Submission-time** (M): The date and time the message was directly submitted.

12.1.3 *Originate RN*

The **Originate RN** abstract operation originates a message whose content is an RN.

```
OriginateRN ::= ABSTRACT-OPERATION
        ARGUMENT SET {
                envelope    [0]  MessageSubmissionEnvelope,
                content     [1]  RN }
        RESULT SET {
                submission-identifier   [0]  MessageSubmissionIdentifier,
                submission-time         [1]  MessageSubmissionTime }
        ERRORS {
                SubscriptionError,
                RecipientImproperlySpecified }
```

An RN shall be originated only by an actual recipient of the subject IPM of whom an RN is requested by means of the Notification-requests component of the subject IPM's subject recipient specifier.

The user shall not have previously originated an RN in response to the subject IPM, by means of either the present abstract operation or auto-acknowledgment.

This abstract operation has the following arguments:

a) **Envelope** (M): A message submission envelope, whose make-up the MTS Abstract Service defines. The UA supplies all but the following envelope components, which the user provides:

   i) The desired per-message options (i.e. priority, per-message indicators, and extensions). Implicit conversion shall be prohibited, priority that of the subject IPM.

   ii) The O/R names of the preferred recipients and the per-recipient options (i.e. explicit conversion and extensions) desired for each.

b) **Content** (M): The RN being originated.

This abstract operation has the following results:

1) **Submission-identifier** (M): The message submission identifier the MTS assigns to the submission.

b) **Submission-time** (M): The date and time the message was directly submitted.

12.1.4 *Originate ON*

There are no ONs defined in this Recommendation. The **Originate ON** abstract operation is for further study**.**

12.2 *Reception abstract operations*

The abstract operations available at a reception port are invoked by the IPMS and performed by the user.

```
reception PORT
    SUPPLIER INVOKES {
            ReceiveReport,
            ReceiveIPM,
            ReceiveRN,
            ReceiveNRN,
            ReceiveON }
    ::= id-pt-reception
```

Note 1 – As abstractly defined, the IPMS provides no storage for received messages because whether or not it does so for a particular user has no impact upon that user's ability to communicate with other users. Thus the provision of storage is a local matter.

Note 2 – Elaborating upon the above, the *Receive IPM* abstract operation, e.g. expels an IPM from the IPMS because its purpose is to clarify the meaning of the receipt transmittal step. In contrast, the capabilities of a user to whom storage for received messages is provided might include a "Display IPM" command that enables the user to view the delivered (and perhaps already received) IPM whose IPM identifier he specifies, and that allows him to do so any number of times by repeatedly invoking the command. The first, but not subsequent uses of the command to view a particular IPM represents the concrete realization of the Receive IPM abstract operation in such an implementation.

12.2.1 *Receive Report*

The **Receive Report** abstract operation receives a report.

```
ReceiveReport ::= ABSTRACT-OPERATION
    ARGUMENT SET {
            envelope            [0]  ReportDeliveryEnvelope,
            undelivered-object  [1]  InformationObject OPTIONAL }
    RESULT
    ERRORS { }
```

The report received may concern any of the following previously originated by the report's recipient:

a) A probe concerning a message whose content was an IPM that was originated with the Originate Probe abstract operation.

b) A message whose content was an NRN that was originated as a result of *auto-discard* of *auto-forward*.

c) A message whose content was an RN that was originated with the Originate RN abstract operation or by *auto-acknowledgment*.

d) A message whose content was an IPM that was originated with the Originate IPM abstract operation or by *auto-forwarding*.

This abstract operation has the following arguments:

1) **Envelope** (M): A report delivery envelope, whose make-up the MTS Abstract Service defines.

2) **Undelivered-object** (C): The content of the message whose status is being reported. An IPM or IPN.

   If the report was provoked by a previous Originate Probe abstract operation invocation, this conditional argument shall be absent. If the report was provoked by a previous Originate IPM abstract operation invocation, the argument shall be present if, and only if, content return was requested. Otherwise (i.e. if the report was provoked by an IPN) the argument shall be absent.

   This abstract operation has no results.

12.2.2 *Receive IPM*

The **Receive IPM** abstract operation receives a message whose content is an IPM.

```
ReceiveIPM ::= ABSTRACT-OPERATION
        ARGUMENT SET {
                envelope    [0]  MessageDeliveryEnvelope,
                content     [1]  IPM }
        RESULT
        ERRORS { }
```

This abstract operation has the following arguments:

a) **Envelope** (M): The message's delivery envelope.

b) **Content** (M): The IPM that is the message's content.

This abstract operation has no results.

12.2.3 *Receive RN*

The **Receive RN** abstract operation receives a message whose content is an RN. The RN is provoked by an IPM originated with the Originate IPM abstract operation.

```
ReceiveRN ::= ABSTRACT-OPERATION
        ARGUMENT SET {
                envelope    [0]  MessageDeliveryEnvelope,
                content     [1]  RN }
        RESULT
        ERRORS { }
```

This abstract operation has the following arguments:

a) **Envelope** (M): The message's delivery envelope.

b) **Content** (M): The RN that is the message's content.

This abstract operation has no results.

12.2.4 *Receive NRN*

The **Receive NRN** abstract operation receives a message whose content is an NRN. The NRN is provoked by an IPM originated with the Originate IPM abstract operation.

```
ReceiveNRN ::= ABSTRACT-OPERATION
        ARGUMENT SET {
                envelope    [0]  MessageDeliveryEnvelope,
                content     [1]  NRN }
        RESULT
        ERRORS { }
```

This abstract operation has the following arguments:

a) **Envelope** (M): The message's delivery envelope.

b) **Content** (M): The NRN that is the message's content.

This abstract operation has no results.

12.2.5 *Receive ON*

There are no ONs defined in this Recommendation. The **Receive ON** abstract operation is for further study**.**

12.3 *Management abstract operations*

The abstract operations available at a management port are invoked by the user and performed by the IPMS.

```
management PORT
        CONSUMER INVOKES {
                ChangeAutoDiscard,
                ChangeAutoAcknowledgment,
                ChangeAutoForwarding }
    ::= id-pt-management
```

12.3.1 *Change Auto-discard*

The **Change Auto-discard** abstract operation enables or disables **auto-discard**, the automatic discard by the IPMS of expired or obsolete IPMs delivered to, but not yet received by the user.

```
ChangeAutoDiscard ::= ABSTRACT-OPERATION
    ARGUMENT SET {
            auto-discard-expired-IPMs     [0]   BOOLEAN,
            auto-discard-obsolete-IPMs    [1]   BOOLEAN }
    RESULT
    ERRORS { }
```

When it auto-discards an IPM, the IPMS originates an NRN on the user's behalf if, and only if, one was requested of him by means of the Notification-requests component of the subject recipient specifier.

This abstract operation has the following arguments:

a) **Auto-discard-expired-IPMs** (M): Whether or not expired IPMs are to be auto-discarded. A Boolean.

b) **Auto-discard-obsolete-IPMs** (M): Whether or not obsolete IPMs are to be auto-discarded. A Boolean.

This abstract operation has no results.

12.3.2 *Change Auto-acknowledgment*

The **Change Auto-acknowledgment** abstract operation enables or disables **auto-acknowledgment**, the automatic origination of RNs by the IPMS on the user's behalf. Such origination occurs upon delivery of IPMs that request RNs of the user by means of the Notification-requests components of their subject recipient specifiers.

```
ChangeAutoAcknowledgment ::= ABSTRACT-OPERATION
    ARGUMENT SET {
            auto-acknowledge-IPMs  [0]  BOOLEAN,
            auto-acknowledge-suppl-receipt-info  [1]
                    SupplementaryInformation OPTIONAL }
    RESULT
    ERRORS {
            SubscriptionError }
```

This abstract operation has the following arguments:

a) **Auto-acknowledge-IPMs** (M): Whether or not IPMs are to be auto-acknowledged. A Boolean.

b) **Auto-acknowledge-suppl-receipt-info** (C): The Suppl Receipt Info receipt field of each RN provoked by auto-acknowledgment.

This conditional argument shall be present if, and only if, the Auto-acknowledge-IPMs argument has the value *true*.

This abstract operation has no results.

12.3.3 *Change Auto-forwarding*

The **Change Auto-forwarding** abstract operation enables or disables **auto-forwarding**, the automatic forwarding of IPMs by the IPMS to pre-specified users or DLs. Such forwarding occurs upon delivery of the IPMs.

```
ChangeAutoForwarding ::= ABSTRACT-OPERATION
    ARGUMENT SET {
        auto-forward-IPMs          [0]  BOOLEAN,
        auto-forward-recipients    [1]  SEQUENCE OF ORName OPTIONAL,
        auto-forward-heading       [2]  Heading OPTIONAL,
        auto-forward-comment       [3]  AutoForwardComment OPTIONAL }
    RESULT
    ERRORS {
        SubscriptionError,
        RecipientImproperlySpecified }
```

The Body of each IPM the IPMS originates as a result of auto-forwarding comprises a single body part of type Message. The content of the message represented by that body part is the forwarded IPM.

When it auto-forwards an IPM, the IPMS originates an NRN on the user's behalf if, and only if, one was requested of him by means of the Notification-requests component of the subject recipient specifier.

This abstract operation has the following arguments:

a) **Auto-forward-IPMs** (M): Whether or not IPMs are to be auto-forwarded. A Boolean.

b) **Auto-forward-recipients** (C): The users or DLs to which IPMs are to be auto-forwarded. A Sequence of O/R names.

This conditional argument shall be present if, and only if, the Auto-forward-IPMs argument has the value *true*.

c) **Auto-forward-heading** (C): The Heading that is to be used for each forwarding IPM. Its Auto-forwarded heading field shall have the value *true*.

This conditional argument shall be present if, and only if, the Auto-forward-IPMs argument has the value *true*.

d) **Auto-forward-comment** (C): The value that is to be supplied as the Auto-forward Comment non-receipt field of each NRN conveyed to the originator of an auto-forwarded IPM.

This conditional argument shall be present if, and only if, the Auto-forward-IPMs argument has the value *true*.

This abstract operation has no results.

Note – This abstract operation is intended to define the essence of auto-forwarding, and not to preclude the provision of more sophisticated auto-forwarding capabilities, e.g. like those of an MS.

13 Abstract errors

The abstract errors that may be reported in response to the invocation of the abstract operations available at origination, reception, and management ports are defined and described below or as part of the MTS Abstract Service definition.

Note – The set of abstract errors represented below is intended to be illustrative, rather than exhaustive.

13.1 *Subscription Error*

The **Subscription Error** abstract error reports that the user has not subscribed to one or more of the elements of service implicit in his invocation of the abstract operation whose performance is aborted.

```
SubscriptionError ::= ABSTRACT-ERROR
        PARAMETER SET {
                problem  [0]  SubscriptionProblem }
```

This abstract error has the following parameters:

– **Problem** (M): The subscription-related problem encountered.

```
SubscriptionProblem ::= ENUMERATED {
        ipms-eos-not-subscribed   (0),
        mts-eos-not-subscribed    (1) }
```

This parameter may assume any one of the following values:

i) *IPMS-eos-not-subscribed:* An IPMS element of service is not subscribed.

ii) *MTS-eos-not-subscribed:* An MTS element of service is not subscribed.

13.2 *Recipient Improperly Specified*

The **Recipient Improperly Specified** abstract error reports that one or more of the O/R names supplied as arguments of the abstract operation whose performance is aborted, or as components of its arguments, are invalid.

This abstract error is defined by the MTS Abstract Service.

14 Other capabilities

In addition to the capabilities embodied in the IPMS Abstract Service, defined above, the IPMS shall transparently extend to each user the other MS and MTS capabilities identified below. (The enumeration of these capabilities necessarily anticipates the fact, stated in clause 16, that MSs and the MTS are among the IPMS' component parts.)

The following additional capabilities shall be provided:

a) *Submission:* Capabilities of the MS or MTS submission port not embodied in the IPMS Abstract Service, e.g, the ability to cancel delivery of a previously originated message whose content is an IPM (but not an RN), if deferred delivery was selected.

b) *Delivery:* Capabilities of the MTS delivery port not embodied in the IPMS Abstract Service, e.g. the ability to temporarily control the kinds of information objects the MTS conveys to the user's *UA*.

c) *Administration:* The capabilities of the MS's or MTS's administration port.

d) *Retrieval:* The capabilities of the MS retrieval port.

In addition to the above and as a local matter, the IPMS may provide to users additional capabilities neither defined nor limited by this Recommendation. Among such capabilities are those of the Directory.

Note – The required capabilities of this clause are excluded from the formal definition of the IPMS Abstract Service for purely pragmatic reasons, in particular, because their inclusion would largely and needlessly reproduce the definitions of the MS and MTS abstract operations upon which the capabilities are based.

SECTION 4 – ABSTRACT SERVICE PROVISION

15 Overview

This section specifies how the IPMS provides the IPMS Abstract Service to users.

This section covers the following topics:

a) Secondary object types;

b) Secondary port types;

c) User agent operation;

d) Message store operation;

e) Message contents;

f) Port realization;

g) Conformance.

16 Secondary Object types

The IPMS can be modelled as comprising lesser objects which interact with one another by means of (additional) ports.

```
ipms-refinement REFINE ipms AS
        mTS
                submission      [S]   PAIRED WITH ipms-ua, ipms-ms
                delivery        [S]   PAIRED WITH ipms-ua, ipms-ms
                administration  [S]   PAIRED WITH ipms-ua, ipms-ms
        ipms-ua  RECURRING
                origination     [S]   VISIBLE
                reception       [S]   VISIBLE
                management      [S]   VISIBLE
        ipms-ms RECURRING
                submission      [S]   PAIRED WITH ipms--ua
                retrieval       [S]   PAIRED WITH ipms-ua
                administration  [S]   PAIRED WITH ipms-ua
        tlma     RECURRING
                origination     [S]   VISIBLE
                reception       [S]   VISIBLE
                management      [S]   VISIBLE
        tlxau    RECURRING
                origination     [S]   VISIBLE
                reception       [S]   VISIBLE
                management      [S]   VISIBLE
        pdau     RECURRING
                reception       [S]   VISIBLE
        ::= id-ref-secondary
```

These lesser objects are referred to as the **secondary objects** of Interpersonal Messaging. They include a single, central object, the MTS, and numerous peripheral objects: *Interpersonal Messaging System user agents (IPMS UAs)*, *Interpersonal Messaging System message stores* (*IPMS MSs*), *telematic agents* (*TLMAs*), *telex access units* (*TLXAUs*), and PDAUs.

The structure of the IPMS is depicted in Figure 4/X.420. As shown by the figure, *IPMS UAs*, *TLMAs*, *TLXAUs*, and PDAUs are the instruments by means of which the IPMS provides the IPMS Abstract Service to users.

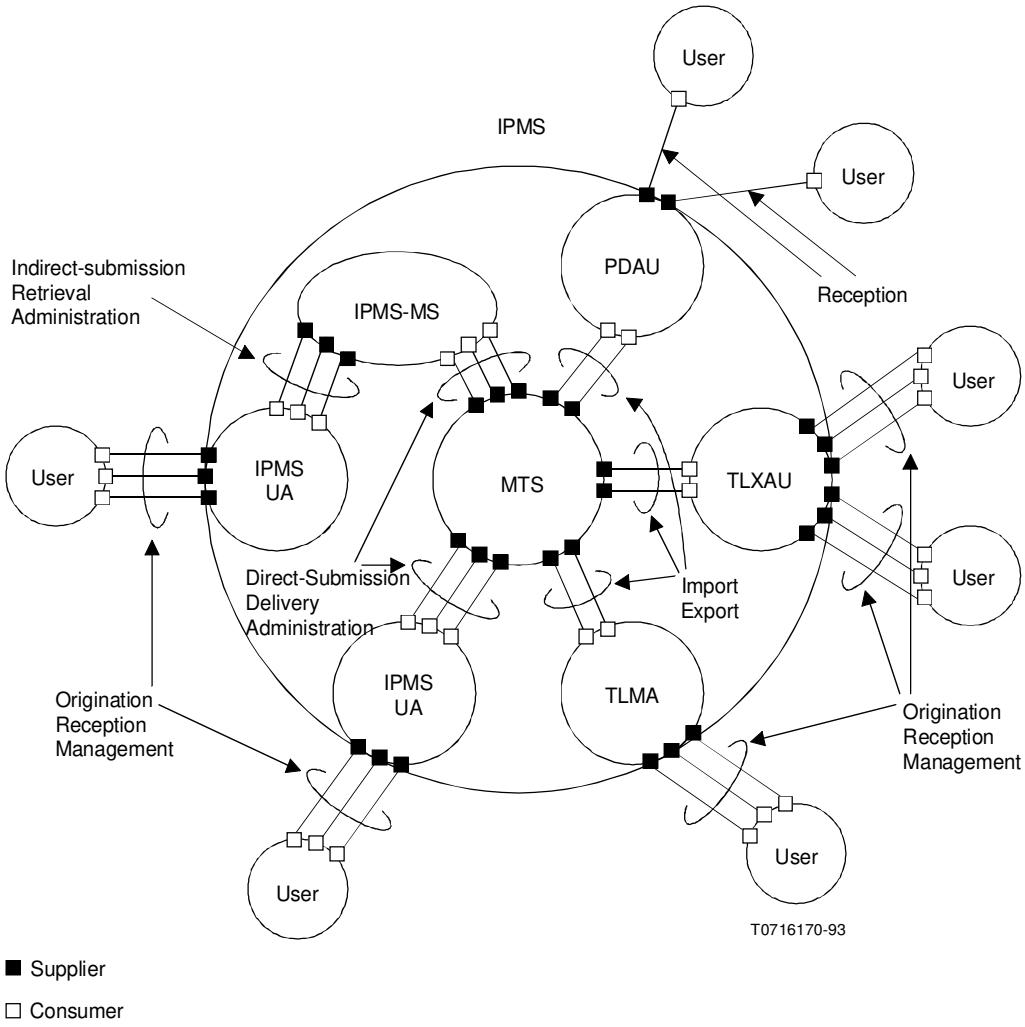

FIGURE 4/X.420
The Interpersonal Messaging System

The secondary object types are defined and described below. The types of ports by means of which they interact are discussed in clause 17.

Note 1 – The refinement above encompasses all possible interconnections of all possible objects. It ignores the possible absence of objects of a particular type (e.g. PDAU), and specific logical configurations of the *IPMS MS*. The latter are identified in Rec. X.402 | ISO/IEC 10021-2.

Note 2 – Recommendation T.330 effectively extends the abstract service of Interpersonal Messaging by its definition of a *miscellanea* port, which is not shown in Figure 4/X.420. See the Note in 16.3.

Note 3 – The MTS supplies import and export ports. However, since those ports are not formally defined (in CCITT Rec. X.411 | ISO/IEC 10021-4), they are not included in the formal refinement above.

16.1 *Interpersonal Messaging System user agent*

An **Interpersonal Messaging System user agent** (**IPMS UA**) is a UA tailored so as to better assist a single user to engage in Interpersonal Messaging. It helps him originate, receive, or both originate and receive messages containing information objects of the types defined in Section 2.

```
ipms-ua OBJECT
        PORTS {
                origination     [S],
                reception       [S],
                management      [S],
                submission      [C],
                delivery        [C],
                retrieval       [C],
                administration  [C] }
        ::= id-ot-ipms-ua
```

The IPMS comprises any number of IPMS UAs.

Note – For brevity, the term "UA" is used throughout the rest of this Recommendation with the meaning of "IPMS UA".

16.2 *Interpersonal Messaging System message store*

An **Interpersonal Messaging System message store** (**IPMS MS**) is an MS tailored so as to better assist a single UA engage in Interpersonal Messaging. It helps it submit, take delivery of, or both submit and take delivery of messages containing information objects of the types defined in Section 2.

```
ipms-ms OBJECT
        PORTS {
                submission      [S],
                retrieval       [S],
                administration  [S],
                submission      [C],
                delivery        [C],
                administration  [C] }
        ::= id-ot-ipms-ms
```

The IPMS comprises any number of IPMS MSs.

Note – For brevity, the term "MS" is used throughout the rest of this Recommendation with the meaning of "IPMS MS".

16.3 *Telematic agent*

A **telematic agent** (**TLMA**) is an AU that helps a single indirect user engage in Interpersonal Messaging from a Telematic terminal, along with that terminal and the network that joins the two. A TLMA helps the user originate, receive, or both originate and receive messages containing information objects of the types defined in Section 2.

```
tlma OBJECT
        PORTS {
                origination     [S],
                reception       [S],
                management      [S],
                miscellanea     [S] }
        ::= id-ot-tlma
```

The IPMS comprises any number of TLMAs.

Note 1 – A TLMA consumes import and export ports. However, since those ports are not formally defined (in CCITT Rec. X.411 | ISO/IEC 10021-4), they are not included in the formal definition of TLMA above.

Note 2 – A TLMA's *miscellanea* port is defined in Recommendation T.330. It is not part of the IPMS Abstract Service in its most general form, which is the subject of this Recommendation. Rather it embodies capabilities available only to a TLMA user. For this reason, it is not considered further here and is not included in the formal refinement of the IPMS found in clause 16.

16.4 *Telex access unit*

A **telex access unit** (**TLXAU**) is an AU that helps any number of indirect users engage in Interpersonal Messaging from Telex terminals. It helps them originate, receive, or both originate and receive messages containing information objects of the types defined in Section 2.

```
tlxau OBJECT
        PORTS {
                origination     [S],
                reception       [S],
                management      [S] }
        ::= id-ot-tlxau
```

The IPMS comprises any number of TLXAUs.

Note – A TLXAU consumes import and export ports. However, since those ports are not formally defined (in CCITT Rec. X.411 | ISO/IEC 10021-4), they are not included in the formal definition of TLXAU above.

16.5 *Physical Delivery Access Unit*

In the present context, a PDAU helps any number of indirect users engage in Interpersonal Messaging by means of a Physical Delivery System (PDS). It helps them receive (but not originate) messages containing information objects of the types defined in Section 2.

```
pdau OBJECT
        PORTS {
                reception       [S] }
        ::= id-ot-pdau
```

The IPMS comprises any number of PDAUs.

Note – A PDAU consumes import and export ports. However, since those ports are not formally defined (in CCITT Rec. X.411 | ISO/IEC 10021-4), they are not included in the formal definition of PDAU above.

16.6 *Message Transfer System*

In the present context, the MTS conveys information objects of the types defined in Section 2 between UAs, MSs, TLMAs, and AUs.

The IPMS comprises a single MTS.

17 Secondary Port types

The secondary objects of Interpersonal Messaging are joined to and interact with one another by means of ports. These ports, which MSs and the MTS supply, are referred to as the **secondary ports** of Interpersonal Messaging. They are of the types identified below.

The capabilities embodied in one submission, one retrieval, and one administration port constitute the MS Abstract Service. They are defined in CCITT Rec. X.413 | ISO/IEC 10021-5.

The capabilities embodied in one submission, one delivery, and one administration port constitute the MTS Abstract Service. They are defined in CCITT Rec. X.411 | ISO/IEC 10021-4.

Note – By means of the abstract bind operation which guards its ports, an MS or the MTS typically authenticates another secondary object before offering its abstract service to that object.

17.1 *Submission*

In the present context, a submission port is the means by which a UA (directly or indirectly) or an MS (directly) submits probes concerning and messages containing information objects of the types defined in Section 2.

An MS supplies one submission port to its UA.

The MTS supplies one submission port to each UA configured without an MS and to each MS.

17.2 *Delivery*

In the present context, a delivery port is the means by which a UA or MS takes delivery of reports concerning and messages containing information objects of the types defined in Section 2.

The MTS supplies one delivery port to each UA configured without an MS and to each MS.

17.3 *Retrieval*

In the present context, a retrieval port is the means by which a UA retrieves reports concerning and messages containing information objects of the types defined in Section 2.

An MS supplies one retrieval port to its UA.

17.4 *Administration*

In the present context, an administration port is the means by which a UA changes information about itself or its user on file with its MS, or a UA or MS changes such information on file with the MTS.

An MS supplies one administration port to its UA.

The MTS supplies one administration port to each UA configured without an MS and to each MS.

17.5 *Import*

In the present context, an import port is the means by which the MTS imports reports concerning and messages containing information objects of the types defined in Section 2.

The MTS supplies one import port to each AU (or TLMA).

17.6 *Export*

In the present context, an export port is the means by which the MTS exports probes concerning and messages containing information objects of the types defined in Section 2.

The MTS supplies one export port to each AU (or TLMA).

18 User agent operation

A UA must employ the MTS in a particular way in order to (correctly) provide the IPMS Abstract Service to its user. If the user is equipped with an MS, the latter contributes to the provision of the abstract service and, therefore, is subject to the same rules.

The rules that govern the operation of a UA (and MS) are the subject of the present clause. The operation of a TLMA or AU is beyond the scope of this Recommendation.

Note 1 – It is for historical reasons that the Recommendation that defines the IPMS Abstract Service also specifies how a UA (and MS), but not a TLMA or AU, provides it.

Note 2 – The purpose of this clause is not to dictate or constrain the implementation of a real UA unnecessarily, but rather to clarify the meaning and intended effect of the IPMS Abstract Service.

18.1 *State variables*

The operation of a UA is described below with the aid of *state variables*. A **state variable** is an information item whose value records the results of the UA's past interactions with its user and influences future interactions. State variables are common to (i.e. shared by) the UA's origination, reception, and management ports.

The UA maintains each state variable continuously, i.e. throughout its user's IPMS subscription. Each Boolean state variable is assigned the value *false* when the subscription commences. The initial values of other state variables are immaterial and therefore unspecified.

The UA alters its state variables when performing or invoking abstract operations. It consults them in determining how to perform, or whether how to invoke abstract operations. Their values (if any) transcend the binding and unbinding of ports.

Note – State variables are pedagogic devices not intended to constrain the implementation of a real UA unnecessarily. In particular, a UA need not maintain run-time data structures corresponding to the state variables if the behaviour required of it can be assured in another way.

18.2 *Performance of Origination operations*

A UA shall perform the abstract operations it makes available at its origination port as prescribed below. The UA alters none of its state variables in the performance of these particular operations.

In the performance of these operations, the UA invokes the following abstract operations of the MTS Abstract Service (which, for the remainder of this clause, are unqualified as to their source):

a) Probe Submission;

b) Message Submission.

Note – In response to the invocation of these abstract operations, a UA reports abstract errors as appropriate. Specification of the precise circumstances under which each abstract error should be reported is beyond the scope of this Recommendation.

18.2.1 *Originate Probe*

A UA shall perform the Originate Probe abstract operation by invoking Probe Submission with the arguments indicated below, and by returning to its user the results indicated below.

The arguments of Probe Submission shall be as follows:

– *Envelope:* The components of this argument that constitute per-probe fields shall be as follows; those not explicitly mentioned below shall be as specified by Originate Probe's Envelope argument:

  i) *Originator-name:* The O/R name of the UA's user.

  ii) *Content-type*, *Content-length*, and *Original-encoded-information-types:* Determined from Originate Probe's Content argument as specified in 20.2 to 20.4.

  iii) *Content-identifier* and *Content-correlator:* Specified or omitted as a local matter.

  The components of this argument that constitute per-recipient fields shall be as specified by Originate Probe's Envelope argument.

The results of Originate Probe shall be as follows:

1) *Submission-identifier:* Probe Submission's Probe-submission-identifier result.

2) *Submission-time:* Probe Submission's Probe-submission-time result.

Note 1 – The UA shall ignore all properties of Originate Probe's Content argument other than those mentioned above.

Note 2 – How the UA employs Probe Submission's Content-identifier result is a local matter.

18.2.2 *Originate IPM*

A UA shall perform the Originate IPM abstract operation by invoking Message Submission with the arguments indicated below, and by returning to its user the results indicated below.

The arguments of Message Submission shall be as follows:

a) *Envelope:* The components of this argument that constitute per-message fields shall be as follows; those not explicitly mentioned below shall be as specified by Originate IPM's Envelope argument:

   i) *Originator-name:* The O/R name of the UA's user.

   ii) *Content-type* and *Original-encoded-information-types:* Determined from Originate IPM's Content argument as specified in 20.2 and 20.4, respectively.

   iii) *Content-identifier* and *Content-correlator:* Specified or omitted as a local matter.

   The components of this argument that constitute per-recipient fields shall be as specified by Originate IPM's Envelope argument.

b) *Content:* Determined from Originate IPM's Content argument (identified as an IPM) as specified in 20.1.

   If the Blind Copy Recipients heading field of the IPM identifies one or more users and DLs, the UA shall invoke Message Submission multiple times, upon each occasion varying the heading field so as to comply with the information hiding requirements of 7.2.6.

The results of Originate IPM shall be as follows:

   1) *Submission-identifier:* Message Submission's Message-submission-identifier result.

   2) *Submission-time:* Message Submission's Message-submission-time result.

Note 1 – How the UA employs Message Submission's Content-identifier result is a local matter.

Note 2 – The inclusion of Message Submission's Extensions result among Originate IPM's results is proper and for further study.

18.2.3 *Originate RN*

A UA shall perform the Originate RN abstract operation by invoking Message Submission with the arguments indicated below, and by returning to its user the results indicated below.

The arguments of Message Submission shall be as follows:

a) *Envelope:* The components of this argument that constitute per-message fields shall be as follows; those not explicitly mentioned below shall be as specified by Originate RN's Envelope argument:

   i) *Originator-name:* The O/R name of the UA's user.

   ii) *Content-type* and *Original-encoded-information-types:* Determined from the RN as specified in 20.2 and 20.4, respectively.

   iii) *Content-identifier* and *Content-correlator:* Specified or omitted as a local matter.

iv) *Deferred-delivery-time:* Omitted.

v) *Per-message-indicators:* notification-type should be set to type 1.

Note – The notification-type indication may become mandatory in a future version of this Recommendation.

The components of this argument that constitute per-recipient fields shall be as specified by Originate RN's Envelope argument.

b) *Content:* Determined from Originate RN's Content argument (identified as an RN) as specified in 20.1.

The results of Originate RN shall be as follows:

1) *Submission-identifier:* Message Submission's Message-submission-identifier result.

2) *Submission-time:* Message Submission's Message-submission-time result.

Note 1 – How the UA employs Message Submission's Content-identifier result is a local matter.

Note 2 – The inclusion of Message Submission's Extensions result among Originate RN's results is proper and for further study**.**

18.2.4 *Originate ON*

Originate ON is for further study.

18.3 *Performance of Management operations*

A UA shall perform the abstract operations it makes available at its management port as specified below. The UA alters one or more of its state variables (see below) in the performance of each operation.

Note – In response to the invocation of these abstract operations, a UA reports abstract errors as appropriate. Specification of the precise circumstances under which each abstract error should be reported is beyond the scope of this Recommendation.

18.3.1 *Change Auto-discard*

To assist it in providing this abstract operation, a UA maintains the following state variables:

a) **Auto-discard-expired-IPMs**: A Boolean that indicates whether or not auto-discard is in effect for expired IPMs.

b) **Auto-discard-obsolete-IPMs**: A Boolean that indicates whether or not auto-discard is in effect for obsolete IPMs.

A UA shall perform the Change Auto-discard abstract operation by recording the values of the Auto-discard-expired-IPMs and Auto-discard-obsolete-IPMs arguments in the correspondingly named state variables.

18.3.2 *Change Auto-acknowledgment*

To assist it in providing this abstract operation, a UA maintains the following state variables:

a) **Auto-acknowledge-IPMs**: A Boolean that indicates whether or not auto-acknowledgment is in effect.

b) **Auto-acknowledge-suppl-receipt-info**: The Suppl Receipt Info field of each RN provoked by auto-acknowledgment.

A UA shall perform the Change Auto-acknowledgment abstract operation by recording the value of the Auto-acknowledge-IPMs argument in the correspondingly named state variable. If that value is *true*, it also shall record the value of the Auto-acknowledge-suppl-receipt-info argument in the correspondingly named state variable.

18.3.3 *Change Auto-forwarding*

To assist it in providing this abstract operation, a UA maintains the following state variables:

a) **Auto-forward-IPMs**: A Boolean that indicates whether or not auto-forwarding is in effect.

b) **Auto-forward-recipients**: A Sequence of O/R names that identify the users and DLs to which IPMs are being auto-forwarded.

c) **Auto-forward-heading**: The Heading of each forwarding IPM provoked by auto-forwarding. Its Auto-forwarded field has the value *true*.

d) **Auto-forward-comment**: The Auto-forward Comment non-receipt field of each NRN conveyed to the originator of an auto-forwarded IPM.

A UA shall perform the Change Auto-forwarding abstract operation by recording the value of the Auto-forward-IPMs argument in the correspondingly named state variable. If that value is *true*, it also shall record the values of the Auto-forward-recipients, Auto-forward-heading, and Auto-forward-comment arguments in the correspondingly named state variables.

18.4 *Invocation of Reception operations*

A UA shall invoke the abstract operations available at its reception port as specified below. The UA alters none of its state variables in connection with its invocation of these operations.

The UA invokes these operations in response to the MTS invocation of the following abstract operations of the MTS Abstract Service (which, for the remainder of this clause, are unqualified as to their source):

a) Report Delivery;

b) Message Delivery.

Note – The abstract operations of a reception port report no errors.

18.4.1 *Receive Report*

Whenever the MTS invokes Report Delivery at a UA's delivery port, the UA shall invoke the Receive Report abstract operation with the following arguments:

a) *Envelope:* Report Delivery's Envelope argument.

b) *Undelivered-object:* Determined from Report Delivery's Returned-content argument as specified in 20.1.

Note – How the UA employs the Content-identifier component of Report Delivery's Envelope argument is a local matter.

18.4.2 *Receive IPM*

Whenever the MTS invokes Message Delivery at a UA's delivery port, and its Content argument encodes an IPM as specified in 20.1, the UA shall invoke the Receive IPM abstract operation with the following arguments, provided that the message is subject to neither auto-forwarding nor auto-discard (see 18.5):

a) *Envelope:* Message Delivery's Envelope argument.

b) *Content:* Determined from Message Delivery's Content argument as specified in 20.1 (but no longer marked as an IPM).

18.4.3 *Receive RN*

Whenever the MTS invokes Message Delivery at a UA's delivery port, and its Content argument encodes an RN as specified in 20.1, the UA shall invoke the Receive RN abstract operation with the following arguments:

a) *Envelope:* Message Delivery's Envelope argument.

b) *Content:* Determined from Message Delivery's Content argument as specified in 20.1 (but no longer marked as an RN).

18.4.4 *Receive NRN*

Whenever the MTS invokes Message Delivery at a UA's delivery port, and its Content argument encodes an NRN as specified in 20.1, the UA shall invoke the Receive NRN abstract operation with the following arguments:

a) *Envelope:* Message Delivery's Envelope argument.

b) *Content:* Determined from Message Delivery's Content argument as specified in 20.1 (but no longer marked as an NRN).

18.5 *Internal procedures*

A UA shall perform as specified below the internal procedures of auto-discard, -acknowledgment, and -forwarding in ultimate fulfilment of the abstract operations available at its management port.

The procedures involve the following abstract operations of the MTS Abstract Service (which, for the remainder of this subclause, are unqualified as to their source):

a) Message Submission;

b) Message Delivery.

As implied by the above, in the course of the procedures, the UA has occasion to invoke Message Submission. What it does with the results of this abstract operation is a local matter.

The UA shall consider as a candidate for each procedure individually every message for which all of the following conditions hold:

a) The MTS has conveyed the message to the UA by invoking Message Delivery at the UA's delivery port.

b) The UA has not conveyed the message to the user by invoking Receive IPM at the user's reception port.

c) The message contains an IPM (rather than an IPN).

Note – With reference to Item b above, the message might be detained in the UA, e.g. as might be typical, because of the user's unavailability.

18.5.1 *Auto-discard*

The UA shall subject to auto-discard each candidate message with respect to whose content either of the following conditions holds:

a) The Auto-discard-expired-IPMs state variable has the value *true* and the date and time denoted by the IPM's Expiry Time field have past.

b) The Auto-discard-obsolete-IPMs state variable has the value *true* and another candidate IPM identifies the present candidate IPM by means of its Obsoleted IPMs heading field.

The UA shall auto-discard each such message as follows.

18.5.1.1 *Discard of IPM*

The UA shall discard the IPM, so as to never convey it to the user.

18.5.1.2 *Construction of NRN*

The UA shall construct an NRN if, and only if, one is requested by means of the Notification-requests component of the IPM's subject recipient specifier.

The NRN shall have the common fields prescribed for auto-acknowledgment (see 18.5.2.1).

The NRN shall have the following receipt fields:

a) *Non-receipt Reason:* The value *ipm-discarded*.

b) *Discard Reason:* The *value ipm-expired* or *ipm-obsoleted*, whichever applies. If both apply, either value may be specified.

c) *Auto-forward Comment:* Omitted.

d) *Returned IPM:* If the IPM's return is requested by means of the Notification-requests component of its subject recipient specifier, and the Converted-encoded-information-types component of Message Delivery's Envelope argument is absent, the IPM. Omitted otherwise.

18.5.1.3 *Submission of NRN*

The UA shall submit the NRN (if any) above by invoking Message Submission. Its Envelope argument shall be as prescribed for auto-acknowledgment (see 18.5.2.2) except that notification-type should be set to type 2

Note – The notification-type indication may become mandatory in a future version of this Recommendation, its Content argument determined from the NRN as specified in 20.1.

18.5.2 *Auto-acknowledgment*

The UA shall subject to auto-acknowledgment each candidate message with respect to whose content the following condition holds:

– The Auto-acknowledgment state variable has the value *true* and the IPM requests an RN of the UA's user by means of the Notification-requests component of the IPM's subject recipient specifier.

The UA shall auto-acknowledge each such message as follows.

18.5.2.1 *Construction of RN*

The UA shall construct an RN.

The RN shall have the following common fields:

a) *Subject IPM:* The IPM's This IPM heading field.

b) *IPN Originator:* Specified or omitted as a local matter (but, of course, in accordance with 8.1.2).

c) *IPM Preferred Recipient:* The Recipient component of the IPM's subject recipient specifier, unless its Formal-name component is the O/R name of the UA's user, in which case the field shall be omitted.

d) *Conversion EITs:* The Converted-encoded-information-types component of Message Delivery's Envelope argument.

The RN shall have the following receipt fields:

1) *Receipt Time:* The current date and time.

2) *Acknowledgment Mode:* The value *automatic*.

3) *Suppl Receipt Info:* The Auto-acknowledge-suppl-receipt-info state variable.

18.5.2.2 *Submission of RN*

The UA shall submit the RN above by invoking Message Submission with the following arguments:

a) *Envelope:* The components of this argument shall be as prescribed for performance of the Originate RN abstract operation with the following exceptions:

   i) *PrioriASN.1 types:* As specified by Message Delivery's Envelope argument.

   ii) *Per-message-indicators:* A local matter, except that *conversion-prohibited* shall be among the values specified and except that the notification-type should be set to type 1

   Note – The notification-type indication may become mandatory in a future version of this Recommendation.

   iii) *Per-recipient-fields:* A single field whose Recipient-name component shall be the Originator-name component of Message Delivery's Envelope argument.

b) *Content:* Determined from the RN as specified in 20.1.

18.5.3 *Auto-forwarding*

The UA shall subject to auto-forwarding every candidate message, provided that the Auto-forward-IPMs state variable has the value *true*.

The UA shall auto-forward each such message as follows.

18.5.3.1 *Prevention of Loops*

The UA shall suppress auto-forwarding if, and only if, the IPM to be forwarded itself contains a forwarding IPM that the UA previously created. Auto-forwarding shall be suppressed whether the forwarding IPM appears (directly) in a Message body part of the IPM to be forwarded, or (nested) in a Message body part of the IPM that appears in such a body part.

The UA shall consider itself to have created the forwarding IPM above (whose Auto-forwarded heading field has the value *true*) if, and only if, the Originator-name component of the IPM's Parameters component matches the O/R name of the UA's user.

Note – Auto-forwarding an IPM of the kind described above would constitute an auto-forwarding "loop".

18.5.3.2 *Construction of IPM*

The UA shall construct a forwarding IPM whose Heading is the Auto-forward-heading state variable (its Auto-forwarded field having the value *true*) and whose Body contains a body part of type Message.

The Message body part shall have the following components:

a) *Parameters:* The Envelope argument and the Delivery Time argument of Message Delivery. See 7.3.8.

b) *Data:* The IPM to be forwarded.

18.5.3.3 *Submission of IPM*

The UA shall submit the IPM it constructed above by invoking Message Submission with the following arguments:

a) *Envelope:* The components of this argument shall be as follows:

   i) *Originator-name:* The O/R name of the UA's user.

   ii) *Content-type* and *Original-encoded-information-types:* Determined from the IPM as specified in 20.2 and 20.4.

   iii) *Content-identifier* and *Content-correlator:* Specified or omitted as a local matter.

iv) *Priority:* As specified by Message Delivery's Envelope argument.

v) *Per-message-indicators* and *Extensions:* A local matter.

vi) *Deferred-delivery-time:* Omitted.

vii) *Per-recipient-fields:* Their Recipient-name components shall be the O/R names that make up the Auto-forward-recipients state variable. Their other components are a local matter.

b) *Content:* Determined from the IPM as specified in 20.1.

18.5.3.4 *Construction of NRN*

The UA shall construct an NRN if, and only if, one is requested by means of the Notification-requests component of the forwarded IPM's subject recipient specifier.

The NRN shall have the common fields prescribed for the performance of auto-acknowledgment.

The NRN shall have the following receipt fields:

a) *Non-receipt Reason:* The value *ipm-auto-forwarded.*

b) *Discard Reason:* Omitted.

c) *Auto-forward Comment:* The Auto-forward-comment state variable.

d) *Returned IPM:* If the IPM's return is requested by means of the Notification-requests component of its subject recipient specifier, and the Converted-encoded-information-types component of Message Delivery's Envelope argument is absent, the IPM. Omitted otherwise.

18.5.3.5 *Submission of NRN*

The UA shall submit the NRN (if any) above by invoking Message Submission. Message Submission's Envelope argument shall be as prescribed for auto-acknowledgment except that notification-type should be set to type 2

Note – The notification-type indication may become mandatory in a future version of this Recommendation), its Content argument determined from the NRN as specified in 20.1.

19 Message Store operation

An MS must perform certain Interpersonal Messaging-specific functions to qualify as an IPMS MS and thus distinguish itself from a generic MS. These functions are the subject of the present clause.

19.1 *Creation of Information objects*

An IPMS MS shall satisfy the following requirements related to the information objects it maintains:

a) The MS shall maintain a separate information object for each (message containing an) IPM or IPN that is delivered to it.

b) The MS shall maintain as a separate information object not only each (message containing a) forwarding IPM [pursuant to item a)] but also each (message containing a) forwarded IPM (recursively).

c) The MS shall maintain as a separate information object the Returned IPM which may be present in an NRN.

19.1.1 *Creation of child-entries*

The following general-attribute-types shall be present in child-entries: content, content-length, content-type, creation-time, entry-status, entry-type, parent-sequence-number, sequence-number. The absence of a delivery envelope precludes the generation of other general attributes which are mandatory in Table 1 of CCITT Rec. X.413 | ISO/IEC 10021-5 for the following types of child-entry:

a) the Returned-content of a delivery report which contains an IPM or an NRN;

b) the Returned IPM optionally present in an NRN;

c) the Message body part (i.e. the forwarded IPM) of a forwarding IPM where the Parameters component of the body part is absent.

In the case where a child-entry is generated from a Message body part in which the Parameters component is present:

1) if Delivery-time is present in Parameters then the message-delivery-time general-attribute-type shall be present;

2) if Delivery-envelope is present in Parameters then all the other mandatory general-attribute-types defined for a delivered-message entry shall be present except for message-delivery-envelope and message-delivery-identifier which shall be absent.

All child-entries except those containing returned content shall be given an entry-type value of delivered-message.

19.2 *Maintenance of Attributes*

An IPMS MS shall satisfy the following requirements related to MS attributes:

a) For each IPM or IPN it holds, the MS shall support the attributes of Annex C as specified therein.

b) For each IPM it holds, the MS shall give the following meanings to the defined values of the MS-status attribute:

   i) *new:* No attribute values have been conveyed to the UA.

   ii) *listed:* At least one attribute value has been conveyed to the UA, and at least one body part has not been conveyed.

   iii) *processed:* All body parts (the body parts as single attributes, or the data component only from all body parts) have been conveyed to the UA.

c) For each IPN it holds, the MS shall give the following meanings to the defined values of the MS-status attribute:

   i) *new:* No attribute values have been conveyed to the UA.

   ii) *listed:* At least one attribute value has been conveyed to the UA, and at least one attribute other than Returned IPM has not been conveyed.

   iii) *processed:* All attributes, with the possible exception of Returned IPM, have been conveyed to the UA.

d) The MS-status attribute shall reflect the state of affairs prior to an abstract operation invocation that alters its value.

e) The Content-type attribute of each (message containing an) IPM or IPN that is delivered to the MS shall have the value id-mct-p2-1984 or id-mct-p2-1988 (see Annex D), as appropriate, depending upon the content type of the delivered message (see 20.2).

19.3 *Notification of Non-receipt*

When it discards an IPM while performing the Delete abstract operation of the MS Abstract Service, the MS shall submit a NRN if one is requested and the IPM's MS-status attribute has the value *listed*.

19.4 *Auto-forwarding*

An IPMS MS shall perform the Auto-forward action of CCITT Rec. X.413 | ISO/IEC 10021-5 as specified in 18.5.3. It makes use of the Other-parameters component of the Auto-forward-registration argument of the Register MS abstract operation of the MS Abstract Service. The data type of the Other-parameters component is defined as follows:

```
ForwardedInfo ::= SET {
        auto-forwarding-comment  [0]  AutoForwardComment OPTIONAL,
        cover-note  [1]  IA5TextBodyPart OPTIONAL,
        this-ipm-prefix  [2]  PrintableString (SIZE
              (1..ub-ipm-identifier-suffix)) OPTIONAL }
```

In addition, the MS shall satisfy the following requirements:

a) Submit an NRN even if it retains a copy of the forwarded IPM.

b) Draw the NRN's Auto-forward Comment field, if any, from the Other-parameters component.

c) Draw the cover-note, if any, to be included with the forwarded IPM, from the Other-parameters component.

d) Prefix the User-relative-identifier component of the This IPM field of the forwarding IPM's heading with, if present, the This-ipm-prefix.

Note – An (IPMS) MS performs neither auto-discard nor auto-acknowledgment, except possibly as a local matter.

19.5 *Manual Forwarding*

An IPMS MS shall support the manual forwarding of a message using the forwarding-request extension of CCITT Rec. X.413 | ISO/IEC 10021-5, as specified in 6.6. The IPMS MS user may submit an IPM, including Heading and Body, using the MessageSubmission Operation, and identify by using the forwarding-request extension a message that is already in the MS which is to be combined with the submitted message body for forwarding to the message's recipient(s).

The submitted message body and the forwarded message are then combined by inserting the forwarded message as a Message body part into the submitted message body. The Message body part becomes the last body part of the submitted message body.

20 Message Contents

As has already been seen, various secondary objects (e.g. UAs) have occasion to convey the information objects of Section 2 as the contents of messages, as well as to convey probes concerning such messages. This clause specifies precisely how they shall do this.

The rules governing the transmittal of such messages and probes, and the semantics and abstract and transfer syntaxes of their contents, are called the **Interpersonal Messaging Protocol** (**P2**).

Note – The name, "P2", reflects the historical fact that this was the second Message Handling protocol to be developed.

20.1 *Content*

A secondary object that submits a message containing an IPM or IPN shall supply as the octets of the Octet String that constitutes the content of the message the result of encoding the InformationObject of Section 2 in accordance with the Basic Encoding Rules of CCITT Rec. X.209 | ISO/IEC 8825.

20.2 *Content type*

A secondary object that submits a message containing an IPM or IPN shall select its content type as follows.

If the IPM or IPN satisfies all of the following constraints, the Integer 2 shall be specified:

i) The Heading and the recipient specifier (of an IPM), or the common fields, non-receipt fields, receipt fields, and the other notification type fields (of an IPN), lack extension fields.

ii) The Body (of an IPM) lacks Externally Defined body parts.

iii) The Parameters element of any Videotex body part (of an IPM) lacks the Syntax member.

iv) Every component of the IPM or IPN that is a value of a data type defined as part of the MTS Abstract Service meets the constraints of Recommendation X.411 (1984).

The types in question are those listed in the IMPORTS clause of the ASN.1 module defined in Annex E. The constraints in question are detailed in an annex of CCITT Rec. X.419 | ISO/IEC 10021-6.

v) The Data element of any Message body part (of an IPM) satisfies these same constraints (recursively).

Otherwise, the Integer 22 shall be specified.

Note 1 – The message content protocol (here) denoted by the Integer 2 is identical to that specified by Recommendation X.420 (1984) (as clarified by Version 6 of the *X.400-Series Implementor's Guide*), except that the Simple Formattable Document body part type, defined in the latter, is omitted from the former.

Note 2 – The Integer 2 is favoured, above, over the Integer 22 to foster interworking between systems conforming to this Recommendation and systems conforming (only) to Recommendation X.420 (1984).

Note 3 – The MTS does not convert between message content protocols. Thus it does not convert between P2 as defined by this Recommendation alone (and denoted by the Integer 22) and P2 as defined by both this Recommendation and Recommendation X.420 (1984) (and denoted by the Integer 2).

20.3 *Content length*

A secondary object that submits a probe concerning a message containing an IPM or IPN shall specify as the length of the message's content the size in octets of the encoding of the instance in question of the InformationObject of Section 2 (a choice of an IPM or an IPN) when the Basic Encoding Rules of CCITT Rec. X.209 | ISO/IEC 8825 are followed. If those rules permit several (e.g. both primitive and constructed) encodings of that InformationObject, the content length may reflect any one of them.

20.4 *Encoded Information types*

A secondary object that submits a message containing an IPM or IPN shall specify the basic encoded information types (EITs) and non-basic parameters (NBPs) of the message as follows.

In the case of an IPN, the basic EITs shall be *unspecified*.

In the case of an IPM, the basic EITs and NBPs shall be specified in accordance with the following rules:

a) *Multiple body parts:* The basic EITs (if any) and NBPs (if any) of the message shall comprise the logical union of the basic EITs and NBPs of the IPM's individual body parts, respectively.

b) *(Forwarded) Message body part:* The basic EITs (if any) and NBPs (if any) of a Message body part shall be those of the forwarded message.

c) *Externally Defined body part:* An Externally Defined body part whose extended type corresponds to a basic type (see Annex B) shall be treated in the manner prescribed for the basic type.

Any other extended body part type shall be handled as follows. If there corresponds to the type one or more externally defined EITs, they shall be specified. Otherwise, the *undefined* EIT shall be indicated. In either case, no NBPs shall be specified.

d) *Basic body part:* The basic EITs (if any) and NBPs (if any) of an individual body part of type other than Message and Externally Defined shall depend upon that body part type as specified in Table 2/X.420. A body part type for which the table specifies no basic EITs shall result in the setting of no bits in the basic EITs Bit String.

e) *Encrypted body part:* The effect of an Encrypted body part upon the basic EITs and NBPs to be specified is for further study.

TABLE 2/X.420

Interpersonal Messaging Basic EITs and NBPs

Body Part Type	Basic EIT	NBPs
IA5 Text	IA5 Text	–
G3 Facsimile	G3 Facsimile	G3 Facsimile
G4 Class 1	G4 Class 1	G4 Class 1/ Mixed-mode
Teletex	Teletex	Teletex
Videotex	Videotex	–
Encrypted	–	–
Message	(see Text)	(see Text)
Mixed-mode	Mixed-mode	G4 Class 1/Mixed-mode
Bilaterally Defined	Undefined	–
Nationally Defined	Undefined	–
Externally Defined	(see text)	(see text)

21 Port Realization

How an MS or the MTS concretely realizes the secondary ports it supplies is specified in CCITT Rec. X.419 | ISO/IEC 10021-6.

How a UA, TLMA, or AU concretely realizes the primary ports it supplies is beyond the scope of this Recommendation.

Note 1 – A UA's user interface is a local matter. A wide variety of interfaces involving, e.g. a wide variety of input/output devices are possible.

Note 2 – A TLMA's realization of its primary ports is specified in part by Recommendation T.330.

Note 3 – An AU provides its primary ports by means of the particular communication system to which that AU provides access.

22 Conformance

The requirements a secondary object (excluding the MTS) and its implementor shall meet when the latter claims the former's conformance to this Recommendation are identified below. A number of the conformance requirements distinguish between *support upon origination* and *support upon reception.*

22.1 *Origination Versus Reception*

A UA, TLMA, or AU shall be said to **support upon origination** a particular heading field, heading extension, basic body part type, or extended body part type if, and only if, it accepts, preserves, and emits, in full accord with this Recommendation, that particular heading field or extension, or body parts of that particular basic or extended type, whenever a user calls upon it to convey an IPM containing them to the MTS or the user's MS (the latter only in the case of a UA).

A UA, TLMA, or AU shall be said to **support upon reception** a particular heading field, heading extension, basic body part type, or extended body part type if, and only if, it accepts, preserves, and emits, in full accord with this Recommendation, that particular heading field or extension, or body parts of that particular basic or extended type, whenever the MTS or a user's MS (the latter only in the case of a UA) calls upon it to convey to the user an IPM containing them.

Note – In point of fact, a PDAU supports nothing upon origination because it is not a supplier of the origination port.

22.2 *Statement Requirements*

The implementor of an IPMS UA, IPMS MS, TLMA, or AU shall state the following. For each item below he shall make separate statements concerning conformance upon origination and conformance upon reception:

a) The heading fields and heading extensions for which he claims conformance.

b) The basic and extended body part types for which he claims conformance.

c) In the case of an IPMS MS, or an IPMS UA accessing an IPMS MS, the Interpersonal Messaging-specific MS attribute-types for which it claims conformance.

d) In the case of an IPMS MS, or an IPMS UA accessing an IPMS MS, any claims of conformance for the auto-forward auto-action defined in CCITT Rec. X.413 | ISO/IEC 10021-5.

In addition, the implementor of a TLMA or AU shall state whether he is claiming conformance for import or export or both.

22.3 *Static requirements*

An IPMS UA, IPMS MS, TLMA, or AU shall satisfy the following static requirements:

a) An IPMS UA, IPMS MS, TLMA, or AU shall implement the heading fields and heading extensions, and the basic and extended body part types for which conformance is claimed.

b) An IPMS MS, or an IPMS UA accessing an IPMS MS, shall support the Interpersonal Messaging-specific MS attribute-types for which conformance is claimed, but including as a minimum those designated mandatory in Annex C.

c) An IPMS UA, IPMS MS, TLMA, or AU shall concretely realize its abstract ports as specified in clause 21.

d) An IPMS UA or IPMS MS shall be able to both submit and accept delivery of messages of both of the content types of 20.2.

e) An IPMS MS, or an IPMS UA accessing an IPMS MS, shall conform to at least one of the MS Access Protocols specified in CCITT Rec. X.419 | ISO/IEC 10021-6.

f) An IPMS MS, or an IPMS UA accessing an IPMS MS, shall support the auto-forward auto-action defined in CCITT Rec. X.413 | ISO/IEC 10021-5 if conformance to it is claimed.

g) A TLMA or AU shall be able to import and/or export such messages, according to the conformance claimed.

22.4 *Dynamic Requirements*

An IPMS UA, IPMS MS, TLMA, or AU shall satisfy the following dynamic requirements:

a) An IPMS UA or IPMS MS shall follow the rules of operation specified in clauses 18 or 19, respectively.

b) An IPMS UA, IPMS MS, TLMA, or AU shall submit and accept delivery of messages whose contents are as specified in clause 20.

c) An IPMS UA, IPMS MS, TLMA, or AU shall register with the MTS its ability to accept delivery of messages of both of the content types of 20.2.

ANNEX A

(to Recommendation X.420)

Heading extensions

(This annex forms an integral part of this Recommendation)

This annex defines all (presently defined) heading extensions.

A.1 *Incomplete Copy*

The **Incomplete Copy** heading extension, by its presence, indicates that one of more body parts or heading fields are absent from the Body of (the present instance of) the IPM. The extension comprises a Null (by default).

```
incomplete-copy IPMS-EXTENSION
        ::= id-hex-incomplete-copy
```

If this extension is absent from the Extensions heading field, all body parts shall be considered present.

A.2 *Languages*

The **Languages** heading extension identifies the languages used in the composition of the IPM's Subject heading field and Body. The extension comprises a Set of zero or more Printable Strings, each one of the two-character language codes identified by ISO 639. The two-character language code may optionally be followed by a space and a two-character ISO 3166 country code (see ISO 639 4.4) if it is necessary to identify a specific national usage of the language (e.g., "en" identifies the English language, "en GB" identifies English as used in the UK, and "en US" identifies English as used in the USA).

```
languages IPMS-EXTENSION
        VALUE SET OF Language
        ::= id-hex-languages

Language  ::= PrintableString (SIZE (2 . . 5) )
```

If this extension is absent from the Extensions heading field or no languages are indicated, the languages shall be considered unspecified.

A.3 *Auto-submitted*

The **Auto-submitted** heading extension indicates whether the IPM was submitted without human intervention, and if so whether the message was auto-generated, auto-replied, or auto-forwarded.

If this heading extension has the value not-auto-submitted, the message-submission is under direct or indirect control of a human.

```
auto-submitted IPMS-EXTENSION
        VALUE AutoSubmitted
        ::= id-hex-auto-submitted

AutoSubmitted ::= ENUMERATED {
        not-auto-submitted (0),
        auto-generated (1),
        auto-replied (2),
        auto-forwarded (3) }
```

The absence of this heading extension indicates that no information is available as to whether the message-submission involved human control.

ANNEX B

(to Recommendation X.420)

Extended body part types

(This annex forms an integral part of this Recommendation)

B.1 *Equivalents of basic body part types*

For each basic body part type, this Recommendation defines as follows an equivalent extended body part type.

```
ia5-text-body-part      EXTENDED-BODY-PART-TYPE
        PARAMETERS  IA5TextParameters  IDENTIFIED BY id-ep-ia5-text
        DATA            IA5TextData
        ::= id-et-ia5-text

g3-facsimile-body-part EXTENDED-BODY-PART-TYPE
        PARAMETERS  G3FacsimileParameters IDENTIFIED BY id-ep-g3-facsimile
        DATA            G3FacsimileData
        ::= id-et-g3-facsimile

g4-class1-body-part EXTENDED-BODY-PART-TYPE
        DATA    G4Class1BodyPart
        ::= id-et-g4-class1

teletex-body-part       EXTENDED-BODY-PART-TYPE
        PARAMETERS  TeletexParameters IDENTIFIED BY id-ep-teletex
        DATA            TeletexData
        ::= id-et-teletex

videotex-body-part      EXTENDED-BODY-PART-TYPE
        PARAMETERS  VideotexParameters IDENTIFIED BY id-ep-videotex
        DATA            VideotexData
        ::= id-et-videotex

encrypted-body-part EXTENDED-BODY-PART-TYPE
        PARAMETERS  EncryptedParameters IDENTIFIED BY id-ep-encrypted
        DATA            EncryptedData
        ::= id-et-encrypted

message-body-part       EXTENDED-BODY-PART-TYPE
        PARAMETERS  MessageParameters IDENTIFIED BY id-ep-message
        DATA            MessageData
        ::= id-et-message

mixed-mode-body-part  EXTENDED-BODY-PART-TYPE
        DATA  MixedModeBodyPart
        ::= id-et-mixed-mode

bilaterally-defined-body-part EXTENDED-BODY-PART-TYPE
        DATA  BilaterallyDefinedBodyPart
        ::= id-et-bilaterally-defined

nationally-defined-body-part EXTENDED-BODY-PART-TYPE
        DATA NationallyDefinedBodyPart
        ::= id-et-nationally-defined
```

B.2 *General Text*

A **General Text** extended body part represents character text of a general nature. It has Parameters and Data components.

```
general-text-body-part EXTENDED-BODY-PART-TYPE
    PARAMETERS    GeneralTextParameters IDENTIFIED BY id-ep-general-text
    DATA          GeneralTextData
    ::= id-et-general-text

GeneralTextParameters ::= SET OF CharacterSetRegistration
GeneralTextData ::= GeneralString
```

The Parameters component comprises a list of the character set registrations that are or may be present in the Data component. Each character set is represented by the registration number defined in the registration of that character set, registered in accordance with ISO 2375.

The implicit character sets (registration numbers 2 and 1) specified by the ASN.1 Basic Encoding Rules should be present in the Parameters component if they are used.

```
CharacterSetRegistration ::= INTEGER (1 . . 32767)
```

The Data component comprises a single General String. Character set designators other than those for character sets defined in the Parameters component shall not be used.

Each General String shall be encoded using 8-bit encoding (not 7-bit).

Within the Data component, lines may be of any length. Whenever the component is rendered (e.g. displayed to or printed for a user), all (rather than only a part) of the text must be communicated (e.g. lines may be folded but shall not be truncated).

For this extended body part type, externally defined EITs are defined [pursuant to item c) of 20.4] as follows. One EIT is used for each character set the Parameters component has explicitly identified. It is denoted by the Object Identifier assigned to that character set.

This annex acts as the registration authority for such Object Identifiers, as follows. All the Object Identifiers are allocated as leaves immediately under the single vertex representing this registration authority (id-cs-eit-authority). The Object Identifier component identifying the character set represented by the leaf is the registration number of that character set as allocated in accordance with ISO 2375.

Example – The externally defined EITs for Latin Alphabet number 1 (ISO 8859-1) are {id-cs-eit-authority 6} for the G0 set and {id-cs-eit-authority 100} for the G1 set.

Note 1 – 1 It is preferred that the list of characters sets within the Parameters component includes only the registration numbers of those character sets which are actually used within the Data component.

Note 2 – The ASN.1 Basic Encoding Rules (ISO/IEC 8825) provide character set 2 and character set 1 as initially designated and invoked into G0 and C0 respectively. These rules require that any other character set designators are inserted within the encoding of the General String. The G sets then have to be invoked using locking shift or single shift control functions.

Note 3 – The registration numbers and the associated escape sequences for the character set designators are defined in the *ISO International Register of Coded Character Sets To Be Used With Escape Sequences*. This is the register established in accordance with ISO 2375.

B.3 *File Transfer*

A **File Transfer** body part represents an information object used to convey the contents, and optionally the attributes, of a stored file. The file transfer body part is based on the file model defined in ISO 8571-2 (FTAM). It has Parameters and Data components.

```
file-transfer-body-part EXTENDED-BODY-PART-TYPE
        PARAMETERS    FileTransferParameters IDENTIFIED BY id-ep-file-transfer
        DATA          FileTransferData
        ::= id-et-file-transfer

FileTransferParameters ::= SEQUENCE {
        related-stored-file     [0]  RelatedStoredFile OPTIONAL,
        contents-type           [1]  ContentsTypeParameter DEFAULT { iso standard 8571
                                                        document-type   (5)
                                                                unstructured-binary    (3) },
        environment             [2]  EnvironmentParameter OPTIONAL,
        compression             [3]  CompressionParameter OPTIONAL,
        file-attributes         [4]  FileAttributes OPTIONAL,
        extensions              [5]  ExtensionsField DEFAULT { } }

FileTransferData  ::= SEQUENCE OF EXTERNAL
-- This conveys a sequence of data values representing file contents.
-- The rules for generating this sequence are implied by the value of the contents-type parameter
```

B.3.1 *Related stored file parameter*

This parameter indicates to the recipient any intended relationship between the file in this body part and any file(s) held by the recipient. Stored files may be identified either by pathname or by reference to previous MHS messages sent. Explicit relationships with stored files that may be indicated include:

– unspecified;

– a new file may be created using the contents in this body part;

– the contents of an existing file may be replaced by the contents in this body part;

– an existing file may be extended using the contents of this body part.

The syntax for this parameter is:

```
RelatedStoredFile := SET OF SEQUENCE {
        file-identifier         FileIdentifier,
        relationship            Relationship DEFAULT unspecified }

FileIdentifier ::= CHOICE {
        pathname-and-version     [0]  PathnameandVersion,
        cross-reference          [1]  CrossReference }

PathnameandVersion ::= SEQUENCE {
        pathname         [0]  Pathname-Attribute,

Pathname-Attribute ::= CHOICE {
        incomplete-pathname        [0]  Pathname,
        complete-pathname         [23]  Pathname }
        file-version               [1]  GraphicString OPTIONAL }

CrossReference ::= SEQUENCE {
        application-cross-reference     [0]  OCTET STRING,
        message-reference               [1]  MessageReference OPTIONAL,
        body-part-reference             [2]  INTEGER OPTIONAL }

MessageReference ::= SET {
        user                        [0]  ORName,
        -- Defined in 8.5.5 of CCITT Rec. X.411 | ISO/IEC 10021-4
        user-relative-identifier    [1]  PrintableString }

Relationship ::= CHOICE {
        explicit-relationship       [0]  ExplicitRelationship,
        descriptive-relationship    [1]  GraphicString }

ExplicitRelationship ::= ENUMERATED {
        unspecified   (0),
        new-file      (1),
        replacement   (2),
        extension     (3) }
```

The pathname option is intended for use in a manner consistent with ISO 8571-2, as amended by Amendment 1. It is a sequence of elements, each of which represents a name component. When more than one element is encoded, the first element shall be the file name and the remaining elements shall be concatenated to represent the file name prefix.

Note – ISO 8571-2 Amendment 1 renamed the "filename" attribute in ISO 8571-2 to the "pathname" attribute.

A message reference has the following components:

a) User: Identifies the user who originated the referenced message. One of the user's OR names.

b) User-relative-identifier: Unambiguously identifies a message, distinguishing it from all other messages that the user who is identified by the User component originates. A Printable String of from zero to a prescribed number of characters. A length of zero is discouraged.

A body part reference uniquely identifies a body part within a message. It is for use when referencing a message with a content type which includes body part references.

B.3.2 *Contents type parameter*

The contents type parameter indicates the abstract data types of the contents of the file and the structuring information which is necessary if the complete file structure and semantics are to be maintained during the transfer of the file.

```
ContentsTypeParameter ::= Contents-Type-Attribute

Contents-Type-Attribute ::= CHOICE {
        document-type     [0]   SEQUENCE {
                document-type-name          Document-Type-Name,
                Parameter                   [0]   ANY OPTIONAL }
            -- The actual types to be used for values of the parameter field
            -- are defined in the named document type
        constraint-set-and-abstract-syntax      [1]   SEQUENCE {
                constraint-set-name         Constraint-Set-Name,
                abstract-syntax-name        Abstract-Syntax-Name } }

Constraint-Set-Name ::= OBJECT IDENTIFIER

Document-Type-Name ::= OBJECT IDENTIFIER

Abstract-Syntax-Name ::= OBJECT IDENTIFIER
```

The value is either a document-type name (possibly with parameters in a single value of any type) or a pair of abstract syntax name and constraint set name. Each of these names are values of the type OBJECT IDENTIFIER.

The concepts of document-type and constraint set are described fully in ISO 8571-1 and ISO 8571-2. Examples of document types which may be used in this body part are:

a) unstructured text file (FTAM-1);

b) unstructured binary file (FTAM-3);

c) sequential binary file (FTAM-4).

B.3.3 *Environment Parameter*

The environment parameter describes the environment (e.g. machine, operating system, and application) from which the file originated. It has the following syntax:

```
EnvironmentParameter ::= SEQUENCE {
    application-reference       [0]  GeneralIdentifier OPTIONAL,
    machine                     [1]  GeneralIdentifier OPTIONAL,
    operating-system            [2]  OBJECT IDENTIFIER OPTIONAL,
    user-visible-string         [3]  SEQUENCE OF GraphicString OPTIONAL }

GeneralIdentifier ::= CHOICE {
    registered-identifier       [0]  OBJECT IDENTIFIER,
    descriptive-identifier      [1]  SEQUENCE  OF GraphicString }
```

The application-reference field is intended to be used for identifying application programs and versions. The machine field is intended to be used for executable code modules to indicate hardware platforms. The operating-system field is intended to be used to identify the operating system of the processor from which the file originated.

B.3.4 *Compression parameter*

The compression parameter describes the compression type if the file is transferred in a compressed mode.

```
CompressionParameter ::= SEQUENCE {
    compression-algorithm-id        [0]  OBJECT IDENTIFIER,
    compression-algorithm-param     [1]  ANY DEFINED BY compression-algorithm-id }
```

B.3.5 *File attributes parameter*

The file attributes parameter conveys values of any of a set of optional file attributes. When the recipient is to create a new file, these values are to be used in establishing the initial file attributes.

Note – Transfer of an attribute value to a recipient should be interpreted as a request only; no particular recipient behaviour is guaranteed as a result.

The file attributes are technically aligned with ISO 8571-2 (FTAM). The semantic descriptions of these attributes in ISO 8571-2 take precedence over the abbreviated descriptions given below. The file attributes which can be conveyed in this parameter are:

- pathname
- permitted actions
- storage account
- date and time of creation
- date and time of last modification
- date and time of last read access
- identity of creator
- identity of last modifier
- identify of last reader
- object size
- future object size
- access control
- legal qualifications
- private use
- attribute-extensions

The syntax for the file attributes parameter is as follows:

```
FileAttributes ::= SEQUENCE {
    pathname                              Pathname-Attribute OPTIONAL,
    permitted-actions                [1]  Permitted-Actions-Attribute OPTIONAL,
    storage-account                  [3]  Account-Attribute OPTIONAL,
    date-and-time-of-creation        [4]  Date-and-Time-Attribute OPTIONAL,
    date-and-time-of-last-modification [5] Date-and-Time-Attribute OPTIONAL,
    date-and-time-of-last-read-access [6] Date-and-Time-Attribute OPTIONAL,
    identity-of-creator              [8]  User-Identity-Attribute OPTIONAL,
    identity-of-last-modifier        [9]  User-Identity-Attribute OPTIONAL,
    identity-of-last-reader          [10] User-Identity-Attribute OPTIONAL,
    object-size                      [13] Object-Size-Attribute OPTIONAL,
    future-object-size               [14] Object-Size-Attribute OPTIONAL,
    access-control                   [15] Access-Control-Attribute OPTIONAL,
    legal-qualifications             [16] Legal-Qualification-Attribute OPTIONAL,
    private-use                      [17] Private-Use-Attribute OPTIONAL,
    attribute-extensions             [22] Attribute-Extensions OPTIONAL }
```

The types of all the above components are imported from ISO 8571-4.

B.3.5.1 *Pathname attribute*

The pathname attribute provides a file name.

B.3.5.2 *Permitted actions attribute*

The permitted actions attribute indicates the set of actions that can be performed on the file.

B.3.5.3 *Storage account attribute*

The storage account attribute identifies the accountable authority responsible for accumulated file storage charges.

```
Account-Attribute ::= CHOICE {
    no-value-available      [0]   NULL,
      -- Indicates partial support of this attribute
    actual-values                 Account }

Account  ::= GraphicsString
```

B.3.5.4 *Date and Time attributes*

The date and time of creation attribute indicates when the file was created.

The date and time of last modification attribute indicates when the contents of the file were last modified.

The date and time of last read access attribute indicates when the contents of the file were last read.

B.3.5.5 *Identity attributes*

The identity of creator, identity of last modifier, and identity of last reader attributes identify the user(s) who created, last modified, and last read the file.

```
Use-Identity-Attribute ::= CHOICE {
    no-value-available      [0]   NULL,
      -- Indicates partial support of this attribute,
    actual-values                 User-Identity }

User-Identity ::= GraphicString
```

B.3.5.6 *Object size attributes*

The object size attribute is set to the nominal size in octets of the complete file.

The future object size attribute indicates the nominal size in octets to which the file may grow as a result of modification and extension.

B.3.5.7 *Access control attribute*

The access control attribute defines conditions under which access to the file is valid.

```
Access-control-Attribute ::= CHOICE {
        no-value-available      [0]   NULL,
          -- Indicates partial support of this attribute,
        actual-values           [1]   SET OF Access-Control-Element }
          -- The semantics of this attribute are described in ISO 8571-2

Access-Control-Element ::= SEQUENCE {
        action-list             [0]   Access-Request,
        concurrency-access      [1]   Concurrency-Access OPTIONAL,
        identity                [2]   User-Identity OPTIONAL
        passwords               [3]   Access-Passwords OPTIONAL,
        location                [4]   Application-Entity-Title OPTIONAL }

Access-Request ::= BIT STRING {
        read                    (0),
        insert                  (1),
        replace                 (2),
        extend                  (3),
        erase                   (4),
        read-attribute          (5),
        change-attribute        (6),
        delete-object           (7) }

Access-Passwords ::= SEQUENCE {
        read-password           (0)   Password,
        insert-password         (1)   Password,
        replace-password        (2)   Password,
        extend-password         (3)   Password,
        erase-password          (4)   Password,
        read-attribute-password (5)   Password,
        change-attribute-password (6) Password,
        delete-password         (7)   Password,
        pass-passwords          (8)   Pass-Passwords,
        link-password           (9)   Password }

Application-Entity-Title  ::= SEQUENCE {
        ap-title        ANY,
        ae-qualifier    ANY }

Password ::= CHOICE { GraphicString, OCTET STRING }

Pass-Passwords ::= SEQUENCE OF Password
```

B.3.5.8 *Legal qualifications attribute*

The legal qualifications attribute conveys information about the legal status of the file and its use.

B.3.5.9 *Private use attribute*

The meaning of the private use attribute is not defined.

B.3.5.10 *Attribute extensions*

The attribute extensions attribute allows for the inclusion of additional attributes in a manner consistent with Amendment 1 of ISO 8571-2 and ISO 8571-4.

B.3.6 *Extensions parameter*

The extensions parameter conveys information accommodated by no other parameter of the file transfer body part. The syntax and usage of this field is the same as that for the IPM content type extensions field as specified in 7.2.17.

B.3.7 *File Transfer body part data*

The data component of the body part contains the file contents being transferred.

The syntax for representation of these contents is implied by the contents-type parameter. When this parameter specifies a document type, the corresponding document type definition describes how to construct a sequence of presentation data values to convey the contents, and identifies the necessary abstract syntax(es). When the contents-type parameter specifies a constraint set and abstract syntax, the contents comprise a sequence of one or more data values from the identified abstract syntax.

The encoding is to be based on the transfer syntax specified as part of the document type definition, if any, or on the ASN.1 basic encoding rules otherwise.

B.3.8 *Encoded information type*

For this extended body part type, an externally defined EIT is defined [persuant to item c) of clause 20.4] by the Object Identifier id-eit-file-transfer.

Additionally, for each of the parameter components Contents Type, application-reference, machine, operating system and Compression Parameter, EITs may be used under the conditions described below:

If the Contents Type parameter is encoded as document type or specified by default, one EIT may be used to describe it and its value is the Object Identifier assigned to that document type. If the Contents Type parameter is encoded as constraint-set-and-abstract-syntax, two EITs may be used to describe the constraint-set-name and, respectively, the abstract-syntax-name, and their values are the corresponding Object Identifiers.

If the parameter components application-reference, machine, operating system and Compression Parameter are present and encoded as object identifiers, one EIT may be used to describe each of them and its value is the corresponding Object Identifier.

B.4 *Voice*

A **Voice** body part represents speech. It has Parameters and Data components.

Note – The voice body part defined here replaces the basic voice body part specified in Recommendation X.420 (1988).

Each instance of this body part in an IPM carries a single voice encoded message. The *voice-encoding-type* is specified in the voice parameters component.

```
voice-body-part EXTENDED-BODY-PART-TYPE
    PARAMETERS    VoiceParameters IDENTIFIED BY id-ep-vmg-body-part
    DATA          VoiceData
    ::= id-et-vmg-body-part
```

The syntax of the Parameters component is defined in Recommendation X.440. It contains the following information:

a) **Voice-message-duration** (O): the duration of the voice encoded data component in seconds.

b) **Voice-encoding-type** (D *32K ADPCM*): an object identifier which identifies the voice encoding applied to the data portion of this body part. The absence of this parameter implies that the voice encoding is per Recommendation G.721:(1988), i.e. 32K ADPCM.

c) **Other-parameters** (O): encoded as VMSsupplementaryInformation. It conveys additional information necessary for processing the voice encoded data component of this body part.

d) **Extension-parameters** (O): conveys parameter extensions needed for processing the voice encoded component of this body part.

Note – There are no extension-parameters defined in this Recommendation.

The Data component is the digital encoding of the voice. It is defined in Recommendation X.440 to be an ASN.1 type ANY that is defined by the value of the voice-encoding-type.

The reference definition of the encoded voice object used is that used by Recommendation.G.721 – 32K ADPCM. The object identifier used in this field is defined in Annex A of Recommendation X.440 (1992). By employing the use of object identifiers, this Recommendation does not draw a distinction between voice, or audio or music quality encoded objects.

For this extended body part type, externally defined EITs are defined [persuant to item c) of 20.4] by use of the same Object Identifier which identifies the **voice-encoding-type.**

ANNEX C

(to Recommendation X.420)

Message store attributes

(This annex forms an integral part of this Recommendation)

As described in CCITT Rec. X.413 | ISO/IEC 10021-5, an MS maintains and provides access to certain attributes (e.g. the importance) of each information object it holds. An attribute comprises a type and, depending upon the type, one or more values. Attributes that may assume several values simultaneously (all pertaining to one object) are termed multi-valued, those that may assume just one value, single-valued. Some attributes pertain to information objects of all kinds, others only to those of certain kinds (e.g. those of Section 2).

This annex defines the MS attributes specific to Interpersonal Messaging.

All of the attributes defined in this annex, except those corresponding to extended body part types (which cannot be enumerated; see C.3.6), are listed alphabetically, for reference, in the first column of Table C.1/X.420. This table records their presence in a delivered IPM, NRN, and RN. Where a delivery report contains returned content, the child-entry so created shall possess the attributes indicated for an IPM, NRN, or RN as appropriate. Where a delivered NRN contains a returned IPM, the child-entry so created shall possess the attributes indicated for an IPM. Where an IPM (whether delivered, in the returned content of a delivery report, or returned in a delivered NRN) contains a Message body part, the child-entry so created shall possess the attributes indicated for an IPM. See CCITT Rec. X.413 | ISO/IEC 10021-5 for an elaboration of the table's legend.

The example in Table C-2/X.420 illustrates the use of child entries in the stored-messages information-base. This Table shows four sets of entries corresponding respectively to a delivered IPM, a delivered RN, a delivered NRN, and a delivered report concerning a previously submitted IPM.

C.1 *Summary attributes*

Some attributes summarize an Interpersonal Messaging information object. These attributes are defined and described below.

TABLE C-1/X.420 (sheet 1 of 3)

Summary of MS attributes

Attribute	V	L	P			L	S
			IPM	NRN	RN		
A							
Acknowledgment mode	S	O	–	–	P	Y	Y
Authorizing users	M	O	C	–	–	Y	N
Auto-forward comment	S	O	–	C	–	Y	N
Auto-forwarded	S	O	P	–	–	Y	Y
Auto-submitted	S	O	C	–	–	Y	N
B							
Bilaterally Defined Body Parts	M	O	C	–	–	N	N
Blind Copy Recipients	M	O	C	–	–	Y	N
Body	S	M	P	–	–	N	N
C							
Conversion EITs	M	O	–	C	C	Y	N
Copy Recipients	M	O	C	–	–	Y	N
D							
Discard Reason	S	O	–	C	–	Y	Y
E							
Encrypted Body Parts	M	O	C	–	–	N	N
Encrypted Data	M	O	C	–	–	N	N
Encrypted Parameters	M	O	C	–	–	N	N
Expiry Time	S	O	C	–	–	Y	N
Extended Body Part Types	M	O	C	–	–	Y	Y
G							
G3 Facsimile Body Parts	M	O	C	–	–	N	N
G3 Facsimile Data	M	O	C	–	–	N	N
G3 Facsimile Parameters	M	O	C	–	–	N	N
G4 Class 1 Body Parts	M	O	C	–	–	N	N
H							
Heading	S	M	P	–	–	N	N

TABLE C-1/X.420 (sheet 2 of 3)

Attribute	V	L	P			L	S
			IPM	NRN	RN		
I							
IA5 Text Body Parts	M	O	C	–	–	N	N
IA5 Text Data	M	O	C	–	–	N	N
IA5 Text Parameters	M	O	C	–	–	N	N
Importance	S	O	P	–	–	Y	Y
Incomplete Copy	S	O	C	–	–	Y	N
IPM Entry Type	S	M	P	P	P	Y	Y
IPM Preferred Recipient	S	O	–	C	C	Y	N
IPM Synopsis	S	O	P	–	–	N	N
IPN Originator	S	O	–	C	C	Y	N
L							
Languages	M	O	C	–	–	Y	N
M							
Message Body Parts	M	O	C	–	–	N	N
Message Data	M	O	C	–	–	N	N
Message Parameters	M	O	C	–	–	N	N
Mixed-mode Body Parts	M	O	C	–	–	N	N
N							
Nationally Defined Body Parts	M	O	C	–	–	N	N
Non-receipt Reason	S	O	–	P	–	Y	Y
NRN Requestors	M	O	C	–	–	Y	N
O							
Obsoleted IMPs	M	O	C	–	–	Y	N
Originator	S	O	C	–	–	Y	N
P							
Primary Recipients	M	O	C	–	–	Y	N

TABLE C-1/X.420 (sheet 3 de 3)

Attribute	V	L	P			L	S
			IPM	NRN	RN		
R							
Receipt Time	S	O	–	–	P	Y	N
Related IPMs	M	O	C	–	–	Y	N
Replied-to IPM	S	O	C	–	–	Y	N
Reply Recipients	M	O	C	–	–	Y	N
Reply Requestors	M	O	C	–	–	Y	N
Reply Time	S	O	C	–	–	Y	N
Returned IPM	S	O	–	C	–	Y	N
RN Requestors	M	O	C	–	–	Y	N
S							
Sensitivity	S	O	C	–	—	Y	Y
Subject	S	O	C	–	—	Y	N
Subject IPM	S	M	–	P	P	Y	N
Suppl Receipt Info	S	O	–	–	C	Y	N
T							
Teletex Body Parts	M	O	C	–	–	N	N
Teletex Data	M	O	C	–	–	N	N
Teletex Parameters	M	O	C	–	–	N	N
This IPM	S	M	P	–	–	Y	N
V							
Videotex Body Parts	M	O	C	–	–	N	N
Videotex Data	M	O	C	–	–	N	N
Videotex Parameters	M	O	C	–	–	N	N

V Single/multi valued
L Support level by MS and access UA
P Presence in delivered message entry
L Available for List. Alert
S Available for Summarize

C.1.1 *IPM Entry Type*

The **IPM Entry Type** attribute identifies an information object's type.

```
ipm-entry-type ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX IPMEntryType
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-sat-ipm-entry-type

IPMEntryType ::= ENUMERATED {
        ipm  (0),
        rn   (1),
        nrn  (2) }
```

This attribute may assume any one of the following values:

a) *ipm:* The information object is an IPM.

b) *rn:* The information object is an RN.

c) *nrn:* The information object is an NRN.

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM or IPN.

TABLE C-2/X.420

Example of use of Child-entries

Sequence number	Entry-type	Child sequence number	Parent sequence number	IPM entry-type	Notes
100	delivered-message	101, 102	–	IPM	Delivered IPM containing two message body parts
101	delivered-message	–	100	IPM	Message body part 1
102	delivered-message	–	100	IPM	Message body part 2
120	delivered-message	–	–	RN	No child-entries possible
130	delivered-message	131	–	NRN	Contains a returned IPM
131	delivered-message	132	130	IPM	Contains one message body part
132	delivered-message	–	131	IPM	Message body part
140	delivered-report	141	–	–	Contains at least one non-delivery report
141	returned-content	142	140	IPM	Contains one message body part
142	delivered-message	–	141	IPM	Message body part

C.1.2 *IPM Synopsis*

The **IPM Synopsis** attribute gives the structure, characteristics, size, and processing status of an IPM at the granularity of individual body parts.

```
ipm-synopsis ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX IPMSynopsis
        SINGLE VALUE
        ::= id-sat-ipm-synopsis
```

The synopsis of an IPM comprises a synopsis of each of its body parts. The synopses appear in the order in which the body parts appear.

```
IPMSynopsis ::= SEQUENCE OF BodyPartSynopsis
```

The synopsis of a body part takes either of two forms depending upon whether the body part is of type Message. This enables the synopsis of a forwarding IPM to encompass the body parts of each forwarded IPM (recursively), as well as those of the forwarding IPM itself.

```
BodyPartSynopsis ::= CHOICE {
        message       [0]  MessageBodyPartSynopsis,
        non-message   [1]  NonMessageBodyPartSynopsis }

MessageBodyPartSynopsis ::= SEQUENCE {
        number        [0]  SequenceNumber,
        synopsis      [1]  IPMSynopsis }

NonMessageBodyPartSynopsis ::= SEQUENCE {
        type          [0]  OBJECT IDENTIFIER,
        parameters    [1]  ExternallyDefinedParameters,
        size          [2]  INTEGER,
        processed     [3]  BOOLEAN DEFAULT FALSE }
```

The synopsis of a Message body part has the following components:

a) **Number** (M): The sequence number that the MS assigns to the entry that the Message body part represents.

b) **Synopsis** (M): The synopsis of the IPM that forms the content of the message that the body part represents.

The synopsis of a body part of type other than Message has the following components. For purposes of this synopsis, the body part is considered to be of type Externally Defined, whether or not (see Annex B) it was so conveyed to the MS:

a) **Type** (M): The body part's extended type, i.e. the Direct-reference component of the body part's Data component. An Object Identifier.

b) **Parameters** (M): The body part's format and control parameters, i.e. the body part's Parameters component. An Any.

c) **Size** (M): The size in octets of the encoding of the Encoding component of the body part's Data component when the Basic Encoding Rules of CCITT Rec. X.209 | ISO/IEC 8825 are followed. If those rules permit several (e.g. both primitive and constructed) encodings of the component, the size may reflect any one of them. An Integer.

d) **Processed** (D *false*): An indication of whether or not the body part (as a single attribute or the data component only) has been conveyed to the UA by means of the MS's List or Fetch abstract operation. A Boolean.

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM.

Note – As a consequence of its variability, the value of the Size component should be considered only an estimate of the body part's size.

C.2 *Heading attributes*

Some attributes are derived from the Heading of an IPM. These attributes are defined and described below.

C.2.1 *Heading*

The **Heading** attribute is the (entire) Heading of an IPM.

```
heading ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX Heading
        SINGLE VALUE
        ::= id-hat-heading
```

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM.

C.2.2 *Heading analyses*

Some attributes have as their values O/R descriptors selected after analysis of the Heading. They identify the "primary", "copy", and blind "copy" recipients of an IPM of whom an RN, NRN, or reply is requested.

```
rn-requestors ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX ORDescriptor
        MATCHES FOR EQUALITY
        MULTI VALUE
        ::= id-hat-rn-requestors

nrn-requestors ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX ORDescriptor
        MATCHES FOR EQUALITY
        MULTI VALUE
        ::= id-hat-nrn-requestors

reply-requestors ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX ORDescriptor
        MATCHES FOR EQUALITY
        MULTI VALUE
        ::= id-hat-reply-requestors
```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose Heading requests, of at least one user or DL, an RN, NRN, or reply, respectively. It shall maintain one attribute value for every recipient specifier in the IPM's Primary, Copy, or Blind Copy Recipients field whose Notification-requests component includes the value rn (in the case of the first attribute) or nrn (in the case of the second), or whose Reply-requested component signifies, by either its presence or its absence, that a reply is requested (in the case of the third). The value shall be the recipient specifier's Recipient component.

C.2.3 *Heading fields*

Some attributes bear the names of heading fields and have those fields as their values. The ordering for the Expiry Time and Reply Time attributes is increasing chronological order.

```
this-ipm ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX ThisIPMField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-hat-this-ipm

originator ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX OriginatorField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-hat-originator
```

replied-to-IPM ATTRIBUTE
WITH ATTRIBUTE-SYNTAX RepliedToIPMField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-replied-to-IPM

subject ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SubjectField
MATCHES FOR EQUALITY SUBSTRINGS
SINGLE VALUE
::= id-hat-subject

expiry-time ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ExpiryTimeField
MATCHES FOR EQUALITY ORDERING
SINGLE VALUE
::= id-hat-expiry-time

reply-time ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ReplyTimeField
MATCHES FOR EQUALITY ORDERING
SINGLE VALUE
::= id-hat-reply-time

importance ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ImportanceField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-importance

sensitivity ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SensitivityField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-sensitivity

auto-forwarded ATTRIBUTE
WITH ATTRIBUTE-SYNTAX AutoForwardedField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-auto-forwarded

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose Heading contains the field whose name the attribute bears.

C.2.4 *Heading sub-fields*

Some attributes bear the names of heading fields and have sub-fields of those fields as their values.

authorizing-users ATTRIBUTE
WITH ATTRIBUTE-SYNTAX AuthorizingUsersSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-authorizing-users

primary-recipients ATTRIBUTE
WITH ATTRIBUTE-SYNTAX PrimaryRecipientsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-primary-recipients

copy-recipients ATTRIBUTE
WITH ATTRIBUTE-SYNTAX CopyRecipientsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-copy-recipients

blind-copy-recipients ATTRIBUTE
WITH ATTRIBUTE-SYNTAX BlindCopyRecipientsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-blind-copy-recipients

obsoleted-IPMs ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ObsoletedIPMsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-obsoleted-IPMs

related-IPMs ATTRIBUTE
WITH ATTRIBUTE-SYNTAX RelatedIPMsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-related-IPMs

reply-recipients ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ReplyRecipientsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-reply-recipients

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose Heading contains the field whose name the attribute bears. It shall maintain one attribute value for each sub-field.

C.2.5 *Heading extensions*

Some attributes bear the names of heading extensions and have as their values the values of those extensions or a part thereof.

incomplete-copy ATTRIBUTE
WITH ATTRIBUTE-SYNTAX IncompleteCopy
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-incomplete-copy

languages ATTRIBUTE
WITH ATTRIBUTE-SYNTAX Language
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-languages

auto-submitted ATTRIBUTE
WITH ATTRIBUTE-SYNTAX AutoSubmitted
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-auto-submitted

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose Heading contains the extension whose name the attribute bears. In the case of the Languages attribute, the MS shall maintain one attribute value for each language the extension identifies.

C.3 *Body attributes*

Some attributes are derived from the Body of an IPM. These attributes are defined and described below.

C.3.1 *Body*

The **Body** attribute is the (entire) Body of an IPM.

body ATTRIBUTE
WITH ATTRIBUTE-SYNTAX Body
SINGLE VALUE
::= id-bat-body

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM.

C.3.2 *Basic body parts*

Some attributes bear the names of basic body part types and have, with one exception, such body parts as their values.

An MS holds each forwarded IPM (i.e. each Message body part) as an information object in its own right, separate from the forwarding IPM. That information object, of course, is a message whose content is an IPM. The Message Body Parts attribute below, therefore, has as its values the sequence numbers the MS assigns to those messages.

```
ia5-text-body-parts ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX IA5TextBodyPart
        MULTI VALUE
        ::= id-bat-ia5-text-body-parts

g3-facsimile-body-parts ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX G3FacsimileBodyPart
        MULTI VALUE
        ::= id-bat-g3-facsimile-body-parts

g4-class1-body-parts ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX G4Class1BodyPart
        MULTI VALUE
        ::= id-bat-g4-class1-body-parts

teletex-body-parts ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX TeletexBodyPart
        MULTI VALUE
        ::= id-bat-teletex-body-parts

videotex-body-parts ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX VideotexBodyPart
        MULTI VALUE
        ::= id-bat-videotex-body-parts

encrypted-body-parts ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX EncryptedBodyPart
        MULTI VALUE
        ::= id-bat-encrypted-body-parts

message-body-parts ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX SequenceNumber
        MULTI VALUE
        ::= id-bat-message-body-parts

mixed-mode-body-parts ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX MixedModeBodyPart
        MULTI VALUE
        ::= id-bat-mixed-mode-body-parts

bilaterally-defined-body-parts ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX BilaterallyDefinedBodyPart
        MULTI VALUE
        ::= id-bat-bilaterally-defined-body-parts

nationally-defined-body-parts ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX NationallyDefinedBodyPart
        MULTI VALUE
        ::= id-bat-nationally-defined-body-parts
```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose Body contains one or more body parts of the type whose name the attribute bears. It shall maintain one attribute value for each such body part.

C.3.3 *Basic body part parameters components*

Some attributes bear the names of basic body part types and have the Parameters components of such body parts as their values.

ia5-text-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX IA5TextParameters
MULTI VALUE
::= id-bat-ia5-text-parameters

g3-facsimile-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX G3FacsimileParameters
MULTI VALUE
::= id-bat-g3-facsimile-parameters

teletex-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX TeletexParameters
MULTI VALUE
::= id-bat-teletex-parameters

videotex-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VideotexParameters
MULTI VALUE
::= id-bat-videotex-parameters

encrypted-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX EncryptedParameters
MULTI VALUE
::= id-bat-encrypted-parameters

message-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX MessageParameters
MULTI VALUE
::= id-bat-message-parameters

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose Body contains one or more body parts of the type whose name the attribute bears. It shall maintain one attribute value for each such body part.

C.3.4 *Basic body part data components*

Some attributes bear the names of basic body part types and have the Data components of such body parts as their values.

ia5-text-data ATTRIBUTE
WITH ATTRIBUTE-SYNTAX IA5TextData
MULTI VALUE
::= id-bat-ia5-text-data

g3-facsimile-data ATTRIBUTE
WITH ATTRIBUTE-SYNTAX G3FacsimileData
MULTI VALUE
::= id-bat-g3-facsimile-data

teletex-data ATTRIBUTE
WITH ATTRIBUTE-SYNTAX TeletexData
MULTI VALUE
::= id-bat-teletex-data

videotex-data ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VideotexData
MULTI VALUE
::= id-bat-videotex-data

encrypted-data ATTRIBUTE

WITH ATTRIBUTE-SYNTAX EncryptedData

MULTI VALUE

::= id-bat-encrypted-data

message-data ATTRIBUTE

WITH ATTRIBUTE-SYNTAX MessageData

MULTI VALUE

::= id-bat-message-data

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose Body contains one or more body parts of the type whose name the attribute bears. It shall maintain one attribute value for each such body part.

C.3.5 *Extended body part types*

The **Extended Body Part Types** attribute identifies the extended body part types represented in an IPM.

extended-body-part-types ATTRIBUTE

WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER

MATCHES FOR EQUALITY

MULTI VALUE

::= id-bat-extended-body-part-types

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose Body contains one or more Externally Defined body parts. It shall maintain one attribute value for every such type present. The value shall denote the type as specified in 7.3.12.

Note – Each value of this attribute identifies both an extended body part type represented in the IPM and the attribute type generated for the Data component(s) of the body part(s) of that body part type, as specified in C.3.6.

C.3.6 *Extended body parts*

Some attributes, unnamed, have as their values the Encoding components (see 7.3.12) of the ASN.1 Externals that constitute the Data components of Externally Defined body parts.

To each extended body part type there correspond two attributes. The first attribute is denoted by the Object Identifier that is the Direct-reference component (again, see 7.3.12) of the External that constitutes the Data component of a body part of that type. The content of this first attribute is that Data component. The second attribute is denoted by the Object Identifier that is the Direct-reference component of the External that constitutes the Parameters component of a body part of that type. The content of this second attribute is that Parameters component. This attribute is not defined for extended body part types for which no Parameters type is defined.

Where a Parameters type is defined for an extended body part type, the sequence of values in the attribute generated from the Data components of body parts of that extended body part type corresponds to the sequence of values in the attribute generated from the Parameters components of the same body parts. Thus the value created for the Data component of a body part occupies the same position in the first attribute as the value created for the Parameters component occupies in the second attribute.

An MS that supports one of these body parts shall maintain the first attribute, and, if defined, the second attribute, for an information object that it holds if, and only if, that object is a message whose content is an IPM whose Body contains one or more body parts of the type that corresponds to that attribute. It shall maintain one value of the first attribute, and, if defined, the second attribute for each such body part.

Note 1 – The extended body part attributes cannot be enumerated in practice because the extended body part types cannot be so enumerated.

Note 2 – The Extended Body Part Types attribute (see C.3.5) determines the extended body part attributes for a particular IPM.

C.4 *Notification attributes*

Some attributes are derived from an IPN. These attributes are defined and described below.

C.4.1 *Common fields*

Some attributes bear the names of common fields and have those fields as their values.

```
subject-ipm ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX SubjectIPMField
        MATCHES FOR EQUALITY SUBSTRINGS
        SINGLE VALUE
        ::= id-nat-subject-ipm

ipn-originator ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX IPNOriginatorField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= d-nat-ipn-originator

ipm-preferred-recipient ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX IPMPreferredRecipientField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-nat-ipm-preferred-recipient

conversion-eits ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX MS-EIT
        MATCHES FOR EQUALITY
        MULTI VALUE
        ::= id-nat-conversion-eits
```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPN that contains the field whose name the attribute bears.

C.4.2 *Non-receipt fields*

Some attributes bear the names of non-receipt fields and have those fields as their values.

```
non-receipt-reason ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX NonReceiptReasonField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-nat-non-receipt-reason

discard-reason ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX DiscardReasonField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-nat-discard-reason

auto-forward-comment ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX AutoForwardCommentField
        MATCHES FOR EQUALITY SUBSTRINGS
        SINGLE VALUE
        ::= id-nat-auto-forward-comment

returned-ipm ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX ReturnedIPMField
        SINGLE VALUE
        ::= id-nat-returned-ipm
```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an NRN that contains the field whose name the attribute bears.

C.4.3 *Receipt fields*

Some attributes bear the names of receipt fields and have those fields as their values. The ordering for the Receipt Time attribute is increasing chronological order.

```
receipt-time ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX ReceiptTimeField
        MATCHES FOR EQUALITY ORDERING
        SINGLE VALUE
        ::= id-nat-receipt-time

acknowledgment-mode ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX AcknowledgmentModeField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-nat-acknowledgment-mode

suppl-receipt-info ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX SupplReceiptInfoField
        MATCHES FOR EQUALITY SUBSTRINGS
        SINGLE VALUE
        ::= id-nat-suppl-receipt-info
```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an RN that contains the field whose name the attribute bears.

ANNEX D

(to Recommendation X.420)

Reference definition of Object Identifiers

(This annex forms an integral part of this Recommendation)

This annex defines for reference purposes various Object Identifiers cited in the ASN.1 modules of subsequent annexes. It uses ASN.1.

All Object Identifiers this Recommendation assigns are assigned in this annex. The annex is definitive for all but those for ASN.1 modules and the IPMS application itself. The definitive assignments for the former occur in the modules themselves; other references to them appear in IMPORT clauses. The latter is fixed.

```
IPMSObjectIdentifiers { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) object-identifiers(0) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS -- nothing -- ;

ID ::= OBJECT IDENTIFIER

-- Interpersonal Messaging (not definitive)

id-ipms ID ::= { joint-iso-ccitt mhs-motis(6) ipms(1) } -- not definitive (defined in Recommendation X.402)

-- Categories

id-mod  ID ::= { id-ipms  0 } -- modules; not definitive
id-ot   ID ::= { id-ipms  1 } -- object types
id-pt   ID ::= { id-ipms  2 } -- port types
id-ref  ID ::= { id-ipms  3 } -- refinements
id-et   ID ::= { id-ipms  4 } -- extended body part types
```

```
id-hex   ID ::= { id-ipms   5 } -- heading extensions
id-sat   ID ::= { id-ipms   6 } -- summary attributes
id-hat   ID ::= { id-ipms   7 } -- heading attributes
id-bat   ID ::= { id-ipms   8 } -- body attributes
id-nat   ID ::= { id-ipms   9 } -- notification attributes
id-mct   ID ::= { id-ipms 10 } -- message content types
id-ep    ID ::= { id-ipms 11 } -- extended body part parameters
id-eit   ID ::= { id-ipms 12 } -- encoded information types

-- Modules

id-mod-object-identifiers              ID ::= { id-mod   0 } -- not definitive
id-mod-functional-objects              ID ::= { id-mod   1 } -- not definitive
id-mod-information-objects             ID ::= { id-mod   2 } -- not definitive
id-mod-abstract-service                ID ::= { id-mod   3 } -- not definitive
id-mod-heading-extensions              ID ::= { id-mod   6 } -- not definitive
id-mod-extended-body-part-types        ID ::= { id-mod   7 } -- not definitive
id-mod-message-store-attributes        ID ::= { id-mod   8 } -- not definitive
id-mod-file-transfer-body-part-type    ID ::= { id-mod   9 } -- not definitive
id-mod-upper-bounds                    ID ::= { id-mod 10 } -- not definitive
id-mod-extended-voice-body-part-type   ID ::= { id-mod 11 } -- not definitive

-- Object types

id-ot-ipme         ID ::= { id-ot 0 }
id-ot-ipms-user    ID ::= { id-ot 1 }
id-ot-ipms         ID ::= { id-ot 2 }
id-ot-ipms-ua      ID ::= { id-ot 3 }
id-ot-ipms-ms      ID ::= { id-ot 4 }
id-ot-tlma         ID ::= { id-ot 5 }
id-ot-tlxau        ID ::= { id-ot 6 }
id-ot-pdau         ID ::= { id-ot 7 }

-- Port types

id-pt-origination      ID ::= { id-pt 0 }
id-pt-reception        ID ::= { id-pt 1 }
id-pt-management       ID ::= { id-pt 2 }

-- Refinements

id-ref-primary      ID ::= { id-ref 0 }
id-ref-secondary    ID ::= { id-ref 1 }

-- Extended body part types

id-et-ia5-text              ID ::= { id-et   0 }
id-et-g3-facsimile          ID ::= { id-et   2 }
id-et-g4-class1             ID ::= { id-et   3 }
id-et-teletex               ID ::= { id-et   4 }
id-et-videotex              ID ::= { id-et   5 }
id-et-encrypted             ID ::= { id-et   6 }
id-et-message               ID ::= { id-et   7 }
id-et-mixed-mode            ID ::= { id-et   8 }
id-et-bilaterally-defined   ID ::= { id-et   9 }
id-et-nationally-defined    ID ::= { id-et 10 }
id-et-general-text          ID ::= { id-et 11 }
id-et-file-transfer         ID ::= { id-et 12 }
id-et-vmg-body-part         ID ::= { id-et 13 }

-- Heading extensions

id-hex-incomplete-copy   ID ::= { id-hex 0 }
id-hex-languages         ID ::= { id-hex 1 }
id-hex-auto-submitted    ID ::= { id-hex 2 }
```

-- Summary attributes

```
id-sat-ipm-entry-type  ID ::= { id-sat 0 }
id-sat-ipm-synopsis    ID ::= { id-sat 1 }
```

-- Heading attributes

```
id-hat-heading                 ID ::= { id-hat   0 }
id-hat-this-ipm                ID ::= { id-hat   1 }
id-hat-originator              ID ::= { id-hat   2 }
id-hat-replied-to-IPM          ID ::= { id-hat   3 }
id-hat-subject                 ID ::= { id-hat   4 }
id-hat-expiry-time             ID ::= { id-hat   5 }
id-hat-reply-time              ID ::= { id-hat   6 }
id-hat-importance              ID ::= { id-hat   7 }
id-hat-sensitivity             ID ::= { id-hat   8 }
id-hat-auto-forwarded          ID ::= { id-hat   9 }
id-hat-authorizing-users       ID ::= { id-hat 10 }
id-hat-primary-recipients      ID ::= { id-hat 11 }
id-hat-copy-recipients         ID ::= { id-hat 12 }
id-hat-blind-copy-recipients   ID ::= { id-hat 13 }
id-hat-obsoleted-IPMs          ID ::= { id-hat 14 }
id-hat-related-IPMs            ID ::= { id-hat 15 }
id-hat-reply-recipients        ID ::= { id-hat 16 }
id-hat-incomplete-copy         ID ::= { id-hat 17 }
id-hat-languages               ID ::= { id-hat 18 }
id-hat-rn-requestors           ID ::= { id-hat 19 }
id-hat-nrn-requestors          ID ::= { id-hat 20 }
id-hat-reply-requestors        ID ::= { id-hat 21 }
id-hat-auto-submitted          ID ::= { id-hat 22 }
```

-- Body attributes

```
id-bat-body                            ID ::= { id-bat   0 }
id-bat-ia5-text-body-parts             ID ::= { id-bat   1 }
id-bat-g3-facsimile-body-parts         ID ::= { id-bat   3 }
id-bat-g4-class1-body-parts            ID ::= { id-bat   4 }
id-bat-teletex-body-parts              ID ::= { id-bat   5 }
id-bat-videotex-body-parts             ID ::= { id-bat   6 }
id-bat-encrypted-body-parts            ID ::= { id-bat   7 }
id-bat-message-body-parts              ID ::= { id-bat   8 }
id-bat-mixed-mode-body-parts           ID ::= { id-bat   9 }
id-bat-bilaterally-defined-body-parts  ID ::= { id-bat 10 }
id-bat-nationally-defined-body-parts   ID ::= { id-bat 11 }
id-bat-extended-body-part-types        ID ::= { id-bat 12 }
id-bat-ia5-text-parameters             ID ::= { id-bat 13 }
id-bat-g3-facsimile-parameters         ID ::= { id-bat 15 }
id-bat-teletex-parameters              ID ::= { id-bat 16 }
id-bat-videotex-parameters             ID ::= { id-bat 17 }
id-bat-encrypted-parameters            ID ::= { id-bat 18 }
id-bat-message-parameters              ID ::= { id-bat 19 }
id-bat-ia5-text-data                   ID ::= { id-bat 20 }
id-bat-g3-facsimile-data               ID ::= { id-bat 22 }
id-bat-teletex-data                    ID ::= { id-bat 23 }
id-bat-videotex-data                   ID ::= { id-bat 24 }
id-bat-encrypted-data                  ID ::= { id-bat 25 }
id-bat-message-data                    ID ::= { id-bat 26 }
```

-- Notification attributes

```
id-nat-subject-ipm             ID ::= { id-nat   0 }
id-nat-ipn-originator          ID ::= { id-nat   1 }
id-nat-ipm-preferred-recipient ID ::= { id-nat   2 }
id-nat-conversion-eits         ID ::= { id-nat   3 }
id-nat-non-receipt-reason      ID ::= { id-nat   4 }
id-nat-discard-reason          ID ::= { id-nat   5 }
```

```
id-nat-auto-forward-comment   ID ::= { id-nat   6 }
id-nat-returned-ipm           ID ::= { id-nat   7 }
id-nat-receipt-time           ID ::= { id-nat   8 }
id-nat-acknowledgment-mode    ID ::= { id-nat   9 }
id-nat-suppl-receipt-info     ID ::= { id-nat 10 }

-- Message content types (for use by MS only)

id-mct-p2-1984 ID ::= { id-mct 0 } -- P2 1984
id-mct-p2-1988 ID ::= { id-mct 1 } -- P2 1988

-- Extended body part parameters

id-ep-ia5-text          ID ::= { id-ep   0 }
id-ep-g3-facsimile      ID ::= { id-ep   2 }
id-ep-teletex           ID ::= { id-ep   4 }
id-ep-videotex          ID ::= { id-ep   5 }
id-ep-encrypted         ID ::= { id-ep   6 }
id-ep-message           ID ::= { id-ep   7 }
id-ep-general-text      ID ::= { id-ep 11 }
id-ep-file-transfer     ID ::= { id-ep 12 }
id-ep-vmg-body-part     ID ::= { id-ep 13 }
id-eit-file-transfer    ID ::= { id-eit 0 } -- EIT

END -- of IPMSObjectIdentifiers

-- For General Text Body Part

IPMSObjectIdentifiers2 { iso standard motis(10021) ipms(7) modules(0) object-identifiers(0) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS -- nothing  -- ;

ID ::= OBJECT IDENTIFIER

-- Interpersonal Messaging (ISO/IEC extensions)

id-iso-ipms ID ::= { iso standard motis(10021) ipms(7) }

-- Categories

id-iso-mod ID ::= { id-iso-ipms 0 } -- modules; not definitive
id-iso-cs  ID ::= { id-iso-ipms  1 } -- character sets

-- Modules

id-mod-object-identifiers-2          ID ::= { id-iso-mod 0 } -- not definitive
id-mod-extended-body-part-types-2    ID ::= { id-iso-mod 1 } -- not definitive

-- Registration Authority for General Text Character Set EITs

id-cs-eit-authority                  ID ::= { id-iso-cs  0 }

END -- of IPMSObjectIdentifiers2
```

ANNEX E

(to Recommendation X.420)

Reference definition of abstract information objects

(This annex forms an integral part of this Recommendation.)

This annex, a supplement to Section 2, defines for reference purposes the abstract information objects of Interpersonal Messaging.

```
IPMSInformationObjects { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) information-objects(2) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS

        -- IPMS Upper bounds (from Recommendation X.420, Annex K)

          ub-auto-forward-comment, ub-free-form-name, ub-ipm-identifier-suffix, ub-local-ipm-identifier, ub-subject-
          field, ub-telephone-number
                ----
                FROM IPMSUpperBounds { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) upper-bounds(10) }

          -- ODIF

          Interchange-Data-Element -- see Recommendation T.415
                ----
                FROM Interchange-Data-Elements { 2 8 1 5 5 }

          -- MTS Abstract Service [from Recommendation X.411 (1992)]

          EncodedInformationTypes, G3FacsimileNonBasicParameters, MessageDeliveryTime, ORName,
          OtherMessageDeliveryFields, SupplementaryInformation, TeletexNonBasicParameters
                ----
                FROM MTSAbstractService { joint-iso-ccitt mhs-motis(6) mts(3) modules(0) mts-abstract-
          service(1) };

Time ::= UTCTime

-- Information object

InformationObject ::= CHOICE {
        ipm [0] IPM,
        ipn [1] IPN }

-- IPM

IPM ::= SEQUENCE {
         heading Heading,
         body      Body }

-- IPMS Extensions

IPMSExtension ::= SEQUENCE {
           type OBJECT IDENTIFIER,
             Value ANY DEFINED BY type DEFAULT NULL NULL }

IPMS-EXTENSION MACRO ::=
BEGIN
       TYPE NOTATION ::= "VALUE" type | empty
           VALUE NOTATION ::= value (VALUE OBJECT IDENTIFIER)
END
```

```
-- Heading

Heading ::= SET {
        this-IPM                    ThisIPMField,
        originator              [0] OriginatorField OPTIONAL,
        authorizing-users       [1] AuthorizingUsersField OPTIONAL,
        primary-recipients      [2] PrimaryRecipientsField DEFAULT { },
        copy-recipients         [3] CopyRecipientsField DEFAULT { },
        blind-copy-recipients   [4] BlindCopyRecipientsField OPTIONAL,
        replied-to-IPM          [5] RepliedToIPMField OPTIONAL,
        obsoleted-IPMs          [6] ObsoletedIPMsField DEFAULT { },
        related-IPMs            [7] RelatedIPMsField DEFAULT { },
        subject                 [8] EXPLICIT SubjectField OPTIONAL,
        expiry-time             [9] ExpiryTimeField OPTIONAL,
        reply-time             [10] ReplyTimeField OPTIONAL,
        reply-recipients       [11] ReplyRecipientsField OPTIONAL,
        importance             [12] ImportanceField DEFAULT normal,
        sensitivity            [13] SensitivityField OPTIONAL,
        auto-forwarded         [14] AutoForwardedField DEFAULT FALSE,
        extensions             [15] ExtensionsField DEFAULT { } }

-- Heading component types

IPMIdentifier ::= [APPLICATION 11] SET {
        user                    ORName OPTIONAL,
        user-relative-identifier LocalIPMIdentifier }

LocalIPMIdentifier ::= PrintableString (SIZE (0..ub-local-ipm-identifier) )

RecipientSpecifier ::= SET {
        recipient               [0] ORDescriptor,
        notification-requests   [1] NotificationRequests DEFAULT { },
        reply-requested[        [2] BOOLEAN DEFAULT FALSE,
        recipient-extensions    [3] RecipientExtensionsField OPTIONAL }

ORDescriptor ::= SET {
        formal-name                 ORName OPTIONAL,
        free-form-name      [0]     FreeFormName OPTIONAL,
        telephone-number    [1]     TelephoneNumber OPTIONAL }

FreeFormName ::= TeletexString (SIZE (0..ub-free-form-name) )

TelephoneNumber ::= PrintableString (SIZE (0..ub-telephone-number) )

NotificationRequests ::= BIT STRING {
        rn          (0),
        nrn         (1),
        ipm-return  (2) }

RecipientExtensionsField ::= SET OF IPMSExtension

-- This IPM heading field

ThisIPMField ::= IPMIdentifier

-- Originator heading field

OriginatorField ::= ORDescriptor

-- Authorizing Users heading field

AuthorizingUsersField      ::= SEQUENCE OF AuthorizingUsersSubfield

AuthorizingUsersSubfield  ::= ORDescriptor

-- Primary Recipients heading field

PrimaryRecipientsField       ::= SEQUENCE OF PrimaryRecipientsSubfield

PrimaryRecipientsSubfield    ::= RecipientSpecifier
```

```
-- Copy Recipients heading field

CopyRecipientsField         ::= SEQUENCE OF CopyRecipientsSubfield

CopyRecipientsSubfield      ::= RecipientSpecifier

-- Blind Copy Recipients heading field

BlindCopyRecipientsField    ::= SEQUENCE OF BlindCopyRecipientsSubfield

BlindCopyRecipientsSubfield ::= RecipientSpecifier

-- Replied-to IPM heading field

RepliedToIPMField ::= IPMIdentifier

-- Obsoleted IPMs heading field

ObsoletedIPMsField          ::= SEQUENCE OF ObsoletedIPMsSubfield

ObsoletedIPMsSubfield       ::= IPMIdentifier

-- Related IPMs heading field

RelatedIPMsField            ::= SEQUENCE OF RelatedIPMsSubfield

RelatedIPMsSubfield         ::= IPMIdentifier

-- Subject heading field

SubjectField ::= TeletexString (SIZE (0..ub-subject-field) )

-- Expiry Time heading field

ExpiryTimeField ::= Time

-- Reply Time heading field

ReplyTimeField ::= Time

-- Reply Recipients heading field

ReplyRecipientsField        ::= SEQUENCE OF ReplyRecipientsSubfield

ReplyRecipientsSubfield     ::= ORDescriptor

-- Importance heading field

ImportanceField ::= ENUMERATED {
        low         (0),
        normal      (1),
        high        (2) }

-- Sensitivity heading field

SensitivityField ::= ENUMERATED {
        personal                (1),
        private                 (2),
        company-confidential    (3) }

-- Auto-forwarded heading field

AutoForwardedField ::= BOOLEAN

-- Extensions heading field

ExtensionsField ::= SET OF IPMSExtension

-- Body

Body ::= SEQUENCE OF BodyPart

BodyPart ::= CHOICE {
        ia5-text            [0]  IA5TextBodyPart,
        g3-facsimile        [3]  G3FacsimileBodyPart,
        g4-class1           [4]  G4Class1BodyPart,
        teletex             [5]  TeletexBodyPart,
        videotex            [6]  VideotexBodyPart,
```

```
        encrypted               [8]   EncryptedBodyPart,
        message                 [9]   MessageBodyPart,
        mixed-mode              [11]  MixedModeBodyPart,
        bilaterally-defined     [14]  BilaterallyDefinedBodyPart,
        nationally-defined      [7]   NationallyDefinedBodyPart,
        externally-defined      [15]  ExternallyDefinedBodyPart }

-- Note – Voice Body Part was removed from Rec. X.420 (1992) and replaced with an externally defined" Voice Body
-- Part".

-- IA5 Text body part

IA5TextBodyPart ::= SEQUENCE {
        parameters  IA5TextParameters,
        data        IA5TextData }

IA5TextParameters ::= SET {
        repertoire [0] Repertoire DEFAULT ia5 }

IA5TextData ::= IA5String

Repertoire ::= ENUMERATED {
        ita2  (2),
        ia5   (5) }

-- G3 Facsimile body part

G3FacsimileBodyPart ::= SEQUENCE {
        parameters   G3FacsimileParameters,
        data         G3FacsimileData }

G3FacsimileParameters ::= SET {
        number-of-pages          [0]   INTEGER OPTIONAL,
        non-basic-parameters     [1]   G3FacsimileNonBasicParameters OPTIONAL }

G3FacsimileData ::= SEQUENCE OF BIT STRING

-- G4 Class 1 and Mixed-mode body parts

G4Class1BodyPart      ::= SEQUENCE OF Interchange-Data-Element

MixedModeBodyPart     ::= SEQUENCE OF Interchange-Data-Element

-- Teletex body part

TeletexBodyPart ::= SEQUENCE {
        parameters     TeletexParameters,
        data           TeletexData }

TeletexParameters ::= SET {
        number-of-pages          [0]   INTEGER OPTIONAL,
        telex-compatible         [1]   BOOLEAN DEFAULT FALSE,
        non-basic-parameters     [2]   TeletexNonBasicParameters OPTIONAL }

TeletexData ::= SEQUENCE OF TeletexString

-- Videotex body part

VideotexBodyPart ::= SEQUENCE {
        parameters     VideotexParameters,
        data           VideotexData }

VideotexParameters ::= SET {
        syntax [0] VideotexSyntax OPTIONAL }

VideotexSyntax ::= INTEGER {
        ids            (0),
        data-syntax1   (1),
        data-syntax2   (2),
        data-syntax3   (3) }

VideotexData ::= VideotexString
```

```
-- Encrypted body part

EncryptedBodyPart ::= SEQUENCE {
        parameters      EncryptedParameters,
        data            EncryptedData }

EncryptedParameters ::= SET OF ANY -- for further study

EncryptedData ::= BIT STRING -- for further study

-- Voice Body Part (see Annex I of this Recommendation)

-- Message body part

MessageBodyPart ::= SEQUENCE {
        parameters  MessageParameters,
        data        MessageData }

MessageParameters ::= SET {
        delivery-time       [0]  MessageDeliveryTime OPTIONAL,
        delivery-envelope   [1]  OtherMessageDeliveryFields OPTIONAL }

MessageData ::= IPM

-- Bilaterally Defined body part

BilaterallyDefinedBodyPart ::= OCTET STRING

-- Nationally Defined body part

NationallyDefinedBodyPart ::= ANY

-- Externally Defined body part

ExternallyDefinedBodyPart ::= SEQUENCE {
        parameters  [0]  ExternallyDefinedParameters OPTIONAL,
        data             ExternallyDefinedData }

ExternallyDefinedParameters ::= EXTERNAL

ExternallyDefinedData ::= EXTERNAL

EXTENDED-BODY-PART-TYPE MACRO ::=
BEGIN
        TYPE NOTATION     ::= Parameters Data
        VALUE NOTATION    ::= value (VALUE OBJECT IDENTIFIER )

        Parameters        ::= "PARAMETERS" type "IDENTIFIED" "BY" value (OBJECT IDENTIFIER) | empty
        Data              ::= "DATA" type
END

-- IPN

IPN ::= SET {
        -- common-fields -- COMPONENTS OF CommonFields,
        choice [0] CHOICE {
                non-receipt-fields              [0]  NonReceiptFields,
                receipt-fields                  [1]  ReceiptFields,
                other-notification-type-fields  [2]  OtherNotificationTypeFields } }

RN ::= IPN -- with receipt-fields chosen

NRN ::= IPN -- with non-receipt-fields chosen

ON ::= IPN -- with other-notification-type-fields chosen

CommonFields ::= SET {
        subject-ipm                 SubjectIPMField,
        ipn-originator          [1] IPNOriginatorField OPTIONAL,
        ipm-preferred-recipient [2] IPMPreferredRecipientField OPTIONAL,
        conversion-eits             ConversionEITsField OPTIONAL,
        notification-extensions [3] NotificationExtensionsField OPTIONAL }
```

```
NonReceiptFields ::= SET {
        non-receipt-reason      [0]  NonReceiptReasonField,
        discard-reason          [1]  DiscardReasonField OPTIONAL,
        auto-forward-comment    [2]  AutoForwardCommentField OPTIONAL,
        returned-ipm            [3]  ReturnedIPMField OPTIONAL,
        nrn-extensions          [4]  NRNExtensionsField OPTIONAL }

ReceiptFields ::= SET {
        receipt-time            [0]  ReceiptTimeField,
        acknowledgment-mode     [1]  AcknowledgmentModeField DEFAULT manual,
        suppl-receipt-info      [2]  SupplReceiptInfoField OPTIONAL,
        rn-extensions           [3]  RNExtensionsField OPTIONAL }

-- Common fields

SubjectIPMField ::= IPMIdentifier

IPNOriginatorField ::= ORDescriptor

IPMPreferredRecipientField ::= ORDescriptor

ConversionEITsField ::= EncodedInformationTypes

NotificationExtensionsField ::= SET OF IPMSExtension

-- Non-receipt fields

NonReceiptReasonField ::= ENUMERATED {
        ipm-discarded       (0),
        ipm-auto-forwarded  (1) }

DiscardReasonField ::= ENUMERATED {
        ipm-expired                 (0),
        ipm-obsoleted               (1),
        user-subscription-terminated (2) }

AutoForwardCommentField ::= AutoForwardComment

AutoForwardComment ::= PrintableString (SIZE (0..ub-auto-forward-comment ) )

ReturnedIPMField ::= IPM

NRNExtensionsField ::= SET OF IPMSExtension

-- Receipt fields

ReceiptTimeField ::= Time

AcknowledgmentModeField ::= ENUMERATED {
        manual      (0),
        automatic   (1) }

SupplReceiptInfoField ::= SupplementaryInformation

RNExtensionsField ::= SET OF IPMSExtension

-- Other Notification Type fields

OtherNotificationTypeFields ::= SET OF IPMSExtension

-- Message Store Realization

ForwardedInfo ::= SET {
        auto-forwarding-comment  [0]  AutoForwardComment OPTIONAL,
        cover-note               [1]  IA5TextBodyPart OPTIONAL,
        this-ipm-prefix          [2]  PrintableString (SIZE (1..ub-ipm-identifier-suffix ) ) OPTIONAL }

END -- of IPMSInformationObjects
```

ANNEX F

(to Recommendation X.420)

Reference definition of functional objects

(This annex forms an integral part of this Recommendation)

This annex, a supplement to clauses 10, 11, and 16, defines for reference purposes the functional objects of Interpersonal Messaging. It uses the OBJECT and REFINE macros of CCITT Rec. X.407 | ISO/IEC 10021-3.

```
IPMSFunctionalObjects { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) functional-objects (1) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS

    -- IPMS Abstract Service (from Annex D)

    management, origination, reception
        ----
        FROM IPMSAbstractService { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) abstract-service (3) }

    -- IPMS Object Identifiers (from Annex D)

    id-ot-ipme, id-ot-ipms, id-ot-ipms-ms, id-ot-ipms-ua, id-ot-ipms-user, id-ot-pdau, id-ot-tlma, id-ot-tlxau, id-
    ref-primary, id-ref-secondary
        ----
        FROM IPMSObjectIdentifiers { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)
    object-identifiers (0) }

    -- TLMA Abstract Service

        miscellanea
        ----
        FROM TLMAAbsService {ccitt recommendation(0) t(20) 330 tlmaabsservice(0) }

    -- MS Abstract Service [from Recommendation X.413 (1992)]

        retrieval
        ----
        FROM MSAbstractService { joint-iso-ccitt mhs-motis(6) ms(4) modules(0) abstract-service(1) }

    -- MTS Abstract Service [from Recommendation X.411 (1992)]

        administration,delivery, mTS, submission
        ----
        FROM MTSAbstractService { joint-iso-ccitt mhs-motis(6) mts(3) modules(0)
    mts-abstract-service (1) }

    -- Abstract service definition conventions [from Recommendation X.407 (1988)]

        OBJECT, REFINE,
        ----
        FROM AbstractServiceNotation { joint-iso-ccitt mhs-motis(6) asdc(2) modules(0) notation(1) };

-- "Root" object type

ipme OBJECT
    ::= id-ot-ipme
```

```
-- Primary refinement

ipme-refinement REFINE ipme AS
        ipms
                origination     [S]   PAIRED WITH ipms-user
                reception       [S]   PAIRED WITH ipms-user
                management      [S]   PAIRED WITH ipms-user
        ipms-user RECURRING
        ::= id-ref-primary

-- Primary object types

ipms-user OBJECT
        PORTS {
                origination     [C],
                reception       [C],
                management      [C] }
        ::= id-ot-ipms-user

ipms OBJECT
        PORTS {
                origination     [S],
                reception       [S],
                management      [S] }
        ::= id-ot-ipms

-- Secondary refinement

ipms-refinement REFINE ipms AS
        mTS
                submission      [S]   PAIRED WITH ipms-ua, ipms-ms
                delivery        [S]   PAIRED WITH ipms-ua, ipms-ms
                administration  [S]   PAIRED WITH ipms-ua, ipms-ms
        ipms-ua RECURRING
                origination     [S]   VISIBLE
                reception       [S]   VISIBLE
                management      [S]   VISIBLE
        ipms-ms RECURRING
                submission      [S]   PAIRED WITH ipms-ua
                retrieval       [S]   PAIRED WITH ipms-ua
                administration  [S]   PAIRED WITH ipms-ua
        tlma RECURRING
                origination     [S]   VISIBLE
                reception       [S]   VISIBLE
                management      [S]   VISIBLE
        tlxau RECURRING
                origination     [S]   VISIBLE
                reception       [S]   VISIBLE
                management      [S]   VISIBLE
       pdau RECURRING
                reception       [S]   VISIBLE
        ::= id-ref-secondary

-- Secondary objects

ipms-ua OBJECT
        PORTS {
                origination     [S],
                reception       [S],
                management      [S],
                submission      [C],
                delivery        [C],
                retrieval       [C],
                administration  [C] }
        ::= id-ot-ipms-ua
```

```
ipms-ms OBJECT
        PORTS {
                submission      [S],
                retrieval       [S],
                administration  [S],
                submission      [C],
                delivery        [C],
                administration  [C] }
        ::= id-ot-ipms-ms

tlma OBJECT
        PORTS {
                origination     [S],
                reception       [S],
                management      [S],
                miscellanea     [S] }
        ::= id-ot-tlma

tlxau OBJECT
        PORTS {
                origination     [S],
                reception       [S],
                management      [S] }
        ::= id-ot-tlxau

pdau OBJECT
        PORTS {
                reception       [S] }
        ::= id-ot-pdau

END -- of IPMSFunctionalObjects
```

ANNEX G

(to Recommendation X.420)

Reference definition of Abstract Service

(This annex forms an integral part of this Recommendation)

This annex, a supplement to clauses 12 and 13, defines for reference purposes the IPMS Abstract Service. It uses the PORT and ABSTRACT-OPERATION and ABSTRACT-ERROR macros of CCITT Rec. X.407 | ISO/IEC 10021-3.

```
IPMSAbstractService { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) abstract-service(3) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS

        -- IPMS Information Objects (from Annex E)

        AutoForwardComment, Heading, InformationObject, IPM, NRN, RN
            ----
            FROM IPMSInformationObjects { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)
        information-objects(2) }

        -- IPMS Object Identifiers (from Annex D)

        id-pt-management, id-pt-origination, id-pt-reception
            ----
            FROM IPMSObjectIdentifiers { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)    object-identifiers(0) }
```

```
        -- MTS Abstract Service [from Recommendation X.411 (1992)]

        MessageDeliveryEnvelope, MessageSubmissionEnvelope, MessageSubmissionIdentifier,
            MessageSubmissionTime, ORName, ProbeSubmissionEnvelope,
            ProbeSubmissionIdentifier, ProbeSubmissionTime, RecipientImproperlySpecified,
            ReportDeliveryEnvelope, SupplementaryInformation
            ----
            FROM MTSAbstractService { joint-iso-ccitt mhs-motis(6) mts(3) modules(0)
        mts-abstract-service(1) }

        -- Abstract service definition conventions [from Recommendation X.407 (1988)]

        ABSTRACT-ERROR, ABSTRACT-OPERATION, PORT
            ----
            FROM AbstractServiceNotation { joint-iso-ccitt mhs-motis(6) asdc(2) modules(0) notation(1) };

Time ::= UTCTime

-- Ports

origination PORT
        CONSUMER INVOKES {
                OriginateProbe,
                OriginateIPM,
                OriginateRN,
                OriginateON }
        ::= id-pt-origination

-- Note – Originate ON is not defined. In this version of the Specification it is a placeholder.

reception PORT
        SUPPLIER INVOKES {
                ReceiveReport,
                ReceiveIPM,
                ReceiveRN,
                ReceiveNRN,
                ReceiveON }
        ::= id-pt-reception

-- Note – ReceiveOn is not defined. In this version of the Specification it is a placeholder.

management PORT
        CONSUMER INVOKES {
                ChangeAutoDiscard,
                ChangeAutoAcknowledgment,
                ChangeAutoForwarding }
        ::= id-pt-management

-- Origination abstract operations

OriginateProbe ::= ABSTRACT-OPERATION
        ARGUMENT SET {
                envelope    [0]  ProbeSubmissionEnvelope,
                content     [1]  IPM }
        RESULT SET {
                submission-identifier  [0]  ProbeSubmissionIdentifier,
                submission-time        [1]  ProbeSubmissionTime }

        ERRORS {
                SubscriptionError,
                RecipientImproperlySpecified }
```

```
OriginateIPM ::= ABSTRACT-OPERATION
        ARGUMENT SET {
                envelope    [0]  MessageSubmissionEnvelope,
                content     [1]  IPM }
        RESULT SET {
                submission-identifier  [0]  MessageSubmissionIdentifier,
                submission-time        [1]  MessageSubmissionTime }
        ERRORS {
                SubscriptionError,
                RecipientImproperlySpecified }

OriginateRN ::= ABSTRACT-OPERATION
        ARGUMENT SET {
                envelope  [0]  MessageSubmissionEnvelope,
                content   [1]  RN }
        RESULT SET {
                submission-identifier  [0]  MessageSubmissionIdentifier,
                submission-time        [1]  MessageSubmissionTime }
        ERRORS {
                SubscriptionError,
                RecipientImproperlySpecified }

-- Reception abstract operations

ReceiveReport ::= ABSTRACT-OPERATION
        ARGUMENT SET {
                envelope              [0]  ReportDeliveryEnvelope,
                undelivered-object    [1]  InformationObject OPTIONAL }
        RESULT
        ERRORS { }

ReceiveIPM ::= ABSTRACT-OPERATION
        ARGUMENT SET {
                envelope    [0]  MessageDeliveryEnvelope,
                content     [1]  IPM }
        RESULT
        ERRORS { }

ReceiveRN ::= ABSTRACT-OPERATION
        ARGUMENT SET {
                envelope    [0]  MessageDeliveryEnvelope,
                content     [1]  RN }
        RESULT
        ERRORS { }

ReceiveNRN ::= ABSTRACT-OPERATION
        ARGUMENT SET {
                envelope    [0]  MessageDeliveryEnvelope,
                content     [1]  NRN }
        RESULT
        ERRORS { }

-- Management abstract operations

ChangeAutoDiscard ::= ABSTRACT-OPERATION
        ARGUMENT SET {
                auto-discard-expired-IPMs    [0]  BOOLEAN,
                auto-discard-obsolete-IPMs   [1]  BOOLEAN }
        RESULT
        ERRORS { }

ChangeAutoAcknowledgment ::= ABSTRACT-OPERATION
        ARGUMENT SET {
                auto-acknowledge-IPMs    [0]  BOOLEAN,
                auto-acknowledge-suppl-receipt-info    [1]
                 SupplementaryInformation }
        RESULT
        ERRORS {
                SubscriptionError }
```

```
ChangeAutoForwarding ::= ABSTRACT-OPERATION
        ARGUMENT SET {
                auto-forward-IPMs       [0]   BOOLEAN,
                auto-forward-recipient  [1]   SEQUENCE OF ORName OPTIONAL,
                auto-forward-heading    [2]   Heading OPTIONAL,
                auto-forward-comment    [3]   AutoForwardComment OPTIONAL }
        RESULT
        ERRORS {
                SubscriptionError,
                RecipientImproperlySpecified }

-- Abstract errors

SubscriptionError ::= ABSTRACT-ERROR
       PARAMETER SET {
                problem  [0]  SubscriptionProblem }

SubscriptionProblem ::= ENUMERATED {
        ipms-eos-not-subscribed  (0),
        mts-eos-not-subscribed   (1) }

END -- of IPMSAbstractService
```

ANNEX H

(to Recommendation X.420)

Reference definition of heading extensions

(This annex forms an integral part of this Recommendation.)

This annex, a supplement to Annex A, defines for reference purposes the heading extensions defined for Interpersonal Messaging. It uses the IPMS-EXTENSION macro of 7.2.17.

```
IPMSHeadingExtensions { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) heading-extensions(6) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS

        -- IPMS Information Objects (from Annex E)

        IPMS-EXTENSION
             ----
             FROM IPMSInformationObjects { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)
        information-objects(2) }

        -- IPMS Object Identifiers  (from Annex D)

        id-hex-auto-submitted, id-hex-incomplete-copy, id-hex-languages
             ----
             FROM IPMSObjectIdentifiers { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)    object-identifiers(0)
};

-- Incomplete Copy

incomplete-copy IPMS-EXTENSION
        ::= id-hex-incomplete-copy

IncompleteCopy ::= NULL
```

```
-- Languages

languages IPMS-EXTENSION
        VALUE SET OF Language
        ::= id-hex-languages

Language ::= PrintableString (SIZE (2..5) )

-- Auto-submitted

auto-submitted IPMS-EXTENSION
        VALUE AutoSubmitted
        ::= id-hex-auto-submitted

AutoSubmitted ::= ENUMERATED {
        not-auto-submitted (0),
        auto-generated     (1),
        auto-replied       (2),
        auto-forwarded     (3) }

END -- of IPMSHeadingExtensions
```

ANNEX I

(to Recommendation X.420)

Reference definition of extended body part types

(This annex forms an integral part of this Recommendation)

This annex, a supplement to Annex B, defines for reference purposes certain extended body part types.

I.1 *Equivalents of basic body part types*

```
IPMSExtendedBodyPartTypes { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) extended-body-part-types(7) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

--Prologue

-- Exports everything.

IMPORTS

        -- IPMS Information Objects (from Annex E)

        BilaterallyDefinedBodyPart, EncryptedData, EncryptedParameters, EXTENDED-BODY-PART-TYPE,
        G3FacsimileData, G3FacsimileParameters, G4Class1BodyPart, IA5TextData, IA5TextParameters,
        MessageData, MessageParameters, MixedModeBodyPart, NationallyDefinedBodyPart, TeletexData,
        TeletexParameters, VideotexData, VideotexParameters
            ----
            FROM IPMSInformationObjects { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) information-
        objects(2) }

        -- IPMS Object Identifiers (from Annex D)

        id-ep-encrypted, id-ep-g3-facsimile, id-ep-ia5-text, id-ep-message, id-ep-teletex, id-ep-videotex,
        id-et-bilaterally-defined, id-et-encrypted id-et-g3-facsimile, id-et-g4-class1, id-et-ia5-text, id-et-message, id-
        et-mixed-mode, id-et-nationally-defined, id-et-teletex, id-et-videotex
            ----
            FROM IPMSObjectIdentifiers { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)    object-identifiers(0)
};
```

```
-- Extended IA5 Text body part

ia5-text-body-part EXTENDED-BODY-PART-TYPE
        PARAMETERS    IA5TextParameters IDENTIFIED BY id-ep-ia5-text
        DATA          IA5TextData
        ::= id-et-ia5-text

-- Extended G3 Facsimile body part

g3-facsimile-body-part EXTENDED-BODY-PART-TYPE
        PARAMETERS    G3FacsimileParameters IDENTIFIED BY id-ep-g3-facsimile
        DATA          G3FacsimileData
        ::= id-et-g3-facsimile

-- Extended G4 Class 1 body part

g4-class1-body-part EXTENDED-BODY-PART-TYPE
        DATA G4Class1BodyPart
        ::= id-et-g4-class1

-- Extended Teletex body part

teletex-body-part EXTENDED-BODY-PART-TYPE
        PARAMETERS    TeletexParameters IDENTIFIED BY id-ep-teletex
        DATA          TeletexData
        ::= id-et-teletex

-- Extended Videotex body part

videotex-body-part EXTENDED-BODY-PART-TYPE
        PARAMETERS    VideotexParameters IDENTIFIED BY id-ep-videotex
        DATA          VideotexData
        ::= id-et-videotex

-- Extended Encrypted body part

encrypted-body-part EXTENDED-BODY-PART-TYPE
        PARAMETERS    EncryptedParameters IDENTIFIED BY id-ep-encrypted
        DATA          EncryptedData
        ::= id-et-encrypted

-- Extended Message body part

message-body-part EXTENDED-BODY-PART-TYPE
        PARAMETERS    MessageParameters IDENTIFIED BY id-ep-message
        DATA          MessageData
        ::= id-et-message

-- Extended Mixed-mode body part

mixed-mode-body-part EXTENDED-BODY-PART-TYPE
        DATA MixedModeBodyPart
        ::= id-et-mixed-mode

-- Extended Bilaterally Defined body part

bilaterally-defined-body-part EXTENDED-BODY-PART-TYPE
        DATA BilaterallyDefinedBodyPart
        ::= id-et-bilaterally-defined

-- Extended Nationally Defined body part

nationally-defined-body-part EXTENDED-BODY-PART-TYPE
        DATA NationallyDefinedBodyPart
        ::= id-et-nationally-defined

END -- of IPMSExtendedBodyPartTypes
```

I.2 *General text*

```
IPMSExtendedBodyPartTypes2 { iso standard motis(10021) ipms(7) modules(0) extended-body-part-types-2(1) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS

        -- IPMS Information Objects  (from Annex E)

        EXTENDED-BODY-PART-TYPE
            ----
            FROM IPMSInformationObjects { joint-iso-ccitt-mhs-motis(6) ipms(1) modules(0)
        information-objects(2) }

        -- IPMS Object Identifiers (from Annex D)

        id-ep-general-text, id-et-general-text
            ----
            FROM IPMSObjectIdentifiers { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)
        object-identifiers(0) };

-- General Text body part

general-text-body-part EXTENDED-BODY-PART-TYPE
        PARAMETERS   GeneralTextParameters IDENTIFIED BY id-ep-general-text
        DATA         GeneralTextData
        ::= id-et-general-text

GeneralTextParameters ::= SET OF CharacterSetRegistration

GeneralTextData ::= GeneralString

CharacterSetRegistration ::= INTEGER (1. . 32767)

END -- of IPMSExtendedBodyPartTypes2
```

I.3 *File Transfer*

```
IPMSFileTransferBodyPartType { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)
                                    file-transfer-body-part-type(9) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

--Prologue

-- Exports everything.

IMPORTS

        -- FTAM Attribute Types (from ISO 8571-4: 1988)

            Attribute-Extensions, Concurrency-Access, Contents-Type-Attribute, Date-and-Time-Attribute, Legal-
            Qualification-Attribute, Object-Size-Attribute, Pathname, Permitted-Actions-Attribute, Private-Use-
            Attribute, User-Identity-Attribute
            ----
            FROM ISO8571-FTAM

        -- ISO 8571-FTAM is { iso standard 8571 abstract-syntax 2) ftam-pei (1) }

        -- IPMS Information Objects (from Annex E)

        EXTENDED-BODY-PART-TYPE, ExtensionsField
            ----
            FROM IPMSInformationObjects { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) information-
        objects(2) }
```

```
        -- IPMS Object Identifiers (from Annex D)

        id-ep-file-transfer, id-et-file-transfer
            ----
            FROM IPMSObjectIdentifiers { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)    object-identifiers(0) }

        -- MTS Abstract Service [from Recommendation X.411 (1992)]

        ORName
            ----
            FROM MTSAbstractService { joint-iso-ccitt mhs-motis(6) mts(3) modules(0)
        mts-abstract-service(1) };

-- File Transfer body part

file-transfer-body-part EXTENDED-BODY-PART-TYPE
        PARAMETERS   FileTransferParameters IDENTIFIED BY id-ep-file-transfer
        DATA         FileTransferData
        ::= id-et-file-transfer

FileTransferParameters ::= SEQUENCE {
        related-stored-file     [0]   RelatedStoredFile OPTIONAL,
        contents-type           [1]   ContentsTypeParameter
                                      DEFAULT { iso standard 8571 document-type (5) unstructured-binary (3) },
        environment             [2]    EnvironmentParameter OPTIONAL,
        compression             [3]   CompressionParameter OPTIONAL,
        file-attributes         [4]   FileAttributes OPTIONAL,
        extensions              [5]   ExtensionsField DEFAULT { } }

FileTransferData ::= SEQUENCE OF EXTERNAL
-- This conveys a sequence of data values representing file contents;
--The rules for generating this sequence are implied by the value of the contents-type parameter

RelatedStoredFile ::= SET OF SEQUENCE {
        file-identifier   FileIdentifier,
        relationship      Relationship DEFAULT unspecified }

FileIdentifier ::= CHOICE {
        pathname-and-version    [0]   PathnameandVersion,
        cross-reference         [1]   CrossReference }

PathnameandVersion ::= SEQUENCE {
        pathname      [0]   Pathname-Attribute,
        file-version  [1]   GraphicString OPTIONAL }

Pathname-Attribute ::= CHOICE {
        incomplete-pathname     [0]   Pathname,
        complete-pathname       [23]  Pathname }

CrossReference ::= SEQUENCE {
        application-crossreference    [0]   OCTET STRING,
        message-reference             [1]   MessageReference OPTIONAL,
        body-part-reference           [2]   INTEGER OPTIONAL }

MessageReference ::= SET {
        user                          [0]   ORName,
        -- Defined in 8.5.5 of CCITT Rec. X.411 | ISO/IEC 10021-4
        user-relative-identifier      [1]   PrintableString }

Relationship ::= CHOICE {
        explicit-relationship         [0]   ExplicitRelationship,
        descriptive-relationship      [1]   GraphicString }

ExplicitRelationship ::= ENUMERATED {
        unspecified     (0),
        new-file        (1),
        replacement     (2),
        extension       (3) }
```

```
ContentsTypeParameter ::= Contents-Type-Attribute

Contents-Type-Attribute ::= CHOICE {
        document-type    [0]   SEQUENCE {
            document-type-name              Document-Type-Name,
            parameter    [0]   ANY OPTIONAL },
          -- The actual types to be used for values of the parameter field
          -- are defined in the named document type
        constraint-set-and-abstract-syntax    (1)   SEQUENCE {
            constraint-set-name             Constraint-set-Name,
            abstract-syntax-name            Abstract-Syntax-Name } }

Constraint-Set-Name ::= OBJECT IDENTIFIER

Document-Type-Name ::= OBJECT IDENTIFIER

Abstract-Syntax-Name ::= OBJECT IDENTIFIER

EnvironmentParameter ::= SEQUENCE {
        application-reference    [0]   GeneralIdentifier OPTIONAL,
        machine                  [1]   GeneralIdentifier OPTIONAL,
        operating-system         [2]   OBJECT IDENTIFIER OPTIONAL,
        user-visible-string      [3]   SEQUENCE OF GraphicString OPTIONAL }

GeneralIdentifier ::= CHOICE {
        registered-identifier    [0]   OBJECT IDENTIFIER,
        descriptive-identifier   [1]   SEQUENCE OF GraphicString }

CompressionParameter ::= SEQUENCE {
        compression-algorithm-id      [0]   OBJECT IDENTIFIER,
        compression-algorithm-param   [1]   ANY DEFINED BY compression-algorithm-id }

FileAttributes ::= SEQUENCE {
        pathname                              Pathname-Attribute OPTIONAL,
        permitted-actions                [1]  Permitted-Actions-Attribute OPTIONAL,
        storage-account                  [3]  Account-Attribute OPTIONAL,
        date-and-time-of-creation        [4]  Date-and-Time-Attribute OPTIONAL,
        date-and-time-of-last-modification  [5]  Date-and-Time-Attribute OPTIONAL,
        date-and-time-of-last-read-access   [6]  Date-and-Time-Attribute OPTIONAL,
        identity-of-creator              [8]  User-Identity-Attribute OPTIONAL,
        identity-of-last-modifier        [9]  User-Identity-Attribute OPTIONAL,
        identity-of-last-reader          [10] User-Identity-Attribute OPTIONAL,
        object-size                      [13] Object-Size-Attribute OPTIONAL,
        future-object-size               [14] Object-Size-Attribute OPTIONAL,
        access-control                   [15] Access-Control-Attribute OPTIONAL,
        legal-qualifications             [16] Legal-Qualification-Attribute OPTIONAL,
        private-use                      [17] Private-Use-Attribute OPTIONAL,
        attribute-extensions             [22] Attribute-Extensions OPTIONAL }

Account-Attribute ::= CHOICE {
        no-value-available          [0]   NULL,
          -- Indicated partial support of this attribute
        actual-values               Account }

Account ::= GraphicString

User-Identity-Attribute ::= CHOICE {
        no-value-available          [0]   NULL,
          -- Indicates partial support of this attribute.
        actual-values               User-Identity }

Access-Control-Attribute ::= CHOICE {
        no-value-available          [0]   NULL,
          -- Indicates partial support of this attribute.
        actual-values               [1]   SET OF Access-Control-Element }
        -- The semantics of this attribute are described in ISO 8571-2
```

```
Access-Control-Element ::= SEQUENCE {
        action-list             [0]  Access-Request
        concurrency-access      [1]  Concurrency-Access OPTIONAL,
        identity                [2]  User-Identity-OPTIONAL,
        passwords               [3]  Access-Passwords OPTIONAL,
        location                [4]  Application-Entity-Title OPTIONAL }

Access-Request ::= BIT STRING {
        read                        (0),
        insert                      (1),
        replace                     (2),
        extend                      (3),
        erase                       (4),
        read-attribute              (5),
        change-attribute            (6),
        delete-object               (7) }

Access-Passwords ::= SEQUENCE {
        read-password           [0]  Password,
        insert-password         [1]  Password,
        replace-password        [2]  Password,
        extend-password         [3]  Password,
        erase-password          [4]  Password,
        read-attribute-password [5]  Password,
        change-attribute-password [6] Password,
        delete-password         [7]  Password,
        pass-passwords          [8]  Pass-Passwords,
        link-password           [9]  Password }

Application-Entity-Title ::= SEQUENCE {
        ap-title        ANY,
        ae-qualifier    ANY }

Password ::= CHOICE { GraphicString, OCTET STRING }

Pass-Passwords ::= SEQUENCE OF Password

User-Identity ::= GraphicString

END -- of IPMSFileTransferBodyPartType
```

I.4 *Voice*

```
IPMSExtendedVoiceBodyPartType { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)
                                extended-voice-body-part-type(11) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS

        -- VMGS Information Objects [from Recommendation X.440 (1992)]

        VoiceData, VoiceParameters,
            ----
            FROM VMGSInformationObjects { joint-iso-ccitt mhs-motis(6) vmgs(8) modules(9)
        information-objects(2) }

        -- IPMS Information Objects (from Annex E)

        EXTENDED-BODY-PART-TYPE
            ----
            FROM IPMSInformationObjects { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)
        information-objects(2) }
```

```
    -- IPMS Object Identifiers

    id-ep-vmg-body-part, id-et-vmg-body-part
        ----
        FROM IPMSObjectIdentifiers { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)
    object-identifiers(0) };

-- Extended Voice body part

voice-body-part EXTENDED-BODY-PART-TYPE
        PARAMETERS   VoiceParameters IDENTIFIED BY id-ep-vmg-body-part
        DATA         VoiceData
        ::= id-et-vmg-body-part

END -- of IPMSExtendedVoiceBodyPartType
```

ANNEX J

(to Recommendation X.420)

Reference definition of message store attributes

(This annex forms an integral part of this Recommendation)

This annex, a supplement to Annex C, defines for reference purposes the MS attributes specific to Interpersonal Messaging. It uses the ATTRIBUTE macro of CCITT Rec. X.501 | ISO/IEC 9594-2.

```
IPMSMessageStoreAttributes { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) message-store-attributes(8) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS

    -- IPMS Heading Extensions (from Annex H)

    AutoSubmitted, IncompleteCopy, Language
        ----
        FROM IPMSHeadingExtensions { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)
    heading-extensions(6) }

    -- IPMS Information Objects (from Annex E)

    AcknowledgmentModeField, AuthorizingUsersSubfield, AutoForwardCommentField, AutoForwardedField,
    BilaterallyDefinedBodyPart, BlindCopyRecipientsSubfield, Body, ConversionEITsField,
    CopyRecipientsSubfield, DiscardReasonField, EncryptedBodyPart, EncryptedData, EncryptedParameters,
    ExpiryTimeField, ExternallyDefinedParameters, G3FacsimileBodyPart, G3FacsimileData,
    G3FacsimileParameters, G4Class1BodyPart, Heading, IA5TextBodyPart, IA5TextData, IA5TextParameters,
    ImportanceField, IPMPreferredRecipientField, IPNOriginatorField, MessageBodyPart, MessageData,
    MessageParameters, MixedModeBodyPart, NationallyDefinedBodyPart, NonReceiptReasonField,
    ObsoletedIPMsSubfield, ORDescriptor, OriginatorField, PrimaryRecipientsSubfield, ReceiptTimeField,
    RelatedIPMsSubfield, RepliedToIPMField, ReplyRecipientsSubfield, ReplyTimeField, ReturnedIPMField,
    SensitivityField, SubjectField, SubjectIPMField, SupplReceiptInfoField, TeletexBodyPart, TeletexData,
    TeletexParameters, ThisIPMField, VideotexBodyPart, VideotexData, VideotexParameters
        ----
        FROM IPMSInformationObjects { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) information-
    objects(2) }
```

```
        -- IPMS Object Identifiers (from Annex D)

        id-bat-bilaterally-defined-body-parts, id-bat-body, id-bat-encrypted-body-parts, id-bat-encrypted-data, id-
        bat-encrypted-parameters, id-bat-extended-body-part-types, id-bat-g3-facsimile-body-parts, id-bat-g3-
        facsimile-data, id-bat-g3-facsimile-parameters, id-bat-g4-class1-body-parts, id-bat-ia5-text-body-parts, id-
        bat-ia5-text-data, id-bat-ia5-text-parameters, id-bat-message-body-parts, id-bat-message-data, id-bat-
        message-parameters, id-bat-mixed-mode-body-parts, id-bat-nationally-defined-body-parts, id-bat-teletex-
        body-parts, id-bat-teletex-data, id-bat-teletex-parameters, id-bat-videotex-body-parts, id-bat-videotex-data,
        id-bat-videotex-parameters, id-hat-authorizing-users, id-hat-auto-forwarded, id-hat-auto-submitted, id-hat-
        blind-copy-recipients, id-hat-copy-recipients, id-hat-expiry-time, id-hat-heading, id-hat-importance, id-hat-
        incomplete-copy, id-hat-languages, id-hat-nrn-requestors, id-hat-obsoleted-IPMs, id-hat-originator, id-hat-
        primary-recipients, id-hat-related-IPMs, id-hat-replied-to-IPM, id-hat-reply-recipients, id-hat-reply-
        requestors, id-hat-reply-time, id-hat-rn-requestors, id-hat-sensitivity, id-hat-subject, id-hat-this-ipm, id-nat-
        acknowledgment-mode, id-nat-auto-forward-comment, id-nat-conversion-eits, id-nat-discard-reason, id-
        nat-ipm-preferred-recipient, id-nat-ipn-originator, id-nat-non-receipt-reason, id-nat-receipt-time, id-nat-
        returned-ipm, id-nat-subject-ipm, id-nat-suppl-receipt-info, id-sat-ipm-entry-type, id-sat-ipm-synopsis
            ----
            FROM IPMSObjectIdentifiers { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)    object-identifiers(0) }

        -- MS Abstract Service (from Recommendation X.413)

        MS-EIT, SequenceNumber
            ----
            FROM MSAbstractService { joint-iso-ccitt mhs-motis(6) ms(4) modules(0) abstract-service(1) }

        -- MTS Abstract Service  (from Recommendation X.411)

        EncodedInformationTypes
            ----
            FROM MTSAbstractService { joint-iso-ccitt mhs-motis(6) mts(3) modules(0)
        mts-abstract-service(1) }

        -- Directory Information Framework [from Annex C of Recommendation X.501 (1988)]

        ATTRIBUTE
            ----
            FROM InformationFramework { joint-iso-ccitt ds(5) modules(1) informationFramework(1) };

Time ::= UTCTime

-- SUMMARY ATTRIBUTES

-- IPM entry type

ipm-entry-type ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX IPMEntryType
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-sat-ipm-entry-type

IPMEntryType ::= ENUMERATED {
        ipm    (0),
        rn     (1),
        nrn    (2) }

-- IPM synopsis

ipm-synopsis ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX IPMSynopsis
        SINGLE VALUE
        ::= id-sat-ipm-synopsis

IPMSynopsis ::= SEQUENCE OF BodyPartSynopsis

BodyPartSynopsis ::= CHOICE {
        message          [0]   MessageBodyPartSynopsis,
        non-message      [1]    NonMessageBodyPartSynopsis }

MessageBodyPartSynopsis ::= SEQUENCE {
        number           [0]   SequenceNumber,
        synopsis         [1]   IPMSynopsis }
```

```
NonMessageBodyPartSynopsis ::= SEQUENCE {
        type            [0]   OBJECT IDENTIFIER,
        parameters      [1]   ExternallyDefinedParameters,
        size            [2]   INTEGER,
        processed       [3]   BOOLEAN DEFAULT FALSE }

-- HEADING ATTRIBUTES

-- Heading

heading ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX Heading
        SINGLE VALUE
        ::= id-hat-heading

-- Heading analyses

rn-requestors ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX ORDescriptor
        MATCHES FOR EQUALITY
        MULTI VALUE
        ::= id-hat-rn-requestors

nrn-requestors ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX ORDescriptor
        MATCHES FOR EQUALITY
        MULTI VALUE
        ::= id-hat-nrn-requestors

reply-requestors ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX ORDescriptor
        MATCHES FOR EQUALITY
        MULTI VALUE
        ::= id-hat-reply-requestors

-- Heading fields

this-ipm ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX ThisIPMField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-hat-this-ipm

originator ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX OriginatorField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-hat-originator

replied-to-IPM ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX RepliedToIPMField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-hat-replied-to-IPM

subject ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX SubjectField
        MATCHES FOR EQUALITY SUBSTRINGS
        SINGLE VALUE
        ::= id-hat-subject

expiry-time ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX ExpiryTimeField
        MATCHES FOR EQUALITY ORDERING
        SINGLE VALUE
        ::= id-hat-expiry-time
```

reply-time ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ReplyTimeField
MATCHES FOR EQUALITY ORDERING
SINGLE VALUE
::= id-hat-reply-time

importance ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ImportanceField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-importance

sensitivity ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SensitivityField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-sensitivity

auto-forwarded ATTRIBUTE
WITH ATTRIBUTE-SYNTAX AutoForwardedField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-auto-forwarded

-- Heading sub-fields

authorizing-users ATTRIBUTE
WITH ATTRIBUTE-SYNTAX AuthorizingUsersSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-authorizing-users

primary-recipients ATTRIBUTE
WITH ATTRIBUTE-SYNTAX PrimaryRecipientsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-primary-recipients

copy-recipients ATTRIBUTE
WITH ATTRIBUTE-SYNTAX CopyRecipientsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-copy-recipients

blind-copy-recipients ATTRIBUTE
WITH ATTRIBUTE-SYNTAX BlindCopyRecipientsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-blind-copy-recipients

obsoleted-IPMs ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ObsoletedIPMsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-obsoleted-IPMs

related-IPMs ATTRIBUTE
WITH ATTRIBUTE-SYNTAX RelatedIPMsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-related-IPMs

reply-recipients ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ReplyRecipientsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-reply-recipients

-- Heading extensions

incomplete-copy ATTRIBUTE
WITH ATTRIBUTE-SYNTAX IncompleteCopy
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-incomplete-copy

languages ATTRIBUTE
WITH ATTRIBUTE-SYNTAX Language
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-languages

auto-submitted ATTRIBUTE
WITH ATTRIBUTE-SYNTAX AutoSubmitted
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-auto-submitted

-- BODY ATTRIBUTES

-- Body

body ATTRIBUTE
WITH ATTRIBUTE-SYNTAX Body
SINGLE VALUE
::= id-bat-body

-- Basic body parts

ia5-text-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX IA5TextBodyPart
MULTI VALUE
::= id-bat-ia5-text-body-parts

g3-facsimile-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX G3FacsimileBodyPart
MULTI VALUE
::= id-bat-g3-facsimile-body-parts

g4-class1-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX G4Class1BodyPart
MULTI VALUE
::= id-bat-g4-class1-body-parts

teletex-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX TeletexBodyPart
MULTI VALUE
::= id-bat-teletex-body-parts

videotex-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VideotexBodyPart
MULTI VALUE
::= id-bat-videotex-body-parts

encrypted-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX EncryptedBodyPart
MULTI VALUE
::= id-bat-encrypted-body-parts

message-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SequenceNumber
MULTI VALUE
::= id-bat-message-body-parts

mixed-mode-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX MixedModeBodyPart
MULTI VALUE
::= id-bat-mixed-mode-body-parts

bilaterally-defined-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX BilaterallyDefinedBodyPart
MULTI VALUE
::= id-bat-bilaterally-defined-body-parts

nationally-defined-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX NationallyDefinedBodyPart
MULTI VALUE
::= id-bat-nationally-defined-body-parts

-- Basic body part parameters components

ia5-text-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX IA5TextParameters
MULTI VALUE
::= id-bat-ia5-text-parameters

g3-facsimile-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX G3FacsimileParameters
MULTI VALUE
::= id-bat-g3-facsimile-parameters

teletex-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX TeletexParameters
MULTI VALUE
::= id-bat-teletex-parameters

videotex-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VideotexParameters
MULTI VALUE
::= id-bat-videotex-parameters

encrypted-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX EncryptedParameters
MULTI VALUE
::= id-bat-encrypted-parameters

message-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX MessageParameters
MULTI VALUE
::= id-bat-message-parameters

-- Basic body part data components

ia5-text-data ATTRIBUTE
WITH ATTRIBUTE-SYNTAX IA5TextData
MULTI VALUE
::= id-bat-ia5-text-data

g3-facsimile-data ATTRIBUTE
WITH ATTRIBUTE-SYNTAX G3FacsimileData
MULTI VALUE
::= id-bat-g3-facsimile-data

teletex-data ATTRIBUTE
WITH ATTRIBUTE-SYNTAX TeletexData
MULTI VALUE
::= id-bat-teletex-data

videotex-data ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VideotexData
MULTI VALUE
::= id-bat-videotex-data

encrypted-data ATTRIBUTE
WITH ATTRIBUTE-SYNTAX EncryptedData
MULTI VALUE
::= id-bat-encrypted-data

```
message-data ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX MessageData
        MULTI VALUE
        ::= id-bat-message-data

-- Extended body part types

extended-body-part-types ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
        MATCHES FOR EQUALITY
        MULTI VALUE
        ::= id-bat-extended-body-part-types

-- Extended body parts

-- (These attributes cannot be enumerated. See clause C.3.6.)

-- NOTIFICATION ATTRIBUTES

-- Common fields

subject-ipm ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX SubjectIPMField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-nat-subject-ipm

ipn-originator ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX IPNOriginatorField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-nat-ipn-originator

ipm-preferred-recipient ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX IPMPreferredRecipientField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-nat-ipm-preferred-recipient

conversion-eits ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX MS-EIT
        MATCHES FOR EQUALITY
        MULTI VALUE
        ::= id-nat-conversion-eits

-- Non-receipt fields

non-receipt-reason ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX NonReceiptReasonField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-nat-non-receipt-reason

discard-reason ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX DiscardReasonField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-nat-discard-reason

auto-forward-comment ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX AutoForwardCommentField
        MATCHES FOR EQUALITY SUBSTRINGS
        SINGLE VALUE
        ::= id-nat-auto-forward-comment

returned-ipm ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX ReturnedIPMField
        SINGLE VALUE
        ::= id-nat-returned-ipm
```

```
-- Receipt fields

receipt-time ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX ReceiptTimeField
        MATCHES FOR EQUALITY ORDERING
        SINGLE VALUE
        ::= id-nat-receipt-time

acknowledgment-mode ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX AcknowledgmentModeField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-nat-acknowledgment-mode

suppl-receipt-info ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX SupplReceiptInfoField
        MATCHES FOR EQUALITY SUBSTRINGS
        SINGLE VALUE
        ::= id-nat-suppl-receipt-info

END -- of IPMSMessageStoreAttributes
```

ANNEX K

(to Recommendation X.420)

Reference definition of upper bounds

(This annex forms an integral part of this Recommendation)

This annex defines for reference purposes the upper bounds of various variable-length information items whose abstract syntaxes are defined in the ASN.1 modules of prior annexes.

```
IPMSUpperBounds { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) upper-bounds(10) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS -- nothing -- ;

-- Upper bounds

ub-auto-forward-comment   INTEGER ::= 256

ub-free-form-name         INTEGER ::= 64

ub-ipm-identifier-suffix  INTEGER ::= 2

ub-local-ipm-identifier   INTEGER ::= 64

ub-subject-field          INTEGER ::= 128

ub-telephone-number       INTEGER ::= 32

END -- of IPMSUpperBounds
```

ANNEX L

(to Recommendation X.420)

Support of the Interpersonal Messaging Service

(This annex forms an integral part of this Recommendation)

The Interpersonal Messaging Service which the IPMS provides to users is described in Rec. F.420 and defined, in non-technical terms, in CCITT Rec. X.400|ISO/IEC 10021-1. The service comprises a number of elements of service (**IPM EOS**), each representing one aspect of the service, and each defined in one or two paragraphs of prose. The present annex indicates in detail how the present, more technical specification realizes each IPM EOS. Equivalently, it identifies the aspects of the specification a UA, e.g. must implement for it to be said to support a particular IPM EOS.

Associated with each IPM EOS are one or more information items that may appear as IPM components. The information item associated with the Sensitivity Indication IPM EOS, e.g. is the Sensitivity heading field. A UA, TLMA, or AU shall be said to support a particular IPM EOS upon origination or reception if, and only if, it supports upon origination or reception (see 22.1) the information items associated with that IPM EOS.

Note 1 – The task of realizing an IPM EOS may fall, in principle, upon any of the secondary objects that result from the refinement of the IPMS. In the present context, however, it is assumed that the MTS and every MS, by virtue of their application-independence, support every IPM EOS, and that they do so without having made special provision for any of them.

Note 2 – As described in clause 14, a UA makes available to its user many of the capabilities that its MS offers. These capabilities realize the elements of the Message Retrieval service which is defined in CCITT Rec. X.400 | ISO/IEC 10021-1. The correspondence between the elements of that service and the associated technical capabilities is given in CCITT Rec. X.413 | ISO/IEC 10021-5.

Note 3 – As described in clause 14, a UA makes available to its user many of the capabilities that the MTS offers. These capabilities realize the elements of the Message Transfer service which is defined in CCITT Rec. X.400 | ISO/IEC 10021-1. The correspondence between the elements of that service and the associated technical capabilities is given in Recommendation X.411 | ISO/IEC 10021-4.

L.1 *Support of recipient specifier components*

Some IPM EOS are realized by means of recipient specifier components. The IPM EOS in this category are listed in the first column of Table L/X.420. The second and third columns identify the recipient specifier component, and the particular value of that component, that are the information items associated with each listed IPM EOS.

L.2 *Support of heading fields*

Some IPM EOS are realized by means of heading fields. The IPM EOS in this category are listed in the first column of Table L-2/X.420. The second column identifies the heading fields that are the information items associated with each listed IPM EOS. In the case of the Extensions field, the second column also identifies, in parentheses, the relevant heading extension.

L.3 *Support of Body aspects*

Some IPM EOS are realized by means of aspects of the Body. The IPM EOS in this category are listed in the first column of Table L-3/X.420. The second column identifies the Body aspect that is the information item associated with each listed IPM EOS.

TABLE L-1/X.420

Support of recipient specifier components

Element of service	Recipient specifier component	Value
Non-receipt notification request	Notification-requests	nrn
Receipt notification request indication	Notification-requests	rn
Reply request indication (see also Table L-2/X.420)	Reply-requests	true

Note 1 – Recipient specifiers appear as sub-fields of the Primary Recipients, Copy Recipients and Blind Copy Recipients heading fields.

Note 2 – Every IPM EOS except Reply Request Indication falls into exactly one category. The Reply Request Indication IPM EOS falls into two categories, as indicated in the table.

TABLE L-2/X.420

Support of heading fields

Element of service	Heading field
Authorizing Users Indication	Authorizing Users
Auto-forwarded Indication	Auto-forwarded
Auto-submitted Indication	Extensions (Auto-submitted)
Blind Copy Recipient Indication	Blind Copy Recipients
Cross-referencing Indication	Related IPMs
Expiry Date Indication	Expiry Time
Importance Indication	Importance
IP-message Identification	This IPM
Incomplete Copy Indication	Extensions (Incomplete Copy)
Language Indication	Extensions (Languages)
Obsoleting Indication	Obsoleted IPMs
Originator Indication	Originator
Primary and Copy Recipients Indication	Primary Recipients Copy Recipients
Reply Request Indication (see also Table L-1/X.420)	Reply Time Reply Recipients
Replying IP-message Indication	Replied-to IPM
Sensitivity Indication	Sensitivity
Subject Indication	Subject

Note – Every IPM EOS except Reply Request Indication falls into exactly one category. The Reply Request Indication IPM EOS falls into two categories, as indicated in the table.

TABLE L-3/X.420

Support of Body Aspects

Element of service	Body aspect
Body Part Encryption Indication	Encrypted body part
Forwarded IP-message Indication	Message body part
Multi-part Body	Body with two or more parts
Typed Body	Body (itself)

Note – Support of the Typed Body IPM EOS is intrinsic to any implementation of any secondary object.

ANNEX M

(to Recommendation X.420)

Differences between CCITT X.420 Recommendation and the corresponding ISO/IEC International Standard

(This annex does not form an integral part of this Recommendation)

This annex lists all but the purely stylistic differences between this Recommendation and the corresponding ISO/IEC International Standard.

The following are the differences that exist:

– The upper bounds of Annex K are an integral part of this Recommendation but are not a part of the corresponding ISO/IEC International Standard.

ANNEX N

(to Recommendation X.420)

Summary of changes to the 1984 version of Recommendation X.420

(This annex does not form an integral part of this Recommendation)

Editorially, this Recommendation differs substantially from Recommendation X.420 (1984). Technically, however, the differences are few. The present annex lists the technical changes. It is intended as an aid to an implementor of Recommendation X.420 (1984), enabling him to see at a glance how his implementation might be affected by the 1992 Recommendation.

The following, and only the following, substantive changes relevant to interworking between 1984 and 1992 UAs, MSs, TLMAs, and AUs are embodied in the present Recommendation. All but the first are changes to the format of the information objects now defined in the ASN.1 module, IPMSInformationObjects:

a) The content type assigned to P2 has changed. Formerly identified by the Integer 2, P2 now is identified by either the Integer 2 or 22, depending upon the functionality employed in a particular instance of communication by means of the MTS (see 20.2).

b) The omission of the user member of IPMIdentifier is now denigrated.

c) The extensions member has been added to Heading. Its grade is optional.

d) The Telex and Simple Formattable Document body part types have been abandoned. (The former had been identified but not defined.)

e) The syntax member has been added to VideotexParameters. Its grade is optional.

f) The presence of the delivery-time member of MessageParameters in the absence of its delivery-envelope member, or vice versa, is now denigrated.)

g) The bilaterally-defined and externally-defined alternatives have been added to BodyPart.

h) The General Text extended body part has been defined.

i) The following protocol elements, defined in CCITT Rec. X.411 | ISO/IEC 10021-4 and incorporated in protocol elements of this Recommendation by reference, have changed:

   i) ORName;

   ii) MessageDeliveryEnvelope;

   iii) EncodedInformationTypes;

   iv) SupplementaryInformation.

j) Specifying a value of zero length of any of the following data types is now denigrated:

   i) LocalIPMIdentifier;

   ii) FreeFormName;

   iii) TelephoneNumber;

   iv) SubjectField;

   v) AutoForwardComment.

k) Upper bounds have been imposed upon certain variable-length protocol elements.

Note – The upper bounds imposed are those found in § 4.3 of Version 6 of the *X.400-Series Implementor's Guide applicable to the 1984 X.400-Series Recommendations.*

ANNEX O

(to Recommendation X.420)

Summary of changes to the 1988 version of Recommendation X.420

(This annex does not form an integral part of this Recommendation)

Editorially, this Recommendation differs slightly from CCITT Recommendation X.420 (1988). The technical differences are as follows:

a) the addition of a general text body part (I.2 of Annex I);

b) the addition of a File Transfer body part (I.3 of Annex I);

c) the addition of a Voice body part (I.4 of Annex I).

Note – The additional body parts are incorporated by use of object identifiers. Hence, interoperating between 1988 and 1992 X.420 systems should not be adversely affected.

ANNEX P

(to Recommendation X.420)

Index

(This annex does not form an integral part of this Recommendation)

This annex indexes this Recommendation. It gives the number(s) of the page(s) on which each item in each of several categories is defined. Its coverage of each category is exhaustive.

This annex indexes terms (if any) in the following categories:

a) abbreviations;

b) terms;

c) information objects;

d) ASN.1 modules;

e) ASN.1 macros;

f) ASN.1 types;

g) ASN.1 values.

	Page
Abbreviations	
IPM EOS	109
IPME	27
IPN	22
P2	i
PDAU	40
TLXAU	40
Terms	
auto-acknowledgment	34
auto-forwarding	35
Auto-forward action	51
Basic body part	53
blind	13
Encrypted body part	53
Externally Defined body part	52
Forwarded) Message body part	52
forwarded IPM	20
forwarding	20
forwarding IPM	20
Interpersonal Messaging	3
Interpersonal Messaging Environment	27
IPMS Abstract Service	29
management port	29
manual forwarding	51
Multiple body parts	52
O/R descriptor	11
origination port	29
primary object	28
primary port	29
reception port	29
secondary port	40
state variable	42

	Page
Information objects	
Administration	36
Auto-acknowledge-IPMs	35
Auto-acknowledge-suppl-receipt-info	35
Auto-discard-expired-IPMs	34
Auto-discard-obsolete-IPMs	34
Auto-forward Comment	26,47,49
Auto-submitted	56
Bilaterally Defined	21
Blind Copy Recipients	13
Body	74
Body part types	16
Change Auto-acknowledgment	34
Change Auto-forwarding	35
Common Fields	23
Content	31,33,44,45,46,49
Content length	52
Content-correlator	43,48
Content-identifier	43,48
Content-type	43,48
Conversion EITs	47
Copy Recipients	13
Data	16
Deferred-delivery-time	44,49
Delivery	36
Delivery-envelope	20
Delivery-time	20
Discard Reason	25,47,49
Encoded Information types	52
Envelope	31,33,43,45,46,48
Extended Body Part Types	77
Extension-parameters	64

	Page
Extensions	49
File Transfer	58
Formal-name	12
Free-form-name	12
G4 Class 1	18
General Text	58
Heading	72
IA5 Text	17
Incomplete Copy	56
interpersonal notification	22
IPM Entry Type	69
IPM Preferred Recipient	47
IPM Synopsis	71
IPN Originator	24,47
Languages	56
Message	20
Message Transfer System	40
Mixed-mode	21
Non-basic-parameters	19
Non-receipt Reason	47,49
NRN Extensions	26
Number	71
Number-of-pages	19
Obsoleted IPMs	14
Original-encoded-information-types	43,48
Originate IPM	31
Originate RN	31
Originator-name	43,48
Other capabilities	36
Other-parameters	64
Parameters	16,71
Per-message-indicators	44,49
Per-recipient-fields	49
Physical Delivery Access Unit	40
Priority	49
Processed	71
Receive IPM	33
Receive RN	33
Recipient Improperly Specified	36
Repertoire	17
Replied-to IPM	14
Retrieval	36
Returned IPM	26,47,49
Size	71
Subject IPM	24,47
Submission	36
Submission-identifier	31
Submission-time	31
Submission-identifier	44
Submission-time	44
Synopsis	71
Telephone-number	12
Teletex	18
Telex-compatible	19
telex access unit	40
Type	16,71
Undelivered-object	33
Undelivered-object	45
Value	16
Voice	17,64
Voice-encoding-type	64
Voice-message-duration	64

ASN.1 modules

IPMSAbstractService	91
IPMSExtendedBodyPartTypes	95
IPMSExtendedBodyPartTypes2	97
IPMSExtendedVoiceBodyPartType	100
IPMSFileTransferBodyPartType	97
IPMSFunctionalObjects	89
IPMSHeadingExtensions	94
IPMSInformationObjects	83
IPMSMessageStoreAttributes	101
IPMSObjectIdentifiers	79
IPMSObjectIdentifiers2	82
IPMSUpperBounds	108

ASN.1 macros

EXTENDED-BODY-PART-TYPE	22,87
IPMS-EXTENSION	16,83

ASN.1 types

AcknowledgmentModeField	88
AuthorizingUsersField	84
AuthorizingUsersSubfield	84
AutoForwardComment	26,88
AutoForwardCommentField	26,88
AutoForwardedField	85
AutoSubmitted	56,95
BilaterallyDefinedBodyPart	21,87
BlindCopyRecipientsField	13,85
BlindCopyRecipientsSubfield	13,85
Body	85
BodyPart	16,85
BodyPartSynopsis	71,102
ChangeAutoAcknowledgment	34,93
ChangeAutoDiscard	93
ChangeAutoForwarding	35,94
CharacterSetRegistration	58,97
CommonFields	23,87
CompressionParameter	61,99
ContentsTypeParameter	60,99

	Page
ConversionEITsField	88
CopyRecipientsField	13,85
CopyRecipientsSubfield	13,85
CrossReference	59,98
DiscardReasonField	25,88
EncryptedBodyPart	87
EncryptedData	87
EncryptedParameters	87
EnvironmentParameter	61,99
ExpiryTimeField	85
ExplicitRelationship	59,98
ExtensionsField	85
ExternallyDefinedBodyPart	87
ExternallyDefinedData	87
ExternallyDefinedParameters	87
File Attributes	62,99
FileIdentifier	59,98
FileTransferData	59,98
FileTransferParameters	59,98
ForwardedInfo	51,88
FreeFormName	12,84
G3FacsimileBodyPart	86
G3FacsimileData	86
G3FacsimileParameters	86
G4Class1BodyPart	18,86
GeneralIdentifier	61,99
GeneralTextData	58,97
GeneralTextParameters	58,97
Heading	12,84
IA5TextBodyPart	17,86
IA5TextData	17,86
IA5TextParameters	17,86
ID	79,82
ImportanceField	85
IncompleteCopy	94
InformationObject	83
IPM	83
IPMEntryType	69,102
IPMIdentifier	10,84
IPMPreferredRecipientField	88
IPMSExtension	83
IPMSynopsis	71,102
IPN	23,87
IPNOriginatorField	24,88
Language	56,95
LocalIPMIdentifier	84
MessageBodyPart	87
MessageBodyPartSynopsis	71,102
MessageData	87
MessageParameters	87
MessageReference	59,98
MixedModeBodyPart	21,86
NationallyDefinedBodyPart	87
NonMessageBodyPartSynopsis	71,103
NonReceiptFields	88
NonReceiptReasonField	88
NotificationExtensionsField	88
NotificationRequests	84
NRN	87
NRNExtensionsField	26,88
ObsoletedIPMsField	14,85
ObsoletedIPMsSubfield	14,85
ON	87
ORDescriptor	11,84
OriginateIPM	31,93
OriginateProbe	92
OriginateRN	31,93
OriginatorField	84
OtherNotificationTypeFields	88
PathnameandVersion	59,98
PrimaryRecipientsField	84
PrimaryRecipientsSubfield	84
ReceiptFields	88
ReceiptTimeField	88
ReceiveIPM	33,93
ReceiveNRN	93
ReceiveReport	93
ReceiveRN	33,93
RecipientExtensionsField	84
RecipientSpecifier	84
RelatedIPMsField	85
RelatedIPMsSubfield	85
RelatedStoredFile	59,98
Relationship	59,98
Repertoire	17,86
RepliedToIPMField	14,85
ReplyRecipientsField	85
ReplyRecipientsSubfield	85
ReplyTimeField	85
ReturnedIPMField	26,88
RN	87
RNExtensionsField	88
SensitivityField	85
SubjectField	85
SubjectIPMField	24,88
SubscriptionError	94
SubscriptionProblem	94
SupplReceiptInfoField	88
TelephoneNumber	12,84
TeletexBodyPart	18,86
TeletexData	18,86
TeletexParameters	18,86
ThisIPMField	84
Time	83,92,102

	Page
VideotexBodyPart	86
VideotexData	86
VideotexParameters	86
VideotexSyntax	86
VoiceBodyPart	17
VoiceData	17,100
VoiceParameters	17,100
ASN.1 values	
acknowledgment-mode	79,108
authorizing-users	73,104
auto-submitted	56,74,95,105
auto-forward-comment	78,107
auto-forwarded	73,104
bilaterally-defined-body-part	57,96
bilaterally-defined-body-parts	75,106
blind-copy-recipients	74,104
body	74,105
conversion-eits	78,107
copy-recipients	73,104
discard-reason	78
encrypted-body-part	57,96
encrypted-body-parts	75,105
encrypted-data	77,106
encrypted-parameters	76,106
expiry-time	73,103
extended-body-part-types	77,107
file-transfer-body-part	59,98
g3-facsimile-body-part	57,96
g3-facsimile-body-parts	75,105
g3-facsimile-data	76,106
g3-facsimile-parameters	76,106
g4-class1-body-part	57,96
g4-class1-body-parts	75,105
general-text-body-part	58,97
heading	72,103
IA5	17
ia5-text-body-part	57,96
ia5-text-body-parts	75,105
ia5-text-data	76,106
ia5-text-parameters	76,106
id-bat	80
id-bat-bilaterally-defined-body-parts	81
id-bat-body	81
id-bat-encrypted-body-parts	81
id-bat-encrypted-data	81
id-bat-encrypted-parameters	81
id-bat-extended-body-part-types	81
id-bat-g3-facsimile-body-parts	81
id-bat-g3-facsimile-data	81
id-bat-g3-facsimile-parameters	81
id-bat-g4-class1-body-parts	81
id-bat-ia5-text-body-parts	81
id-bat-ia5-text-data	81
id-bat-ia5-text-parameters	81
id-bat-message-body-parts	81
id-bat-message-data	81
id-bat-message-parameters	81
id-bat-mixed-mode-body-parts	81
id-bat-nationally-defined-body-parts	81
id-bat-teletex-body-parts	81
id-bat-teletex-data	81
id-bat-teletex-parameters	81
id-bat-videotex-body-parts	81
id-bat-videotex-data	81
id-bat-videotex-parameters	81
id-cs-eit-authority	82
id-eit	80
id-eit-file-transfer	82
id-ep	80
id-ep-encrypted	82
id-ep-file-transfer	82
id-ep-g3-facsimile	82
id-ep-general-text	82
id-ep-ia5-text	82
id-ep-message	82
id-ep-teletex	82
id-ep-videotex	82
id-ep-vmg-body-part	82
id-et	79
id-et-bilaterally-defined	80
id-et-encrypted	80
id-et-file-transfer	80
id-et-g3-facsimile	80
id-et-g4-class	80
id-et-general-text	80
id-et-ia5-text	80
id-et-message	80
id-et-mixed-mode	80
id-et-nationally-defined	80
id-et-teletex	80
id-et-videotex	80
id-et-vmg-body-part	80
id-hat	80
id-hat-authorizing-users	81
id-hat-auto-forwarded	81
id-hat-auto-submitted	81
id-hat-blind-copy-recipients	81
id-hat-copy-recipients	81
id-hat-expiry-time	81
id-hat-heading	81
id-hat-importance	81
id-hat-incomplete-copy	81
id-hat-languages	81

	Page
id-hat-nrn-requestors	81
id-hat-obsoleted-IPMs	81
id-hat-originator	81
id-hat-primary-recipients	81
id-hat-related-IPMs	81
id-hat-replied-to-IPM	81
id-hat-reply-recipients	81
id-hat-reply-requestors	81
id-hat-reply-time	81
id-hat-rn-requestors	81
id-hat-sensitivity	81
id-hat-subject	81
id-hat-this-ipm	81
id-hex	80
id-hex-auto-submitted	80
id-hex-incomplete-copy	80
id-hex-languages	80
id-ipms	79
id-iso-cs	82
id-iso-mod	82
id-mct	80
id-mct-p2-1984	82
id-mct-p2-1988	82
id-mod	79
id-mod-abstract-service	80
id-mod-extended-body-part-types	80
id-mod-extended-body-part-types-2	82
id-mod-extended-voice-body-part-type	80
id-mod-file-transfer-body-part-type	80
id-mod-functional-objects	80
id-mod-heading-extensions	80
id-mod-information-objects	80
id-mod-message-store-attributes	80
id-mod-object-identifiers	80
id-mod-object-identifiers-2	82
id-mod-upper-bounds	80
id-nat	80
id-nat-acknowledgment-mode	82
id-nat-auto-forward-comment	82
id-nat-conversion-eits	81
id-nat-discard-reason	81
id-nat-ipm-preferred-recipient	81
id-nat-ipn-originator	81
id-nat-non-receipt-reason	81
id-nat-receipt-time	82
id-nat-returned-ipm	82
id-nat-subject-ipm	81
id-nat-suppl-receipt-info	82
id-ot	79
id-ot-ipme	80
id-ot-ipms	80
id-ot-ipms-ms	80
id-ot-ipms-ua	80
id-ot-ipms-user	80
id-ot-pdau	80
id-ot-tlma	80
id-ot-tlxau	80
id-pt	79
id-pt-management	80
id-pt-origination	80
id-pt-reception	80
id-ref	79
id-ref-primary	80
id-ref-secondary	80
id-sat	80
id-sat-ipm-entry-type	81
id-sat-ipm-synopsis	81
importance	73,104
incomplete-copy	56,74,94,105
IPM	70
ipm-entry-type	69,102
ipm-expired: Auto-discard	25
ipm-obsoleted: Auto-discard	25
ipm-preferred-recipient	78,107
ipm-synopsis	71,102
ipme	27,89
ipme-refinement	27
ipme-refinement	90
ipms	29,90
ipms-refinement	37
ipms-user	28
IPMS-eos-not-subscribed	36
ipms-ms	39,91
ipms-refinement	90
ipms-ua	39,90
ipms-user	90
ipn-originator	78,107
ITA2	17
languages	56,74,95,105
management	34,92
message-body-part	57,96
message-body-parts	75,105
message-data	77,107
message-parameters	76,106
mixed-mode-body-part	57,96
mixed-mode-body-parts	75,105
MTS-eos-not-subscribed	36
nationally-defined-body-part	57,96
nationally-defined-body-parts	75,106
non-receipt-reason	78,107
NRN	70
nrn-requestors	72,103
obsoleted-IPMs	74,104
origination	30,92

	Page		Page
originator	72,103	teletex-data	76,106
pdau	40,91	teletex-parameters	76,106
primary-recipients	73,104	this-ipm	72,103
receipt-time	79,108	tlma	39,91
reception	32,92	tlxau	40,91
related-IPMs	74,104	ub-auto-forward-comment	108
replied-to-IPM	73,103	ub-free-form-name	108
reply-recipients	74,104	ub-ipm-identifier-suffix	108
reply-requestors	72,103	ub-local-ipm-identifier	108
reply-time	73,104	ub-subject-field	108
returned-ipm	78,107	ub-telephone-number	108
RN	70	user-subscription-terminated	25
rn-requestors	72,103	videotex-body-part	57,96
sensitivity	73,104	videotex-body-parts	75,105
subject	73,103	videotex-data	76,106
subject-ipm	78,107	videotex-parameters	76,106
suppl-receipt-info	79,108	voice-body-part	101
teletex-body-part	57,96	voice-body-part	64
teletex-body-parts	75,105		

ANNEX Q

(to Recommendation X.420)

Alphabetical list of abbreviations used in this Recommendation

ADPCM	Adaptive differential pulse code modulation
ASN.1	Abstract syntax notation one
AU	Access unit
DL	Distribution list
EIT	Encoded information type
FTAM	File transfer access and management
IA5	International Alphabet No. 5
IPM	Interpersonal message
IPM EOS	Interpersonal message element of service
IPME	Interpersonal messaging environment
IPMS	Interpersonal messaging system

IPMS MS	Interpersonal messaging system message store
IPMS UA	Interpersonal messaging system user agent
IPMS user	Interpersonal messaging system user
IPN	Interpersonal notification
ITA5	International Telegraph Alphabet No. 5
MHS	Message handling system
MOTIS	Message-oriented text interchange system
MS	Message store
MTA	Message transfer agent
MTS	Message transfer system
NBP	Non-basic parameter
NRN	Non-receipt notification
O/R	Originator/recipient
ODA	Open document architecture
ODIF	Open document interchange format
ON	Other notification
OSI	Open System Interconnection
P2	Protocol 2
PDAU	Physical delivery access unit
PDS	Physical delivery system
RN	Receipt notification
TLMA	Telematic agent
TLXAU	Telex access unit
UA	User agent