INTERNATIONAL STANDARD

ISO/IEC 10021-7

Third edition
2003-12-15

Information technology — Message Handling Systems (MHS): Interpersonal messaging system

Technologies de l'information — Systèmes de messagerie (MHS): Système de messagerie de personne à personne

Reference number
ISO/IEC 10021-7:2003(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

CONTENTS

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 10021-7 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems* in collaboration with ITU-T. The identical text is published as ITU-T Rec. X.420.

This third edition cancels and replaces the second edition (ISO/IEC 10021-7:1997), which has been technically revised. It also incorporates Amendment 1:1998 and Corrigendum 1:1998.

ISO/IEC 10021 consists of the following parts, under the general title *Information technology — Message Handling Systems (MHS)*:

— *Part 1: System and service overview*

— *Part 2: Overall architecture*

— *Part 4: Message transfer system — Abstract service definition and procedures*

— *Part 5: Message store: Abstract service definition*

— *Part 6: Protocol specifications*

— *Part 7: Interpersonal messaging system*

— *Part 8: Electronic Data Interchange Messaging Service*

— *Part 9: Electronic Data Interchange Messaging System*

— *Part 10: MHS routing*

— *Part 11: MHS Routing — Guide for messaging systems managers* [Technical Report]

Introduction

This Specification is one of a set of Recommendations | International Standards 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 Specification defines the Message Handling application called *Interpersonal Messaging*, specifying in the process the message content type and associated procedures known as *P2*.

This Specification was developed jointly by ITU-T and ISO/IEC. It is published as common text as ITU-T Rec. X.420 | ISO/IEC 10021-7.

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

Information technology – Message Handling Systems (MHS) – Interpersonal Messaging System

SECTION 1 – INTRODUCTION

1 Scope

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

This Recommendation | International Standard is one of a series on Message Handling. ITU-T 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. ITU-T Rec. X.402 | ISO/IEC 10021-2 identifies those documents as well.

This Recommendation | International Standard is structured as follows. Section one is this introduction. Section two defines the kinds of information objects exchanged in Interpersonal Messaging. Section three defines the associated abstract service. Section four specifies how it is provided. Annexes provide important supplemental information.

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

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. 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 | International Standard 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 Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Open Systems Interconnection

This Specification cites the following OSI specifications:

- ITU-T Recommendation X.227 (1995) | ISO/IEC 8650-1:1995, *Information technology – Open Systems Interconnection – Connection-oriented protocol for the Association Control Service Element: Protocol specification*.
- ITU-T Recommendation X.680 (1997) | ISO/IEC 8824-1:1998, *Information technology – Abstract Syntax Notation One (ASN.1) – Specification of Basic Notation*.
- ITU-T Recommendation X.681 (1997) | ISO/IEC 8824-2:1998, *Information technology – Abstract Syntax Notation One (ASN.1) – Information Object Specification*.
- ITU-T Recommendation X.682 (1997) | ISO/IEC 8824-3:1998, *Information technology – Abstract Syntax Notation One (ASN.1) – Constraint Specification*.
- ITU-T Recommendation X.683 (1997) | ISO/IEC 8824-4:1998, *Information technology – Abstract Syntax Notation One (ASN.1) – Parameterization of ASN.1 Specifications*.
- ITU-T Recommendation X.690 (1997) | ISO/IEC 8825-1:1998, *Information technology – ASN.1 Encoding Rules – Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*.

– ITU-T Recommendation X.880 (1994) | ISO/IEC 13712-1:1995, *Information technology – Remote Operations – Concepts, Model and Notation*.

2.2 Message Handling Systems

This Specification cites the following Message Handling System specifications:

– ITU-T Recommendation F.400/X.400 (1999), *Message handling: System and service overview*.

ISO/IEC 10021-1:2003, *Information technology – Message Handling Systems (MHS) – Part 1: System and service overview*.

– ITU-T Recommendation X.402 (1999) | ISO/IEC 10021-2:2003, *Information technology – Message Handling Systems (MHS) – Overall architecture*.

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

– ITU-T Recommendation X.411 (1999) | ISO/IEC 10021-4:2003, *Information technology – Message Handling Systems (MHS) – Message transfer system — Abstract service definition and procedures*.

– ITU-T Recommendation X.413 (1999) | ISO/IEC 10021-5:1999, *Information technology – Message Handling Systems (MHS) – Message store: Abstract service definition*.

– ITU-T Recommendation X.419 (1999) | ISO/IEC 10021-6:2003, *Information technology – Message Handling Systems (MHS) – Protocol specifications*.

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

2.3 Directory Systems

This Specification cites the following Directory System specifications:

– ITU-T Recommendation X.501 (1997) | ISO/IEC 9594-2:1998, *Information technology – Open Systems Interconnection – The Directory – Models*.

– ITU-T Recommendation X.509 (1997) | ISO/IEC 9594-8:1998, *Information technology – Open Systems Interconnection – The Directory – Authentication Framework*.

– ITU-T Recommendation X.520 (1997) | ISO/IEC 9594-6:1998, *Information technology – Open Systems Interconnection – The Directory – Selected Attribute Types*.

2.4 Language Code

This Specification cites the following Language Code specification:

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

2.5 Character Sets

This Specification cites the following Character Set specifications:

– ISO/IEC 2022:1994, *Information technology – Character code structure and extension techniques*.

– 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*.

– ISO 10646-1:1993, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*.

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

2.6 Telematic Services

This Specification cites the following Telematic Service specifications:

– ITU-T Recommendation T.4 (1993), *Standardization of group 3 facsimile apparatus for document transmission*.

- ITU-T Recommendation T.30 (1993), *Procedures for document facsimile transmission in the general switched telephone network*.
- CCITT Recommendation T.100 (1988), *International information exchange for interactive videotex*.
- ITU-T Recommendation T.101 (1994), *International interworking for videotex services*.
- CCITT Recommendation T.330 (1988), *Telematic access to interpersonal messaging system*.

2.7 File Transfer

This Specification 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*.
- ISO 8571-2:1988, *Information processing systems – Open Systems Interconnection – File Transfer, Access and Management – Part 2: Virtual Filestore Definition*.
- ISO 8571-2:1988/Amd.1:1992, *Information processing systems – Open Systems Interconnection – File Transfer, Access and Management – Part 2: Virtual Filestore Definition – Amendment 1: Filestore Management*.
- ISO 8571-4:1988, *Information processing systems – Open Systems Interconnection – File Transfer, Access and Management – Part 4: File Protocol Specification*.
- ISO 8571-4:1988/Amd.1:1992, *Information processing systems – Open Systems Interconnection – File Transfer, Access and Management – Part 4: File Protocol Specification – Amendment 1: Filestore Management*.

2.8 Open Document Architecture

This Specification cites the following Open Document Architecture specifications:

- ITU-T Recommendation T.415 (1993), | ISO/IEC 8613-5:1994, *Information technology – Open document architecture (ODA) and Interchange Format: Open Document Interchange Format*.

2.9 Digital Encoding of Sound

This Specification cites the following specifications on the Digital Encoding of Sound:

- CCITT Recommendation G.711 (1988), *Pulse code modulation (PCM) of voice frequencies*.
- CCITT Recommendation G.726 (1990), *40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)*.
- CCITT Recommendation G.728 (1992), *Coding of Speech at 16 kbit/s Using Low-Delay Code Excited Linear Prediction*.
- IEC 908: 1987, *Compact Disc Digital Audio*.

2.10 Cryptography

This Specification cites the following Cryptographic specification:

- RSA Laboratories. PKCS#7: *Cryptographic Message Syntax Standard. Version 1.5, November 1993*.

3 Definitions

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

4 Abbreviations

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

5 Conventions

This Specification uses the descriptive conventions identified below.

5.1 ASN.1

This Specification 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. ASN.1 is defined in ITU-T Rec. X.680 | ISO/IEC 8824-1, ITU-T Rec. X.681 | ISO/IEC 8824-2, ITU-T Rec. X.682 | ISO/IEC 8824-3 and ITU-T Rec. X.683 | ISO/IEC 8824-4.

b) To define the functional objects of Interpersonal Messaging, the MHS-OBJECT information object class of ITU-T Rec. X.411 | ISO/IEC 10021-4.

c) To define the abstract service of Interpersonal Messaging, the PORT and ABSTRACT-OPERATION and ABSTRACT-ERROR information object class of ITU-T Rec. X.411 | ISO/IEC 10021-4 and the CONTRACT information object class of ITU-T Rec. X.880 | ISO/IEC 13712-1.

d) To define the *IPMS extensions*, the IPMS-EXTENSION information object class of 7.2.17.

e) To define *extended body part types*, the EXTENDED-BODY-PART-TYPE information object class of 7.3.1.

f) To define IPMS-MS attributes, the ATTRIBUTE information object class of ITU-T Rec. X.413 | ISO/IEC 10021-5.

The abstract-syntax defined in this Specification may be mapped to that used in previous editions as follows. All ASN.1 definitions of object sets and Enumerated types which contain the ASN.1 extensions marker ("...") are treated as if any extension additions following the marker are absent. For ASN.1 definitions where the extension marker is not used, the ASN.1 comment "-- *1994 extension* --" has a similar interpretation. See 5.7 of ITU-T Rec. X.413 | ISO/IEC 10021-5. The effect of this is that certain attribute-types, matching-rules, and auto-actions are not standardized for use in 1988 Application Contexts.

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

Table 1 – Uses of the ASN.1 Notation

Subject Matter	Exposition	Reference
Object Identifiers	-	annex C
Abstract information objects	section two	annex D
Extended body part types	clauses 7.3, 7.4	annex E
Functional objects	clauses 10, 11, 16	annex F
Abstract service	clauses 12-13	annex G
Message store attributes	clause 19	annex I
Message store auto-actions	clause 19	annex J
Heading extensions	annex A	annex H
Security extensions	annex B	annex K
Upper bounds	-	annex L

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

Except for Annex J, 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 information object classes, derived from the correspondingly named information object classes 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 information object classes and with minimal additional specification, actually could be invoked via ROS is immaterial in the present context.

5.2 Grade

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

5.3 Terms

Throughout this Specification, 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.

5.4 Conventions for attribute-types used in Table 5

This Specification uses the conventions listed below in its definition of the attribute-types for the IPMS-MS abstract-service.

For the column headed 'Single/Multi-valued' the following values can occur:

S	single-valued
M	multi-valued.

For the column headed 'Source' the following values can occur:

IPM	Originate-IPM, Receive-IPM abstract-operations;
Mod	Modify abstract-operation;
MS	IPMS Message Store;
NRN	Originate-NRN, Receive-NRN abstract-operations;
ON	Originate other-notifications, Receive other-notifications;
RN	Originate-RN, Receive-RN abstract-operations.

5.5 Interpretation of UTC Time values

Dates and times in the MHS protocols are represented using the ASN.1 *UTCTime* type which uses only two decimal digits to represent the year, leaving the century unspecified. Since MHS systems must deal with dates both in the past (e.g. submission times of old messages which may be held in local storage or forwarded) and in the future (expiry time, deferred delivery time), it is important to observe a standard convention to avoid inaccurate display or malfunction of the MHS when dates from different centuries are compared.

The two decimal digits give 100 different years that can be expressed; an implementation has to associate each of these values with a particular century. The chosen convention is that dates up to ten years prior to the current time and up to forty years ahead of the current time should be associated with the corresponding century, with the interpretation of the remaining 49 values being implementation dependent. For example, for a system operating in 1996 the values "86" to "99" are interpreted as 1986 to 1999 and the values "00" to "36" are interpreted as 2000 to 2036, and the values "37" to "85" are implementation dependent.

NOTE – This convention permits two possible implementation strategies. An implementation can choose a fixed interpretation of all the year values, such that the convention is satisfied throughout the expected life of the product, or it can interpret the dates dynamically, based on the current date, such that the implementation remains valid indefinitely. For example, an implementation could choose the fixed range 1970 to 2069 for the available values, meaning that the implementation would require revision if it is still in use by the year 2029.

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 7.4.7) 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).

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.

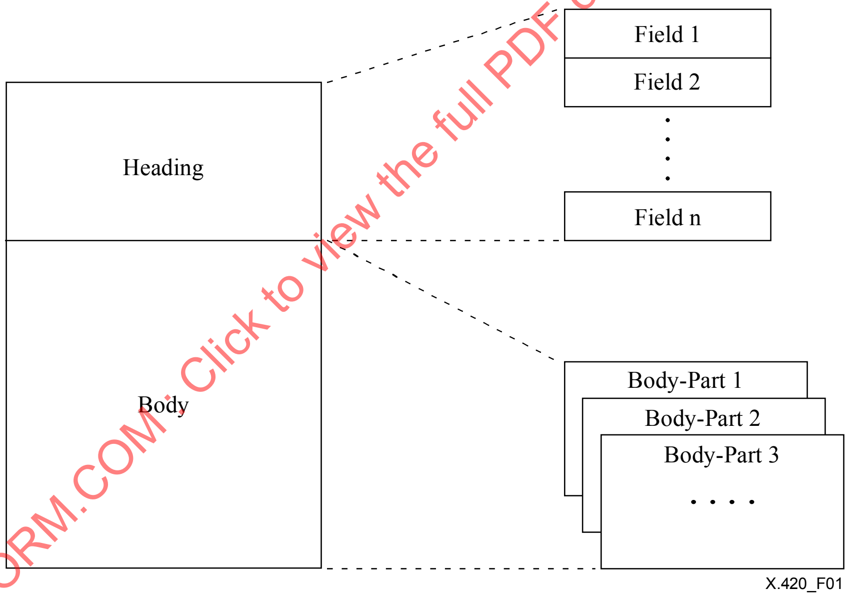

Figure 1 – An Interpersonal Message

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.

7.1 Heading Field Component Types

Information items of several kinds appear throughout the Heading. These Heading field component types – *IPM identifier*, *recipient specifier*, *OR-descriptor* and *IPMS extension* – 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 OR-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 L)|]. 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 Specification assigns.

7.1.2 Recipient Specifier

A **recipient specifier** is an information item that identifies an (intended) 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 recipient in question. An *OR-descriptor*.

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

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

```
NotificationRequests ::= BIT STRING {
       rn             (0),
       nrn            (1),
       ipm-return     (2),
       an-supported   (3),
       suppress-an    (4)}
```

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*.

iv) *an-supported*: It is requested that *advice notifications* (conveying *absence advice* or *change of address advice*) are conveyed within ONs as prescribed in clause 8.

v) *suppress-an*: It is requested that no *advice notifications* (conveying *absence advice* or *change of address advice*) are sent in response to this IPM.

c) **Reply-requested** (D *false*): Indicates whether a reply is requested of the 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 intended 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 {{ RecipientExtensions }}
```

General recipient extensions are defined in Annex A. Recipient extensions for IPMS Security are defined in Annex B. There are no other recipient extensions defined in this Specification.

7.1.3 OR-Descriptor

An **OR-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 OR-descriptor has the following components:

a) **Formal-name** (C): Identifies the user or DL in question. One of its OR-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 OR-descriptor appears in the *Reply Recipients* heading field.

iii) The OR-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 L)|], 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 L)|], 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 1 – One or more OR-descriptors may appear in each of the following heading fields: Originator, Authorizing Users, Primary Recipients, Copy Recipients, Blind Copy Recipients, Circulation List Recipients, Reply Recipients and Distribution Codes. In addition, an OR-descriptor may appear in the following notification fields (see clause 8): IPN Originator and IPM Intended Recipient.

NOTE 2 – The omission of the formal name can be used to indicate names of recipients and other users who do not receive the message through MHS, and who do not have MHS OR-names.

7.1.4 IPMS Extension

An **IPMS extension** (or **extension**) is an information item which enables extensions to be defined.

```
IPMSExtension {IPMS-EXTENSION:ChosenFrom} ::= SEQUENCE {
       type  IPMS-EXTENSION.&id({ChosenFrom}),
       value IPMS-EXTENSION.&Type({ChosenFrom} {@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. Each extension type shall occur at most once in a set of ExtensionsField, unless multiple occurrences are explicitly permitted in the definition of the extension type. The same extension type may occur in different places in the protocol. This applies to both standardised extensions and private extensions.

Every extension is defined by means of the following information object class.

```
IPMS-EXTENSION ::= CLASS {
       &id OBJECT IDENTIFIER UNIQUE,
       &Type DEFAULT NULL }
       WITH SYNTAX { [VALUE &Type , ] IDENTIFIED BY &id }
```

The &id value field is the identifier field for the class. It distinguishes an IPMS Extension from all other instances of the class.

The &Type field defines the data type to which every value of this instance of IPMS Extension shall conform.

```
PrivateIPMSExtensions_IPMS-EXTENSION ::= { ... }
```

NOTE – Whenever an IPMS extension is defined, it should also be considered whether new MS attributes should also be defined (see 19.6), and whether the UA operation needs to be modified (see clause 18). As any extension may be ignored, privately-defined extensions are not able to mandate support of the new MS attributes or UA operation.

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 OR-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 OR-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 Specification; it is given meaning by users.

This conditional field shall be present 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 Specification; 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 Specification; 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** 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. It may be present as a Heading field, or alternatively as an equivalent MTS extension that may be present in the per-recipient-message-submission-extensions field of a message-submission envelope and in the message-delivery-extensions field of a message-delivery envelope.

```
BlindCopyRecipientsField    ::= SEQUENCE OF BlindCopyRecipientsSubfield

BlindCopyRecipientsSubfield ::= RecipientSpecifier

blind-copy-recipients EXTENSION  ::= {
       BlindCopyRecipientsField,
       IDENTIFIED BY standard-extension:41 }
```

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.

NOTE – When submitting through an MS which provides Storage on Submission, the use of the alternative envelope encoding will result in a single submitted-message entry instead of an additional submitted-message entry for each blind copy recipient, which gives greater efficiency of submission, better correspondence between the user's perception of submitting one IPM and the resultant stored entry, and improved correlation of reports and notifications for blind copy recipients with the submitted-message entry. However, if the blind copy recipient's MS or UA conforms to an earlier version of this specification, then use of the alternative envelope encoding will result in the absence of requested notifications and the recipient being aware only implicitly rather than explicitly that he was a blind copy recipient.

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 Specification; 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 L)|], chosen from the graphic subset of the Teletex String character set. A length of zero is discouraged.

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

The *Extended Subject* (see A.1.9), which has an unrestricted character repertoire, takes precedence over this heading field if both are present on reception.

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 recipients of any replies to the present IPM. It comprises a Sequence of sub-fields, each an OR-descriptor, one for each user or DL.

```
ReplyRecipientsField    ::= SEQUENCE OF ReplyRecipientsSubfield

ReplyRecipientsSubfield ::= ORDescriptor (WITH COMPONENTS{..., formal-name PRESENT})
```

This conditional field shall be present 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 Specification; 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 intended recipients as individuals, rather than in their professional capacities.

b) *private*: The IPM should be conveyed to no one other than its intended 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 each conveying one such information item.

```
ExtensionsField ::= SET OF IPMSExtension {{ HeadingExtensions }}

HeadingExtensions IPMS-EXTENSION ::= {
        authorization-time |
        auto-submitted |
        body-part-signatures |
        circulation-list-recipients |
        distribution-codes |
        extended-subject |
        incomplete-copy |
        information-category |
        ipm-security-label |
        languages |
        manual-handling-instructions |
        originators-reference |
        precedence-policy-identifier |
        PrivateIPMSExtensions, ... }
```

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

NOTE – Future addenda or versions of this Specification may define additional extensions. Furthermore, future addenda and versions are likely to add information to the Heading only by means of this field.

7.3 Body Parts

The **IPM Body Part Table** information object set has as its members the types of body part that may appear in the Body of an IPM:

```
IPMBodyPartTable EXTENDED-BODY-PART-TYPE ::= {
        StandardBodyParts |
        ApplicationSpecificBodyParts}
```

The **Standard Body Parts** information object set has as its members the types of body part defined in this Specification:

```
StandardBodyParts EXTENDED-BODY-PART-TYPE ::= {
        ia5-text-body-part |
        g3-facsimile-body-part |
        g4-class1-body-part |
        teletex-body-part |
        videotex-body-part |
        encrypted-body-part |
        message-body-part |
        mixed-mode-body-part |
        bilaterally-defined-body-part |
        nationally-defined-body-part |
        general-text-body-part |
        file-transfer-body-part |
        voice-body-part |
        report-body-part |
        notification-body-part |
        content-body-part |{{1 2 3 -- RELATIVE-OID to be provided --}}
        pkcs7-body-part,
        ... }
```

The **Application Specific Body Parts** information object set has as its members the types of body part defined in other Specifications, and defined for proprietary or private use:

```
ApplicationSpecificBodyParts EXTENDED-BODY-PART-TYPE ::= {
        --any body part defined in other Specifications, or for proprietary or private use --
        ... }
```

Some types of body part contain a single component, the *Data* component. Other types of body part contain two components, *Parameters* and *Data*. The **Parameters** component, if defined, comprises information items that describe the information object the body part represents, and typically contains format and control parameters. The **Data** component is the information object itself.

7.3.1 Extended Body Part

Every body part type defined in this Specification is defined as an instance of the **Extended Body Part Type** information object class. Every body part type defined elsewhere shall be defined in the same way.

```
EXTENDED-BODY-PART-TYPE ::= CLASS {
       &parameters   TYPE-IDENTIFIER OPTIONAL,
       &data         TYPE-IDENTIFIER }
       WITH SYNTAX { [PARAMETERS  &parameters,] DATA  &data }
```

An instance of the Extended Body Part Type information object class defines, by means of its &parameters field, the type of the data value that is represented by the Parameters component of such a body part, and the Object Identifier that identifies this Parameters component. The presence of the &parameters field implies the presence of the Parameters component in every instance of that specific extended body part type; its omission implies the absence of the Parameters component in every instance.

An instance of the Extended Body Part Type information object class also defines, by means of its &data field, the type of the data value that is represented by the Data component of such a body part, and the Object Identifier that identifies this Data component. The Object Identifier identifies the encoding rules for the body part. Those body parts whose types are defined in this Specification shall be encoded using ASN.1's basic encoding rules.

Specific Extended body part types may be defined by users to convey any type of information object. The definitions in 7.4.11, 7.4.12, 7.4.14 and 7.4.15 give examples where the information object is an ASN.1 Type, while the examples below show definitions for information objects not defined by ASN.1. User-defined Extended body part types are appropriate where the recipient's UA will be configured to invoke a process to render that body part, but where the requirement is to move information into the recipient's filestore the File Transfer body part (see 7.4.12) may be more appropriate.

Examples

An organisation has two products for which it wishes to define Extended body parts. Its 'Simple Spreadsheet' uses a single file of octets for its data, while its 'Whizzbang Wordprocessor' always uses two related data files of octets (e.g., one for format proforma, and one for the text using these proforma). The organisation has obtained a branch of the Object Identifier tree from its national registration authority, called here *local-object-identifier*.

The following definition is an Extended body part for the 'Simple Spreadsheet' data file, and the same object identifier ({ local-object-identifier 1 }) can be used for its Encoded Information Type:

```
simple-spreadsheet-body-part EXTENDED-BODY-PART-TYPE ::= {
       DATA       { OCTET STRING IDENTIFIED BY { local-object-identifier 1 } } }
```

The following definition is an Extended body part for the 'Whizzbang Wordprocessor' data files, with the format proforma file being placed in the Parameters component and the text file in the Data component. The same object identifier as used for the Data component ({ local-object-identifier 3 }) can be used for its Encoded Information Type:

```
whizzbang-wordprocessor-body-part EXTENDED-BODY-PART-TYPE ::= {
       PARAMETERS { OCTET STRING IDENTIFIED BY { local-object-identifier 2 } },
       DATA       { OCTET STRING IDENTIFIED BY { local-object-identifier 3 } } }
```

NOTE 1 – This body part type enables the exchange of information objects of all kinds, each kind unambiguously identified. This unambiguous identification relies upon the use of Object Identifiers. Object Identifiers are easily obtained, e.g., by national bodies and private organizations. It is desirable that the identification is also unique, and anyone defining a data format is encouraged to define and publicise an Extended body part for that format in order to remove the likely non-uniqueness if such definition is left to the users of that data format.

NOTE 2 – If an Extended body part has a Parameters component, its Object Identifier is allocated at the same time and by the same naming authority as that for the Data component, and must be different to it.

NOTE 3 – When a new Extended body part type and a new Encoded Information Type are defined and have a one to one relationship, then the same Object Identifier may be used for the data component and the Encoded Information Type.

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

7.3.2 Body Part Encoding

Each body part present in the Body of an IPM is represented as follows:

```
BodyPart ::= CHOICE {
       basic  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 },
       extended                   [15] ExtendedBodyPart{{IPMBodyPartTable}}
```

All body part types are divided into two classes as follows:

a) basic: Denoted by an integer (an ASN.1 context-specific tag).

All basic body part types are defined in this Specification. Each basic body part type also has an equivalent extended body part definition.

b) **extended**: Denoted by an Object Identifier.

Some extended body part types are defined in this Specification. Some are defined in other standards (e.g., the ODA body part defined in ITU-T Rec. T.411 | ISO/IEC 8613-1). Others may be defined by users. The extended body part type enables the exchange of information objects of any kind, each kind unambiguously and uniquely identified.

An instance of the extended body part contains an information object whose semantics and abstract syntax are denoted by the Object Identifier which the body part carries. It has Parameters and Data components.

```
ExtendedBodyPart{EXTENDED-BODY-PART-TYPE:IPMBodyPartTable} ::= SEQUENCE {
       parameters [0] INSTANCE OF TYPE-IDENTIFIER OPTIONAL,
       data           INSTANCE OF TYPE-IDENTIFIER }
             (CONSTRAINED BY {-- must correspond to the &parameters field and &data field
             -- of a member of -- IPMBodyPartTable})
```

The Parameters and Data components correspond to fields of the Extended Body Part Type information object class (see 7.3.1). Hence each component is defined as an instance of the Type-Identifier information object class (see Annex A of ITU-T Rec. X.681 | ISO/IEC 8824-2).

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 ISO6937Text body parts, respectively, which are no longer defined. In CCITT Recommendation X.420 (1984), CCITT Recommendation 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.

NOTE 3 – The basic body part types exist for purely historical reasons, predating the extended body part type.

NOTE 4 – In editions of this Specification published before 1994, the Parameters and Data components of extended body parts were defined as Externals. When the single-ASN1-type encoding alternative of the External is used, the value of the encoding of an External is identical with that of Type-Identifier. However, to accommodate the case where the octet-aligned encoding alternative is used, an associated type for the representation of the extended body part may be used on origination, and should be supported on reception. This associated type is assumed to be defined in an environment of EXPLICIT TAGS:

```
SEQUENCE {
      parameters [0] IMPLICIT SEQUENCE {
            direct-reference EXTENDED-BODY-PART-TYPE.&parameters.&id,
            encoding CHOICE {
                  single-ASN1-type [0] EXTENDED-BODY-PART-TYPE.&parameters.&Type,
                  octet-aligned    [1] IMPLICIT OCTET STRING } } OPTIONAL,
      data [UNIVERSAL 8] IMPLICIT SEQUENCE {
            direct-reference EXTENDED-BODY-PART-TYPE.&data.&id,
            encoding CHOICE {
                  single-ASN1-type [0] EXTENDED-BODY-PART-TYPE.&data.&Type,
                  octet-aligned    [1] IMPLICIT OCTET STRING } } }
```

If the Parameters or Data component is defined as an Octet String, or comprises octet-aligned data not defined using ASN.1, its encoding may occupy either the single-ASN1-type alternative (as an explicitly tagged Octet String) or the octet-aligned alternative (as an implicitly tagged Octet String). Otherwise, the encoding should occupy the single-ASN1-type alternative.

7.4 Standard Body Part Types

The standard body part types defined in this Specification are enumerated below.

7.4.1 IA5 Text

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

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

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

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

IA5TextData ::= IA5String
```

The Parameters component comprises the following parameters:

a) **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).

7.4.2 G3 Facsimile

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

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

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

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 CCITT 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 ITU-T 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 ITU-T 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.

NOTE 4 – All of the pages in any one G3 Facsimile body part are required to have the same values for the non-basic parameters. When a message is created containing facsimile data where pages have different values for these parameters (e.g., different paper size) it will be necessary to use a separate body part for each group of adjacent pages that have the same parameter values. Except where the originator of the message deliberately requests separate body parts, all consecutive pages that have the same parameter values should be placed in a single body part. In particular, when receiving pages from a facsimile terminal, if a new set of parameters are signalled, they should be compared with the parameters of the previous page; a new body part should be created only if the parameters have changed.

NOTE 5 – The G3 Facsimile body part is primarily intended for the conveyance of bitmap graphical data. ITU-T Recs. T.4 and T.30 provide encodings which could, in principle, allow this body part to be used for a variety of other applications (e.g., file-transfer). However, more effective interworking will be achieved if these data types are encoded using more specific body parts (such as the file-transfer body part, see 7.4.12), or content-types (such as EDI-messaging).

7.4.3 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 ITU-T Rec. T.415 | ISO/IEC 8613-5, which describe the document's layout structure.

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

g4-class1-body-part EXTENDED-BODY-PART-TYPE ::= {
       DATA {G4Class1BodyPart IDENTIFIED BY id-et-g4-class1} }
```

7.4.4 Teletex

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

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

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

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 CCITT 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 CCITT 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 CCITT Recommendation T.61.

7.4.5 Videotex

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

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

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

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

VideotexData ::= VideotexString
```

The Parameters component comprises the following parameters:

a) **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 CCITT Recommendations T.100 and T.101:

i) *ids*: The IDS syntax.

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

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

iv) *data-syntax*3: 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.

7.4.6 Encrypted

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

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

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

EncryptedParameters ::= SET {
       algorithm-identifier      AlgorithmIdentifier,
       originator-certificates   ExtendedCertificates OPTIONAL,
       ... }

EncryptedData ::= ENCRYPTED { BodyPart }
```

The Parameters component comprises the following parameters:

a) **Algorithm-identifier** (M): Identifies the encryption algorithm used; see 8.5.10 of ITU-T Rec. X.411 | ISO/IEC 10021-4 and clause 8 of ITU-T Rec. X.509 | ISO/IEC 9594-8. Where this identifies a symmetric algorithm, a session key may be conveyed in the *Body Part Encryption Token* in recipient-extensions, see B.6.1.

NOTE – A symmetric algorithm may also be used where the message has a single recipient, or where the key is shared between the originator and all recipients (e.g. a closed user group), and key management (distribution) is performed outside the IPM. An asymmetric algorithm may be used directly for encryption of a body part (rather than for encryption of a session key) if the message has a single recipient, or where a private key is shared between all recipients.

b) **Originator-certificates** (C): This may be used to convey a verified copy of the public-asymmetric-encryption-key of the originator who encrypted the body part (i.e. a Certificate), or the name of a Directory entry containing the originator's certificate, or multiple certificates (or Directory names) where the public-asymmetric-encryption-key is verified with different certification paths or by different certification authorities. This component shall be present only if the asymmetric encryption algorithm requires a key of both originator and recipient (for example, the Diffie-Hellman algorithm). Where the originator certificate is required but is not included within these parameters, then the originator certificate which may be present on the envelope shall be used. If no certificates are present, it is assumed that another method (e.g. use of the Directory) is to be used to obtain a verified copy of the originator's public key.

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 ITU-T Rec. X.690 | ISO/IEC 8825-1.

7.4.7 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}

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

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. For a delivered message, 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**.

If the forwarded IPM represents a previously submitted IPM (rather than a delivered IPM) then a simulated delivery-envelope may be constructed to contain message-submission-time; the originator-name and this-recipient-name components of this envelope each contain the OR-address of the IPM's originator.

NOTE 1 – The possible future inclusion of a message identifier in the Parameters component may be the subject of future standardization. Its present omission provides compatibility with CCITT 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.

7.4.8 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 ITU-T Rec. T.415 | ISO/IEC 8613-5, which describe the document's layout structure.

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

mixed-mode-body-part EXTENDED-BODY-PART-TYPE ::= {
       DATA {MixedModeBodyPart IDENTIFIED BY id-et-mixed-mode} }
```

7.4.9 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

bilaterally-defined-body-part EXTENDED-BODY-PART-TYPE ::= {
        DATA {BilaterallyDefinedBodyPart IDENTIFIED BY id-et-bilaterally-defined} }
```

NOTE – The use of this body part type is discouraged. It predates the Extended body part type and is retained for backward compatibility with CCITT Recommendation X.420 (1984). The Extended 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.

7.4.10 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.

```
NATIONAL-BODY-PARTS ::= CLASS {&Type}

NationallyDefinedBodyPart ::= NATIONAL-BODY-PARTS.&Type
        -- Provided for Historic reasons. Use is strongly deprecated.

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

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 Extended body part type and is retained for backward compatibility with CCITT Recommendation X.420 (1984). The Extended 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.

7.4.11 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 IDENTIFIED BY 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, extended EITs are defined (for the purposes of 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 specification 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 1 – The extended EITs for Latin Alphabet number 1 (ISO 8859-1) are {id-cs-eit-authority 1} for the C0 set, {id-cs-eit-authority 6} for the G0 set and {id-cs-eit-authority 100} for the G1 set.

EXAMPLE 2 – The extended EITs for the Basic Multilingual Plane of ISO/IEC 10646-1 (16-bit encoding without restrictions on combining characters) are {id-cs-eit-authority 176} for the G0 set, {id-cs-eit-authority 1} for the basic C0 set, and (if required) {id-cs-eit-authority 77} for the C1 set of ISO 6429.

NOTE 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 (ITU-T Recs. of the X.690-series | ISO/IEC 8825) provide default character sets designated and invoked for G0 and C0. These rules require that any other character set designators are inserted within the encoding of the General String. However, since different editions of these rules specify different defaults, it is recommended that all required G sets (even sets 2 or 6 for IA5) are explicitly designated. 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.

7.4.12 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. It has Parameters and Data components.

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

FileTransferParameters ::= SEQUENCE {
       related-stored-file  [0] RelatedStoredFile OPTIONAL,
       contents-type        [1] ContentsTypeParameter DEFAULT document-type:
                                { document-type-name {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 OPTIONAL }

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.
```

NOTE – A number of elements of the FileTransferParameters are of type GraphicString. The ASN.1 Basic Encoding Rules specify that the encoding for these strings may include ISO/IEC 2022 escape sequences to designate and invoke graphic character sets. A default G0 set is provided, but any other G sets that are required must be explicitly designated and invoked. This requires that implementations accept on reception various combinations of escape sequences and/or shift functions even if only IA5 characters are supported. On origination, it is recommended that the default is ignored and that all required sets are explicitly designated and invoked using the appropriate escape sequences and locking shift functions. For implementations wishing to originate IA5 characters, it is recommended that character set 6 be invoked and designated (the octets required to achieve this are ESC, 2/8, 4/2, LS0).

7.4.12.1 Related Stored File Parameter

The related stored file 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 explicit-relationship: unspecified }

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

PathnameandVersion ::= SEQUENCE {
       pathname      [0] Pathname-Attribute,
       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 OPTIONAL,
        -- Defined in 8.5.5 of ITU-T Rec. X.411 | ISO/IEC 10021-4
        user-relative-identifier    [1] PrintableString }

Relationship ::= CHOICE {
        explicit-relationship       [0] ExplicitRelationship,
        descriptive-relationship    [1] GraphicString }

ExplicitRelationship ::= INTEGER {
        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 1 – 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** (C): Identifies the user who originated the referenced message. One of the user's OR-names. This conditional component shall be present unless the reference is an IPM identifier which does not contain a User component.

b) **User-relative-identifier** (M): 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.

NOTE 2 – The MessageReference shares the same value set with the IPMIdentifier, EDIMIdentifier and VMIdentifier. Hence a file transfer body part is capable of referencing IPM, EDIM or VM messages.

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.

7.4.12.2 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] DOCUMENT-PARAMETER.&Type 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 } }

Document-Type-Name ::= OBJECT IDENTIFIER

DOCUMENT-PARAMETER ::= CLASS {&Type}

Constraint-Set-Name ::= OBJECT IDENTIFIER

Abstract-Syntax-Name ::= OBJECT IDENTIFIER
```

The value is either a document-type name (optionally with parameters of the open type) or a pair of abstract syntax name and constraint set name. Each of these names is an 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)

7.4.12.3 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.

7.4.12.4 Compression Parameter

The compression parameter describes the compression type if the file is transferred in a compressed mode.

```
CompressionParameter ::= SEQUENCE {
        compression-algorithm-id     [0]
                         COMPRESSION-ALGORITHM.&id ({CompressionAlgorithmTable}),
        compression-algorithm-param  [1] COMPRESSION-ALGORITHM.&Type
                         ({CompressionAlgorithmTable} {@compression-algorithm-id})}

COMPRESSION-ALGORITHM ::= TYPE-IDENTIFIER

CompressionAlgorithmTable COMPRESSION-ALGORITHM ::= { ... }
```

7.4.12.5 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. 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
date and time of last attribute modification
identity of creator
identity of last modifier
identity of last reader
identity of last attribute modifier
availability
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,
       date-and-time-of-last-attribute-modification
                                              [7] 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,
       identity-of-last-attribute-modifier   [11] User-Identity-Attribute OPTIONAL,
       object-availability                   [12] Object-Availability-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 defined below or imported from ISO 8571-4.

7.4.12.5.1 Pathname Attribute

The pathname attribute provides a file name.

```
Pathname-Attribute ::=  CHOICE {
       incomplete-pathname  [0]  Pathname,
       complete-pathname    [23] Pathname }
```

7.4.12.5.2 Permitted Actions Attribute

The permitted actions attribute indicates the set of actions that can be performed on the file.

7.4.12.5.3 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 ::= GraphicString
```

7.4.12.5.4 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.

The date and time of last attribute modification attribute indicates when the attributes of the file were last modified.

7.4.12.5.5 Identity Attributes

The identity of creator, identity of last modifier, identity of last reader, and identity of last attribute modifier attributes identify the user(s) who created, last modified, and last read the file.

```
User-Identity-Attribute ::= CHOICE {
       no-value-available  [0] NULL,
       -- Indicates partial support of this attribute.
       actual-values           User-Identity }

User-Identity ::= GraphicString
```

7.4.12.5.6 Availability Attribute

The availability attribute indicates whether the file had (or should have) immediate or deferred availability (e.g., whether it was stored on permanently mounted or demountable storage media).

7.4.12.5.7 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.

7.4.12.5.8 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 }

Password ::= CHOICE { graphic-string GraphicString, octet-string OCTET STRING }

Pass-Passwords ::= SEQUENCE OF Password

Application-Entity-Title ::= SEQUENCE {
        ap-title     AP-title,
        ae-qualifier AE-qualifier }
-- AP-title and AE-qualifier are defined in ITU-T Rec. X.227 | ISO/IEC 8650-1
```

7.4.12.5.9 Legal Qualifications Attribute

The legal qualifications attribute conveys information about the legal status of the file and its use.

7.4.12.5.10 Private Use Attribute

The meaning of the private use attribute is not defined.

7.4.12.5.11 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.

7.4.12.6 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 are the same as those of the extensions heading field specified in 7.2.17.

7.4.12.7 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 the data values representing the file contents, and identifies the necessary abstract syntax(es). Where the file content comprises more than one data value, each value is placed in a separate External in the FileTransferData. 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.

For the purposes of FileTransferData, this Specification places additional restrictions on the encoding of the External ASN.1 type, excluding some of the implementation options permitted by the ASN.1 Basic Encoding Rules in 8.18 of ITU-T Rec. X.690 | ISO/IEC 8825-1:

a) If the data value is a single ASN.1 type, the single-ASN1-type choice shall be used; the options to place a BER-encoding of the data value in the octet-aligned or arbitrary choices are excluded.

b) If the data value comprises an integral number of octets, but is not a single ASN.1 type, the octet-aligned choice shall be used; the option to place octet-aligned data in the arbitrary choice is excluded.

A data value comprising a single ASN.1 Octet String and a data value comprising octets which are not specified as any ASN.1 type are considered equivalent, and either of the applicable encodings may be used (i.e. the single-ASN1-type choice containing an explicitly tagged Octet String, or the octet-aligned choice containing just the data octets without additional Octet String encoding).

7.4.12.8 Encoded Information Type

For this extended body part type, an extended EIT is defined (for the purposes of item c of 20.4) by the object identifier id-eit-file-transfer. This value shall be used in all instances of a message containing the File Transfer body part.

Additional EITs may optionally be derived from each of the parameter components Contents Type Parameter, Environment Parameter and Compression Parameter. If used, these shall be derived as follows:

a) If the Contents Type Parameter is encoded as document-type or specified by default, the additional EIT is the Object Identifier assigned to that document type. If the Contents Type Parameter is encoded as constraint-set-and-abstract-syntax, two additional EITs are defined, having the values of the Object Identifiers assigned to the constraint-set-name and abstract-syntax-name.

b) If the Environment Parameter contains an application-reference which is encoded as an Object Identifier, the additional EIT is the Object Identifier assigned to that application.

c) If the Compression Parameter is present, the additional EIT is the Object Identifier assigned to the compression-algorithm-id.

NOTE 1 – When a file-transfer body part is being submitted by a UA which did not originally encode it (e.g., when forwarding a message), the information necessary to encode the additional EITs may not be available. In this case, only the primary EIT id-eit-file-transfer needs to be used.

NOTE 2 – The use of additional EITs where the recipient has not registered to allow reception of those EITs will cause non-delivery. Practical use of additional EITs will therefore be subject to bilateral agreement between originator and recipient.

7.4.13 Voice

A **Voice** body part represents speech, or other types of audio data such as music. It has Parameters and Data components.

NOTE – The voice body part defined here replaces the basic voice body part specified in CCITT Rec. X.420 (1988) | ISO/IEC 10021-7: 1990.

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-voice},
        DATA       {VoiceData IDENTIFIED BY id-et-voice} }

VoiceParameters ::= SEQUENCE {
        voice-message-duration [0] INTEGER OPTIONAL,   -- In seconds
        voice-encoding-type [1] OBJECT IDENTIFIER,
        supplementary-information [2] IA5String OPTIONAL }

VoiceData ::= OCTET STRING
```

The Parameters component contains the following components:

a) **Voice-message-duration** (O): the duration of the voice encoded data component in seconds.

b) **Voice-encoding-type** (M): an object identifier which identifies the voice encoding applied to the data portion of this body part.

c) **Supplementary-information** (O): conveys additional information to advise the recipient of the content of the voice encoded data component.

Four values for voice-encoding-type are defined by this Specification to identify voice encodings commonly used with personal computers:

a) *id-voice-11khz-sample*: Identifies encoding as 8-bit linear-law samples at a nominal 11 Khz (11025 Hz) sampling frequency. Each sample comprises a signed integer value in the range -128 to +127, bearing a linear relationship to the amplitude of the analogue wave-form. Each sample is encoded as a single octet, containing a 2's complement representation of the integer value, with the least significant bit of the value in the least significant bit of the octet.

b) *id-voice-22khz-sample*: Identifies encoding as 8-bit linear-law samples at a nominal 22 Khz (22050 Hz) sampling frequency. Each sample comprises a signed integer value in the range -128 to +127, bearing a linear relationship to the amplitude of the analogue wave-form. Each sample is encoded as a single octet, containing a 2's complement representation of the integer value, with the least significant bit of the value in the least significant bit of the octet.

c) *id-voice-cd-quality*: Identifies encoding of 'Compact Disc' quality, as specified in IEC 908. Sampling occurs at a rate of 44100 Hz. Each sample comprises two 16-bit values, for left and right channels of a stereo pair, and is encoded as four octets in the following order: 8 least significant bits of 'left' channel; 8 most significant bits of 'left' channel; 8 least significant bits of 'right' channel; 8 most significant bits of 'right' channel. Within each octet, the most significant bit of the sample is placed in the most significant bit of the octet, and the least significant bit of the sample is placed in the least significant bit of the octet. The length of the data shall be a multiple of 4 octets.

d) *id-voice-g711-mu-law*: Identifies pulse code modulation (PCM) encoding at 8000 samples/sec as specified in CCITT Recommendation G.711, using µ-law. Each value specified by G.711 comprises 8 bits; each value is encoded as a single octet, with bit No. 1 (polarity bit) of the sample in the most significant bit of the octet, and bit No. 8 (the least significant bit) encoded in the least significant bit of the octet.

NOTE – The ability of individual implementations to record or replay voice body parts will be limited by hardware capabilities. In some cases the sampling rate will be slightly higher or lower than the nominal rate specified, or interpolation will be required to match 11 Khz samples with hardware capable of only 22 Khz.

Two further values for voice-encoding-type are defined to identify voice encodings commonly used in digital telephony:

a) *id-voice-g726-32k-adpcm*: Identifies 32 kbit/s adaptive differential pulse code modulation (ADPCM) encoding as specified in CCITT Recommendation G.726. The 4-bit code words of the G.726 encoding shall be packed into the octets of the OCTET STRING as follows: the first code word is placed in the four least significant bits of the first octet, with the least significant bit of the code word in the least significant bit of the octet; the second code word is placed in the four most significant bits of the first octet, with the most significant bit of the code word in the most significant bit of the octet. Subsequent pairs of code words shall be packed in the same way into successive octets, with the first code word of each pair placed in the least significant four bits of the octet. It is preferred that the voice sample be extended with silence such that the encoded value comprises an even number of code words. However, if the voice sample comprises an odd number of code words, then the last code word shall be discarded.

b) *id-voice-g728-16k-ld-celp*: Identifies 16 kbit/s low-delay code excited linear prediction (LD-CELP) encoding as specified in CCITT Recommendation G.728. The 10-bit code words of the G.728 encoding shall be packed into the octets of the OCTET STRING by considering groups of four code words as follows.

The eight least significant bits of the first code word are placed in the first octet, with the least significant bit of the code word as the least significant bit of the octet. The two most significant bits of the first code word are placed in the two least significant bits of the second octet, with the less significant of the bits from the code word as the least significant bit of the octet. The six least significant bits of the second code word are placed in the six most significant bits of the second octet, with the most significant bit of those taken from the code word as the most significant bit of the octet. The four most significant bits of the second code word are placed in the four least significant bits of the third octet, with the least significant of the bits from the code word as the least significant bit of the octet. The four least significant bits of the third code word are placed in the four most significant bits of the third octet, with the most significant bit of those taken from the code word as the most significant bit of the octet. The six most significant bits of the third code word are placed in the six least significant bits of the fourth octet, with the least significant of the bits from the code word as the least significant bit of the octet. The two least significant bits of the fourth code word are placed in the two most significant bits of the fourth octet, with the most significant bit of those taken from the code word as the most significant bit of the octet. The eight most significant bits of the fourth code word are placed in the fifth octet, with the most significant bit of the code word as the most significant bit of the octet.

Subsequent quadruplets of code words shall be packed in the same way into successive groups of five octets, with the eight least significant bits of the first code word of each group placed in the first octet. It is preferred that the voice sample be extended with silence such that the encoded value comprises a multiple of four of code words. However, if the voice sample comprises a number of code words which is not a multiple of four, then the last remaining code words (i.e., those in excess of an exact multiple of four) shall be discarded.

The Data component is the digital encoding of the voice, an Octet String. The encoding is specified by the value of the voice-encoding-type component of the Parameters.

For this extended body part type, extended EITs are defined for the purposes of item c in 20.4. One EIT is defined for each voice body part; it comprises the value of the voice-encoding-type component of the Parameters.

7.4.14 Report

A **Report** body part represents an MTS Report. It has a Data component.

```
report-body-part EXTENDED-BODY-PART-TYPE ::= {
       DATA {ReportDeliveryArgument IDENTIFIED BY id-et-report} }
```

The Data component is the argument of the MTS Report-delivery abstract-operation (see 8.3.1.2 of ITU-T Rec. X.411 | ISO/IEC 10021-4).

If returned-content is present in the MTS Report, it is a local matter whether it is copied into the report body part.

NOTE – Forwarding of delivery reports containing returned content may present a security risk. It is desirable that UAs generating the report body part provide an option to allow the user to choose whether to include the Returned Content.

7.4.15 Notification

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

```
notification-body-part EXTENDED-BODY-PART-TYPE ::= {
       PARAMETERS {MessageParameters IDENTIFIED BY id-ep-notification},
       DATA       {IPN IDENTIFIED BY id-et-notification} }
```

The Parameters component is identical to that for a message body part and is defined in 7.4.7.

The Data component is the *IPN*.

If *returned-ipm* is present in an *NRN*, it is a local matter whether it is copied into the notification body part.

NOTE – Forwarding of notifications containing returned IPMs may present a security risk. It is desirable that UAs generating the notification body part provide an option to allow the user to choose whether to include the Returned IPM.

7.4.16 Forwarded Content

A **Forwarded Content** body part represents a message (not necessarily an IPM) which has been previously transferred by the MTS. It has Parameters and Data components.

A family of these body parts is defined, one for each possible content-type. It is therefore possible to forward a message of any type, using the appropriate forwarded content body part.

These body parts are defined using the following template:

```
content-body-part {ExtendedContentType:content-type} EXTENDED-BODY-PART-TYPE ::= {
       PARAMETERS {ForwardedContentParameters IDENTIFIED BY
                       {id-ep-content -- concatenated with content-type --  content-type}},
       DATA       {Content IDENTIFIED BY {id-et-content -- concatenated with content-type --
content-type}}}

ForwardedContentParameters ::= SET {
       delivery-time      [0] MessageDeliveryTime OPTIONAL,
       delivery-envelope  [1] OtherMessageDeliveryFields OPTIONAL,
       mts-identifier     [2] MessageDeliveryIdentifier OPTIONAL,
       submission-proof   [3] SubmissionProof OPTIONAL}
```

To realise an instance of this family of body parts, the values of Object Identifiers to identify the Parameters and Data components are constructed by concatenating the prefixes id-ep-content and id-et-content respectively with the Object Identifier defined for the content-type of the message to be forwarded. For example, to forward a message of the inner-envelope content-type (see ITU-T Rec. X.411 | ISO/IEC 10021-4), the parameters would be identified by { id-ep-content id-cont-inner-envelope } or { 2 6 1 11 17 2 6 3 3 1 }:

```
inner-envelope-content-body-part EXTENDED-BODY-PART-TYPE ::=
        content-body-part {id-cont-inner-envelope}
```

The Parameters component comprises the following parameters:

a) **Delivery-time** (O): The date and time the original message was delivered.

b) **MTS-identifier** (O): The MTS identifier assigned to the original message.

c) **Delivery-envelope** (O): The original message's other message delivery fields.

d) **Submission-proof** (C): The proof-of-submission of the original message together with the associated certificate of the public key of the MTA which generated that proof and the message-submission-envelope, if the content represents a message previously submitted to the MTS.

```
SubmissionProof ::= SET {
        proof-of-submission             [0] ProofOfSubmission,
        originating-MTA-certificate     [1] OriginatingMTACertificate,
        message-submission-envelope         MessageSubmissionEnvelope}
```

The Data component is the Content of the original message.

When forwarding an IPM, the Message body part (see 7.4.7) should be used unless it is necessary to represent the message in precisely the form that was transferred by the MTS.

NOTE 1 – An example of a case where it may be necessary to use the forwarded content body part to forward an IPM is where the content of the original message is encrypted or secured by content-integrity-check. Use of the forwarded content body part in these cases allows the recipient to verify the security arguments.

NOTE 2 – While the components of the parameters are optional (to ease implementation in simple applications), for many applications their omission would render the body part useless.

If the original message's delivery envelope contains a message-token which contains encrypted-data then it may be necessary to create a *Forwarded Content Token* (see B.6.2) for each recipient of the forwarding IPM. This is required, for example, when an asymmetric algorithm is used for encrypted-data that contains a content-confidentiality-key.

For this extended body part type, extended EITs are defined for the purposes of item c in 20.4. One EIT is defined for each forwarded content body part; it comprises the Object Identifier which identifies the Data component.

7.4.17 PKCS7

A **PKCS7** body part represents an information object which has had some public key cryptographic service applied, typically encryption or signature. It provides a mechanism for securing an information object without using an encoding specific to any messaging system (e.g. the ASN.1 BodyPart syntax), thus allowing the secured information object to be removed from the messaging system whilst preserving its security information.

NOTE 1 – When it is required that the receiving MHS system should verify or decrypt security parameters then use of MHS security services is recommended; the PKCS7 body part is recommended when the secured object plus its security parameters are to be exported from MHS.

```
pkcs7-body-part EXTENDED-BODY-PART-TYPE ::= {
        DATA { ContentInfo IDENTIFIED BY id-et-pkcs7 } }
```

The Data component is the PKCS#7 ContentInfo type.

For this extended body part type an extended EIT is defined for the purposes of item c in 20.4; it comprises the Object Identifier which identifies the Data component.

NOTE 2 – If a PKCS7 body part contains encrypted data and uses an asymmetric key encryption algorithm, the originating IPMS-user will only be able to decrypt this data from a stored copy of the IPM if a RecipientInfo is included for the originating IPMS-user within the PKCS#7 ContentInfo. Forwarding an IPM that contains a PKCS7 body part that contains encrypted data that uses an asymmetric key encryption algorithm is unlikely to be useful, as the recipient of the forwarded IPM is unlikely to be able to decrypt the PKCS7 body part.

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 or accept an IPM.

```
NRN ::= IPN (WITH COMPONENTS {
               ... ,
               choice (WITH COMPONENTS { non-receipt-fields PRESENT }) })
```

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 COMPONENTS {
               ... ,
               choice (WITH COMPONENTS { receipt-fields PRESENT }) })
```

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

```
ON ::= IPN (WITH COMPONENTS {
               ... ,
               choice (WITH COMPONENTS
               { other-notification-type-fields PRESENT }) })
```

This version of the Specification defines two ONs, which are instances of *advice notification*. Additional uses of ON may be defined in future versions of the Specification 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 NRN or RN, and one or more ONs which shall be conveyed to the subject IPM's originator alone.

NOTE 1 – An IPM originator may receive an ON indicating that the ON's originator is temporarily absent, and subsequently receive an RN when the IPM is eventually received.

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, Blind Copy, and Circulation List Recipients heading fields, and Blind Copy Recipients envelope field. 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 OR-descriptor whose Formal-name component is present and has as its value the value of either the OR-name from the first element of the redirection-history argument (if present) or else the this-recipient-name argument of the Message Delivery operation.

NOTE 2 – In the case where the subject IPM has been delivered as a result of DL Expansion, an IPN is not generated. This is in order to prevent inadvertent disclosure of the membership of the DL (which should be the subject of the DL policy). However, IPNs may be generated in the case of Redirection or alias OR-names.

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 notification type field**: A notification field applicable to ONs alone.

The structure of an IPN is depicted in Figure 2.

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-intended-recipient   [2] IPMIntendedRecipientField OPTIONAL,
       conversion-eits              ConversionEITsField OPTIONAL,
       notification-extensions  [3] NotificationExtensionsField OPTIONAL}
```

8.1.1 Subject IPM

The **Subject IPM** common field (M) contains the value of the This IPM field of 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 OR-descriptor.

```
IPNOriginatorField ::= ORDescriptor
```

If the IPN's originator is a intended recipient of the subject IPM, the OR-descriptor above shall be precisely that which is the value of the Recipient component of the subject recipient specifier.

8.1.3 IPM Intended Recipient

The **IPM Intended Recipient** common field (C) identifies the originally specified recipient which gave rise to the subject IPM's delivery to the IPN's originator. It comprises an OR-descriptor.

```
IPMIntendedRecipientField ::= ORDescriptor
```

The OR-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, the OR-address of the IPN's originator is different from that of the subject recipient specifier, i.e. when the IPN's originator received the message as a result of redirection, or where the subject recipient specifier contained another, non-preferred, OR-address of the same user.

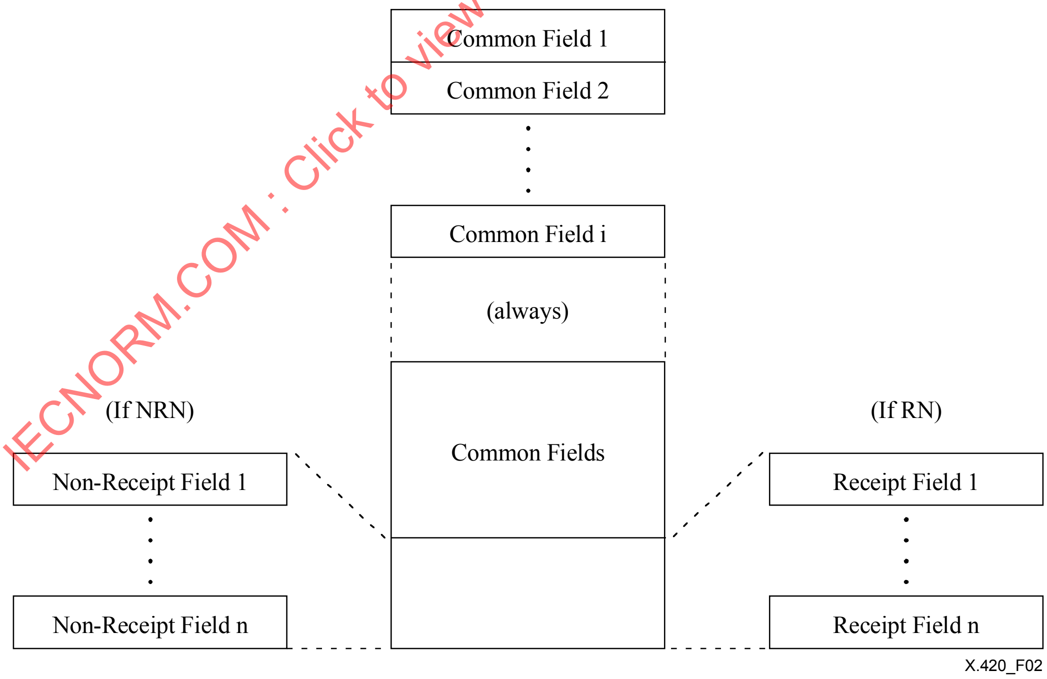

Figure 2 – An Interpersonal Notification

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 {{ NotificationExtensions }}

        NotificationExtensions IPMS-EXTENSION ::= {
                ipn-security-response |
                PrivateIPMSExtensions, ... }
```

Notification extensions for IPMS Security are defined in Annex B. There are no other notification extensions defined in this version of the Specification.

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).

[*ITU-T version:*

```
        DiscardReasonField ::= ENUMERATED {
                ipm-expired                  (0),
                ipm-obsoleted                (1),
                user-subscription-terminated (2),
                not-used                     (3) }
```

| *ISO/IEC version:*

```
        DiscardReasonField ::= ENUMERATED {
                ipm-expired                  (0),
                ipm-obsoleted                (1),
                user-subscription-terminated (2),
                -- The following value may not be supported by
                -- implementations of earlier versions of this Specification
                ipm-deleted                  (3),
                ... }
```

]

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.

[*ITU-T version:*

d) *not-used*: Implementations shall not generate this value. However, since this value is used by implementations conforming to ITU-T Recs. of the X.400-series | ISO/IEC 10021 to indicate *ipm-deleted*, messages may be encountered that contain this value.

| *ISO/IEC version:*

d) *ipm-deleted*: The subject IPM was deleted before receipt occurred. Where a message store is involved, deletion occurred before the IPM's status became *processed*.

]

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 L), 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-return* 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 {{ NRNExtensions }}

NRNExtensions IPMS-EXTENSION ::= {
        PrivateIPMSExtensions, ... }
```

There are no NRN Extensions defined in this version of the Specification.

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 ITU-T Rec. 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 {{ RNExtensions }}

RNExtensions_IPMS-EXTENSION ::= {
        PrivateIPMSExtensions, ... }
```

There are no RN Extensions defined in this version of the Specification.

8.4 Other Notification Type Fields

The other notification type fields are defined and described below.

```
OtherNotificationTypeFields ::= SET OF IPMSExtension {{ OtherNotifications }}

OtherNotifications IPMS-EXTENSION ::= {
        AdviceNotifications |
        PrivateIPMSExtensions, ... }

AdviceNotifications IPMS-EXTENSION ::= {
        absence-advice |
        change-of-address-advice,
        ... }
```

An **advice notification** (AN) is a type of ON that reports its originator's (temporary) absence, or (permanent) change of address. The *absence advice* and *change of address advice* fields together constitute the advice notification fields.

NOTE – The Absence Advice and Change of Address Advice fields may both be present in the same AN.

8.4.1 Absence Advice

The **Absence Advice** advice notification field (O) indicates that the AN's originator is unable to receive the IPM at the present time. It contains information previously supplied for the purpose by the AN's originator, and comprises an IPM body part, and, optionally, the time at which the user may next be available to take receipt of IPMs. The value of this field shall be that of the corresponding field of the registered IPM auto-advise auto-action responsible for causing the generation of the AN.

```
absence-advice IPMS-EXTENSION ::= {
        VALUE AbsenceAdvice, IDENTIFIED BY id-on-absence-advice}

AbsenceAdvice ::= SEQUENCE {
        advice           BodyPart OPTIONAL,
        next-available   Time OPTIONAL
        -- at least one component shall be present --  }
```

8.4.2 Change of Address Advice

The **Change of Address Advice** advice notification field (O) indicates a change of address for the AN's originator. It contains information previously supplied for the purpose by the AN's originator, and comprises an OR-descriptor, and, optionally, the time from which the new address will be effective. The value of this field shall be that of the corresponding field of the registered IPM auto-advise auto-action responsible for causing the generation of the AN.

```
change-of-address-advice IPMS-EXTENSION ::= {
       VALUE ChangeOfAddressAdvice, IDENTIFIED BY id-on-change-of-address-advice}

ChangeOfAddressAdvice ::= SEQUENCE {
       new-address      [0] ORDescriptor (WITH COMPONENTS {
                                 ... ,
                                 formal-name PRESENT }),
       effective-from   [1] Time OPTIONAL }
```

NOTE – For example, Change of Address Advice may be used where the Redirection of incoming messages element of service is not available.

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 ITU-T Rec. X.402 | ISO/IEC 10021-2.

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**).

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

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.

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

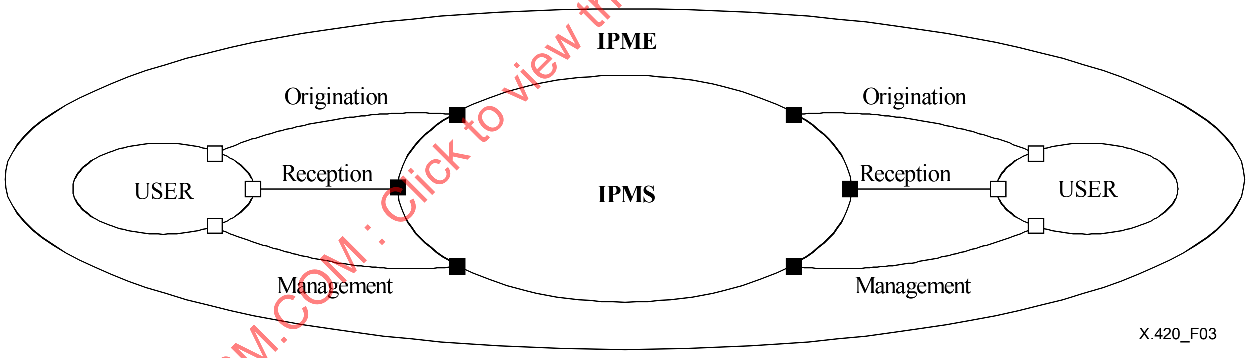

Figure 3 – The Interpersonal Messaging Environment

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 two.

```
ipms-user MHS-OBJECT ::= {
        INITIATES    {ipms-access-contract}
        ID           id-ot-ipms-user }

ipms-access-contract CONTRACT ::= {
        INITIATOR CONSUMER OF {origination | reception | management} }
```

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 Specification 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 Specification 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 MHS-OBJECT ::= {
        RESPONDS    {ipms-access-contract}
        ID          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 to follow, 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 two. 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 two. 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 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 two. 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 to follow, 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 {originate-probe | originate-ipm | originate-rn |
originate-on}
       ID               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.

```
originate-probe ABSTRACT-OPERATION ::= {
       ARGUMENT SET {
              envelope [0] ProbeSubmissionEnvelope,
              content   [1] IPM}
       RESULT SET {
              submission-identifier [0] ProbeSubmissionIdentifier,
              submission-time       [1] ProbeSubmissionTime}
       ERRORS {
              subscription-error |
              recipient-improperly-specified} }
```

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 OR-names of the intended 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:

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

b) **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.

```
originate-ipm ABSTRACT-OPERATION ::= {
       ARGUMENT SET {
              envelope [0] MessageSubmissionEnvelope,
              content   [1] IPM}
       RESULT SET {
              submission-identifier [0] MessageSubmissionIdentifier,
              submission-time       [1] MessageSubmissionTime}
       ERRORS {
              subscription-error |
              recipient-improperly-specified} }
```

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 OR-names of the intended 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:

a) **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.

12.1.3 Originate RN

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

```
originate-rn ABSTRACT-OPERATION ::= {
        ARGUMENT SET {
            envelope [0] MessageSubmissionEnvelope,
            content  [1] RN}
        RESULT SET {
            submission-identifier [0] MessageSubmissionIdentifier,
            submission-time       [1] MessageSubmissionTime}
        ERRORS {
            subscription-error |
            recipient-improperly-specified} }
```

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.

NOTE – If more than one copy of the same subject IPM is delivered to a recipient then an RN may be originated for each copy received.

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). Priority shall be that of the subject IPM.

   ii) The OR-name of the intended recipient (the Originator from the subject IPM) and the per-recipient options desired for each.

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

This abstract operation has the following results:

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

b) **Submission-time** (M): The date and time of submission as returned by the MTA.

12.1.4 Originate ON

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

```
originate-on ABSTRACT-OPERATION ::= {
       ARGUMENT SET {
              envelope [0] MessageSubmissionEnvelope,
              content  [1] ON}
       RESULT SET {
              submission-identifier [0] MessageSubmissionIdentifier,
              submission-time       [1] MessageSubmissionTime}
       ERRORS {
              subscription-error |
              recipient-improperly-specified} }
```

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).

   ii) The OR-name of the intended recipient (the Originator from the subject IPM) and the per-recipient options desired for each.

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

This abstract operation has the following results:

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

b) **Submission-time** (M): The date and time of submission as returned by the MTA.

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 { receive-report | receive-ipm | receive-rn | receive-nrn |
                           receive-on }
        ID                 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.

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

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 ON that was originated with the Originate ON abstract operation.

e) 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:

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

b) **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.

```
receive-ipm ABSTRACT-OPERATION ::= {
        ARGUMENT SET {
                envelope [0] MessageDeliveryEnvelope,
                content  [1] IPM} }
```

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.

```
receive-rn ABSTRACT-OPERATION ::= {
        ARGUMENT SET {
                envelope [0] MessageDeliveryEnvelope,
                content  [1] RN} }
```

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.

```
receive-nrn ABSTRACT-OPERATION ::= {
        ARGUMENT SET {
                envelope [0] MessageDeliveryEnvelope,
                content  [1] NRN} }
```

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

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

```
receive-on ABSTRACT-OPERATION ::= {
        ARGUMENT SET {
                envelope [0] MessageDeliveryEnvelope,
                content  [1] ON} }
```

This abstract operation has the following arguments:

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

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

This abstract operation has no results.

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 { change-auto-discard | change-auto-acknowledgment |
                           change-auto-forwarding }
        ID               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.

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

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.

```
change-auto-acknowledgment ABSTRACT-OPERATION ::= {
      ARGUMENT SET {
            auto-acknowledge-IPMs [0] BOOLEAN,
            auto-acknowledge-suppl-receipt-info [1]
                                        SupplementaryInformation OPTIONAL}
      ERRORS {
            subscription-error} }
```

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.

```
change-auto-forwarding 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}
      ERRORS {
            subscription-error |
            recipient-improperly-specified} }
```

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 OR-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.

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

This abstract error has the following parameters:

a) **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 OR-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 Specification. 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.

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. 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.

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 ITU-T Rec. X.402 | ISO/IEC 10021-2.

NOTE 2 –CCITT Recommendation T.330 effectively extends the abstract service of Interpersonal Messaging by its definition of a *miscellanea* port, which is not shown in the figure. See the note in 16.3.

NOTE 3 – The MTS supplies import and export ports. However, since those ports are not formally defined (in ITU-T 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 two.

The IPMS comprises any number of IPMS UAs.

NOTE – For brevity, the term "UA" is used throughout the rest of this Specification 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 two.

The IPMS comprises any number of IPMS-MSs.

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 two.

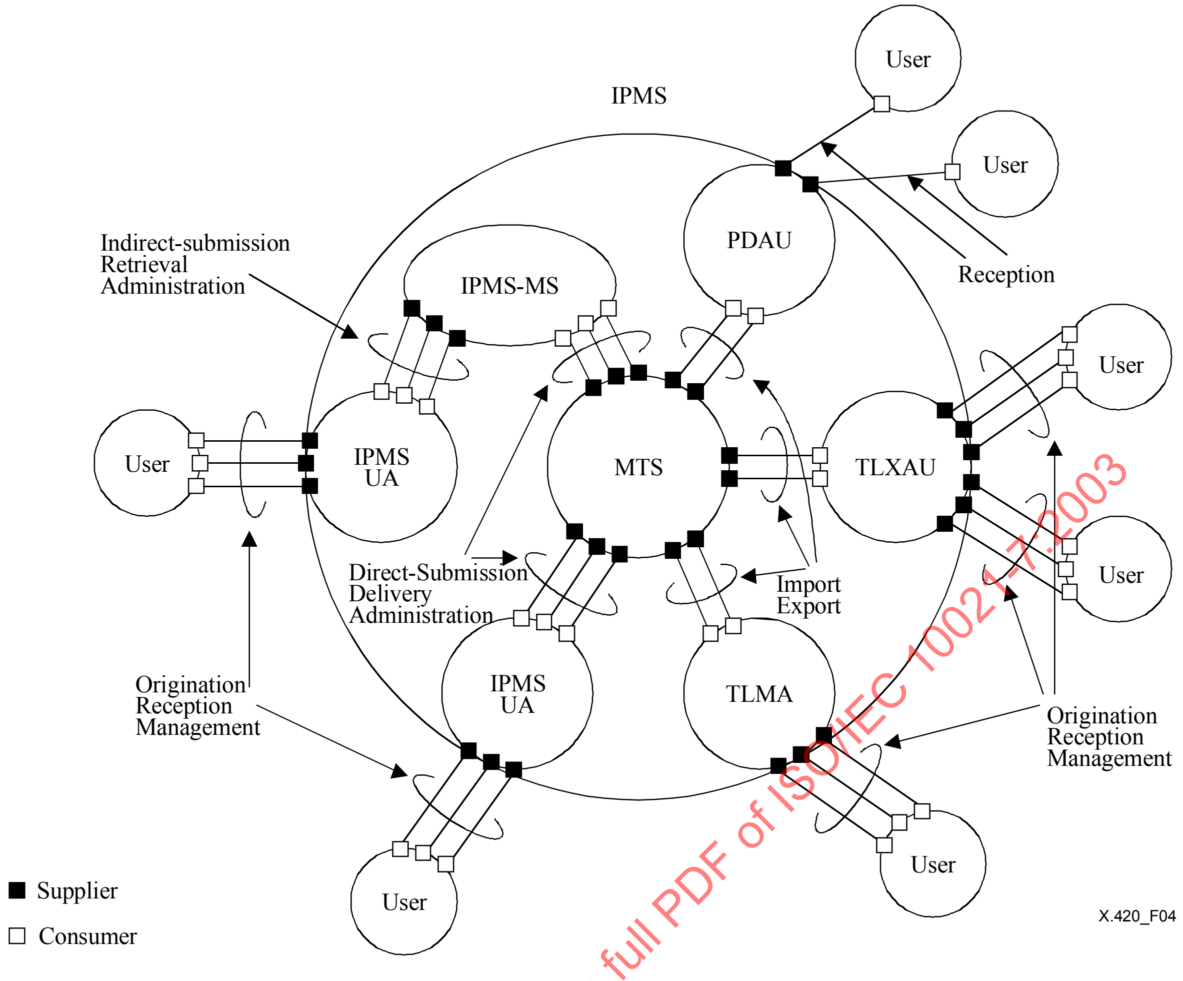

Figure 4 – The Interpersonal Messaging System

The IPMS comprises any number of TLMAs.

NOTE 1 – TLMA consumes import and export ports. However, since those ports are not formally defined (in ITU-T 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 CCITT Recommendation T.330. It is not part of the IPMS Abstract Service in its most general form, which is the subject of this Specification. 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 two.

The IPMS comprises any number of TLXAUs.

NOTE – A TLXAU consumes import and export ports. However, since those ports are not formally defined (in ITU-T 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 PDS. It helps them receive (but not originate) messages containing information objects of the types defined in section two.

The IPMS comprises any number of PDAUs.

Note – A PDAU consumes import and export ports. However, since those ports are not formally defined (in ITU-T 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 two 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 ITU-T 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 ITU-T 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 two.

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 two.

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 two.

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 two.

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 two.

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 Specification.

NOTE 1 – It is for historical reasons that the Specification 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 or 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 Specification.

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:

a) *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 OR-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 clauses 20.2-20.4.

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:

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

b) *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 OR-name of the UA's user.

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

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

iv) *Blind-copy-recipients* and *Disclosure-of-other-recipients*: If blind copy recipients are specified in the envelope then Disclosure-of-other-recipients shall have the value *disclosure-of-other-recipients-prohibited* (either explicitly or by default), and the Blind Copy Recipients heading field shall be absent within the Content.

v) *Priority*: If the *Precedence* field is supported then the priority value shall be selected based on the precedence values determined from the Originate IPM's Content argument. The precedence values assigned to all the recipients of a IPM shall affect the priority attribute in the submission envelope. Precedence values in the range of 0 to 42 shall correspond to a priority value of *non-urgent*. Precedence values in the range of 43 to 85 shall correspond to a priority value of *normal*. Precedence values in the range of 86 to 127 shall correspond to a priority value of *urgent*. If different priority values are indicated by the various specific precedence values selected for each recipient, the originating UA may perform one of the following actions: the UA may submit a single message using the highest priority value indicated by any of the precedence values; otherwise, the UA may make multiple submissions of the message, each with a different priority value.

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 This IPM heading field of the IPM shall contain the same value for each instance of such a multiple submission.

NOTE – An alternative to the multiple submissions required by the Blind Copy Recipients heading field is a single submission with the blind copy recipients encoded in the Blind-copy-recipients per-recipient field in the envelope.

If the *Precedence* field is supported, and if different priority values are indicated by the various specific precedence values selected for each recipient, then the UA may invoke Message Submission twice or three times using different priority values (see A.2.2). If the UA makes multiple submissions in this instance, then the Content argument shall be identical on each occasion, and thus the same value shall be used in the This IPM heading field for each submission.

The results of Originate IPM shall be as follows:

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

b) *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 may be the subject of future standardization.

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 OR-name of the UA's user.

ii) *Content-type* and *Original-encoded-information-types*: Determined from the RN as specified in clauses 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 may be set to type 1.

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:

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

b) *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 may be the subject of future standardization.

18.2.4 Originate ON

A UA shall perform the Originate ON 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 ON's Envelope argument:

i) *Originator-name*: The OR-name of the UA's user.

ii) *Content-type* and *Original-encoded-information-types*: Determined from the ON as specified in clauses 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 may be set as appropriate.

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

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

The results of Originate ON shall be as follows:

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

b) *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 ON's results is proper and may be the subject of future standardization.

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 Specification.

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 OR-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).

18.4.5 Receive ON

Whenever the MTS invokes Message Delivery at a UA's delivery port, and its Content argument encodes an ON as specified in 20.1, the UA shall invoke the Receive ON 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 ON).

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 clause, 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 non-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 return of the IPM 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 may be set to type 2, 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:

a) 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 Intended Recipient*: The Recipient component of the IPM's subject recipient specifier, unless its Formal-name component is the OR-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:

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

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

c) *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) *Priority*: 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 may be set to type 1.

   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 OR-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 7.4.7.

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 OR-name of the UA's user.

   ii) *Content-type* and *Original-encoded-information-types*: Determined from the IPM as specified in clauses 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 OR-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 (see 18.5.2.1).

The NRN shall have the following non-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 return of the IPM 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 may be set to type 2, its Content argument determined from the NRN as specified in 20.1.

19 Message Store Operation

ITU-T Rec. X.413 | ISO/IEC 10021-5 defines the abstract service for a general content-independent Message Store (MS). This is an optional component in MHS, whose purpose is to provide a continuously available storage medium to take delivery of messages on the UA's behalf and to enable their subsequent retrieval by the UA. In addition, the MS provides the UA with facilities for the storage of submitted messages, the classification of stored messages, the correlation of reports with the messages to which they refer, the modification by the MS-user of certain attributes of stored messages, and the logging of submission and delivery operations. The MS can also perform certain predefined auto-actions on the MS-user's behalf.

All the entry-classes, abstract-operations, general attribute-types and general auto-actions defined in ITU-T Rec. X.413 | ISO/IEC 10021-5 are available for use in Interpersonal Messaging.

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.

NOTE 1 – Because the MS is an optional system component in MHS, use of the word "shall" with respect to MS specifications should not be construed as mandating the provision of an MS or the services it provides. Use of the word "shall" with respect to MS specifications should be construed as mandating the specifications of an MS, if one is provided, and the relevant service component is supported.

NOTE 2 – In this Specification the description of the IPMS-MS abstract-service assumes that all defined entry-classes are available for use. In practice, the behaviour of a given IPMS-MS implementation will depend on its support for optional service components (e.g., the optional entry-classes, attribute-types, matching-rules, and auto-actions) and on subscription.

NOTE 3 – Several new service components have been introduced in the 1994 edition of this Specification. However, all basic and essential optional requirements defined for the IPMS Message Store are the same as those defined for editions published prior to 1994. Consequently, all enhanced facilities introduced in the 1994 edition are additional optional.

19.1 Binding to the IPMS-MS

The IPMS-MS-user binds to the IPMS-MS as described in 7.1 of ITU-T Rec. X.413 | ISO/IEC 10021-5. The following should be noted when using the MS for Interpersonal Messaging.

19.1.1 MS-Bind-argument

The following components of the fetch-restrictions parameter defined in 7.1.1 of ITU-T Rec. X.413 | ISO/IEC 10021-5 have particular significance in this Specification:

a) *Allowed-content-types*: The names of the Object Identifiers for the IPM content types defined in this Specification are id-mct-p2-1984 and id-mct-p2-1988. See Annex C.

b) *Allowed-EITs*: The names of the Object Identifiers for the encoded-information-types defined in this Specification are enumerated in Annex C. See also 20.4.

NOTE – An extension to the MS-Bind abstract-operation for the IPMS-MS is defined in 19.5.1.

19.1.2 MS-Bind-result

The **available-auto-actions** parameter defined in 7.1.2 of ITU-T Rec. X.413 | ISO/IEC 10021-5 has particular significance in this Specification. Where this indicates support for the IPM auto-forward auto-action, this shall operate as defined in 19.8.2; where support for the IPM auto-acknowledgement auto-action is indicated, this shall operate as defined in 19.8.3; where support for the IPM auto-correlate auto-action is indicated, this shall operate as defined in 19.8.4; where support for the IPM auto-discard auto-action is indicated, this shall operate as defined in 19.8.5.

Where the **Assembly Capability** extension defined in 19.5.2 indicates support for IPM-assembly-instructions, this shall operate as defined in 19.5.3.1.

19.2 Creation of Information Objects

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

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

b) The IPMS-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 object (recursively), where the forwarded object is conveyed as a Message, Forwarded Content, Notification, or Report body part.

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

19.2.1 Mapping an IPMS Message to an MS entry

When an IPM or IPN is stored in the IPMS-MS, a corresponding MS entry is created in the appropriate entry-class. The attributes of such an entry are derived from various sources:

a) some attributes, such as Sequence-number and Creation Time, are generated by the MS for administrative purposes;

b) some attributes are derived from components of the MHS Envelope;

c) some attributes summarize the contents of the IPM;

d) some attributes are derived from the Heading fields of the IPM;

e) some attributes are derived from the body parts of the IPM;

f) some attributes are derived from the component fields of the IPN;

g) some attributes correlate IPMs and IPNs with other messages to which they are related;

h) some attributes are created by the IPMS-MS-user by means of the Modify abstract-operation.

Besides these direct mappings, the IPMS-MS shall also create attributes corresponding to the complete Envelope, the complete Content, and the complete IPM Heading. Thus the same information may be logically present in more than one attribute.

If the Message-log entry-class is supported, a Message-log entry is created for each Stored-message main-entry. Message-log child-entries are not created.

Figure 5 illustrates the mapping of an IPM to an MS entry.

19.2.2 Mapping of forwarding messages in the IPMS-MS

The IPMS-MS shall model a forwarding IPM as a main-entry with one child-entry for each forwarded object (i.e., Message, Forwarded Content, Notification, or Report body part) the message contains. A simple illustration of this mapping is shown in Figure 6.

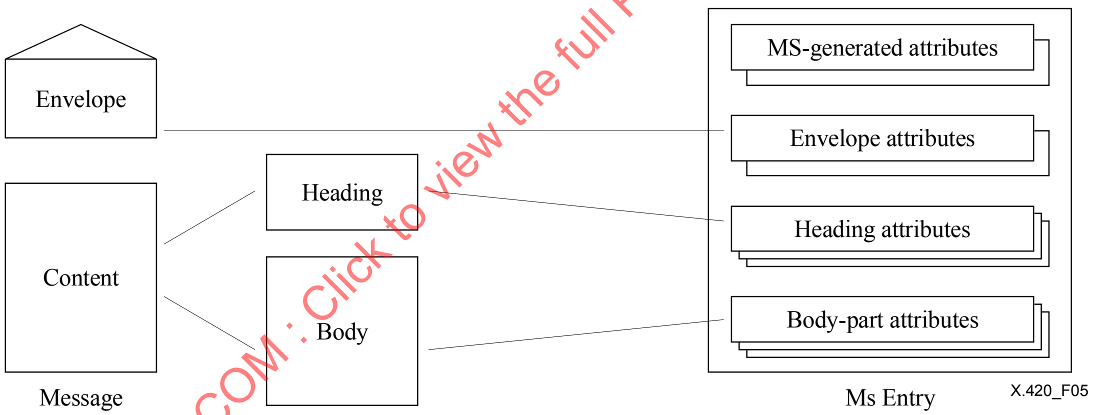

Figure 5 – Mapping an IPM to an Ms entry

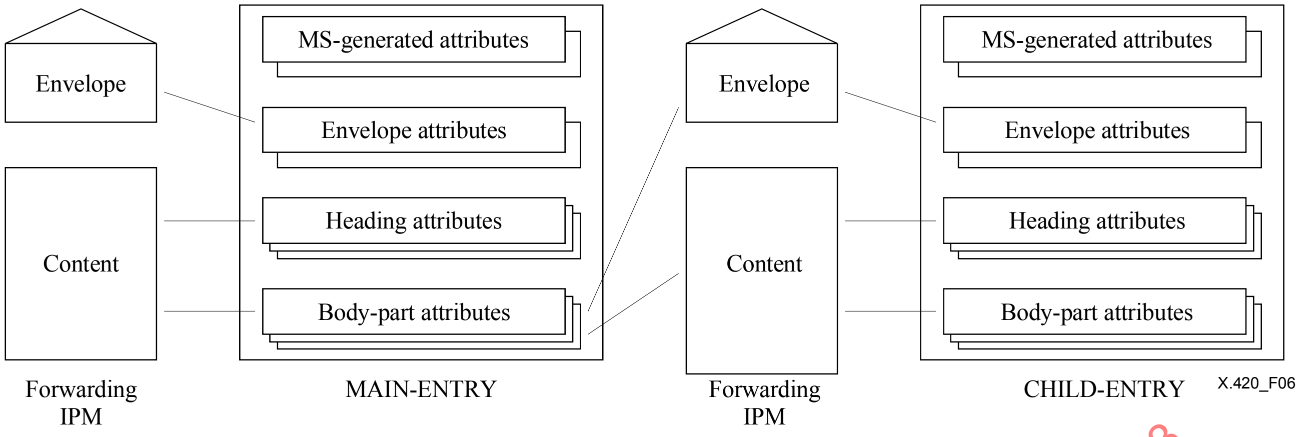

Figure 6 – Mapping a Forwarding message to an IPMS-MS entry

19.2.3 Presence of General-attributes in child-entries

The following general attribute-types shall be present in child-entries of an IPM or NRN when stored in an entry-class for which the attribute is defined: content-length, content-type, creation-time, entry-type, parent-sequence-number, retrieval-status, sequence-number. The absence of a delivery envelope precludes the generation of other general attributes which are mandatory in Table 2 of ITU-T Rec. X.413 | ISO/IEC 10021-5 for the following types of child-entry:

a) the Returned IPM optionally present in an NRN;

b) the Message body part (i.e. the forwarded IPM) of a forwarding IPM where the Parameters component of the body part is empty;

c) the Notification body part of a forwarding IPM where the Parameters component of the body part is empty;

d) the Forwarded Content body part of a forwarding IPM where the Parameters component of the body part is empty.

In the case where a child-entry is generated from an IPM's Message or Notification body part in which the Parameters component is present:

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

b) 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.

Where a child-entry is generated from an IPM's Report body part the general attribute-types which are mandatory for a Report in Table 2 of ITU-T Rec. X.413 | ISO/IEC 10021-5 shall be present.

Where a child-entry is generated from an IPM's Forwarded Content body part the content general attribute-type shall be present. Additionally, content-specific attribute-types appropriate to that child-entry's content-type shall be present if that content-type is supported by the MS, provided that the content is not encrypted.

In the case where a child-entry is generated from an IPM's Forwarded Content body part in which the Parameters component is present:

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

b) if MTS-identifier is present in Parameters then the message-delivery-identifier general-attribute-type shall be present;

c) if Delivery-envelope is present in Parameters then all the other mandatory general-attribute-types defined for a delivered-message entry (except for message-delivery-envelope) shall be present;

d) if Delivery-time, MTS-identifier and Delivery-envelope are all present in Parameters then the message-delivery-envelope general-attribute-type shall be present;

e) if Submission-proof is present in Parameters then the proof-of-submission, originating-MTA-certificate, and message-submission-envelope general-attribute-types shall be present.

The entry-type general-attribute of child-entries in the Delivery and Delivery-log entry-classes shall have the value *delivered-message*, except those containing returned content which shall have the value *returned-content*. The entry-type general-attribute of child-entries present in the Submission and Submission-log entry-classes shall have the value *submitted-message*.

The example in Table 2 illustrates the use of child-entries in the Delivery entry-class. 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.

Table 2 – Example of the use of child-entries

Sequence number	Entry-type	Child sequence numbers	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

19.3 Maintenance of Attributes

An IPMS-MS shall satisfy the following requirements related to the MS attributes which it supports:

a) For each IPM or IPN it holds, including the child-entry of a delivery report containing Returned-content, the IPMS-MS shall support the attributes defined in 19.6.

b) For each body part type present in a stored IPM, the IPMS-MS shall maintain an Extended body part attribute (and, when appropriate, an attribute corresponding to the Parameters component of that body part type) such that it contains all body parts of that type regardless of whether they were conveyed to the IPMS-MS as basic or Extended body parts of the IPM.

c) For each IPM it holds, the IPMS-MS shall give the following meanings to the defined values of the MS retrieval-status general-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, or the Body attribute, or the Content general-attribute) have been conveyed to the UA.

NOTE – The IPMS-MS-user may use the Modify abstract-operation to change the value of the retrieval-status attribute. See 11.2.68 of ITU-T Rec. X.413 | ISO/IEC 10021-5.

d) For each IPN it holds, the IPMS-MS shall give the following meanings to the defined values of the MS retrieval-status general-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.

e) When the MS retrieval-status general-attribute is retrieved in the result of an abstract-operation, the value returned shall reflect the state of affairs prior to the execution of that abstract operation.

f) The performance of the IPM auto-forward auto-action may cause the MS retrieval-status general-attribute of the auto-forwarded entry to be set to *processed*, see 19.8.2.

g) The content-type attribute of each (message containing an) IPM or IPN that is stored in the IPMS-MS shall have the value id-mct-p2-1984 or id-mct-p2-1988 (see annex C), as appropriate, depending upon the content type of the message (see 20.2).

The general (content independent) attributes that may occur in the MS entry-classes are documented in ITU-T Rec. X.413 | ISO/IEC 10021-5. All content independent MS attributes can be used for the content defined in this Specification. The IPMS specific attributes are defined in 19.6. All general attribute types classified as mandatory in Tables 2 and 3 of ITU-T Rec. X.413 | ISO/IEC 10021-5 shall be supported.

19.4 Notification of Non-receipt

When it deletes an IPM while performing the Delete abstract-operation or the Auto-delete auto-action of the MS Abstract Service, the IPMS-MS shall generate an NRN if, and only if, one is requested by means of the Notification-requests component of the subject recipient specifier of the deleted IPM, and the IPM's entry-status has the value *listed*. In the case of the Delete abstract-operation, the NRN shall not be generated if prevent-NRN-generation is specified in the delete-extensions parameter of the Delete abstract-operation which deletes the IPM (see 19.5.6).

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

The NRN shall have the following non-receipt fields:

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

b) *Discard reason*: [Omitted|The value *ipm-deleted*].

c) *Auto-forward comment*: Omitted.

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

The IPMS-MS shall submit the NRN by invoking MS-message-submission. Message Submission's Envelope argument shall be as prescribed for auto-acknowledgement (see 18.5.2.2), except that notification-type may be set to type 2, its Content argument determined from the NRN as specified in 20.1. If the IPM auto-correlate auto-action is in effect then the IPMS-MS shall add the sequence-number of the submitted IPN to the AC Submitted IPNs attribute of the entry representing the deleted IPM in the Message-log entry-class; in addition, that entry's AC Submitted IPN Status attribute is given the value *ipm-discarded*.

19.5 IPMS-MS abstract-operation extensions

The MS abstract-service defined in ITU-T Rec. X.413 | ISO/IEC 10021-5 provides a general mechanism for extending various abstract-operations and errors, in order to satisfy additional requirements specific to particular content-types. The extensions used by the IPMS-MS are defined below.

With the exception of the forwarding-request extension (see 19.5.5), each extension is defined as an instance of the MS-EXTENSION information object class (see 6.6 of ITU-T Rec. X.413 | ISO/IEC 10021-5).

19.5.1 MS-Bind extensions

The IPMS-MS-user may make use of the bind-extensions parameter of the MS-Bind abstract-operation (see 7.1.1 of ITU-T Rec. X.413 | ISO/IEC 10021-5) to cause the suspension of the IPM auto-acknowledgement auto-action, as defined in 19.8.3. The suspend-auto-acknowledgement information object is defined as follows:

```
suspend-auto-acknowledgement MS-EXTENSION ::= {
       NULL  IDENTIFIED BY  id-mst-suspend-auto-acknowledgement }
```

The presence of this object in the bind-extensions parameter of the MS-Bind abstract-operation causes the suspension of the IPM auto-acknowledgement auto-action for entries whose retrieval-status becomes *processed* during the abstract-association. There are no parameters. Where an IPMS-MS does not provide the IPM auto-acknowledgement auto-action, it shall ignore the presence of the suspend-auto-acknowledgement bind-extension.

NOTE – Where a UA itself generates RNs, it should select suspend-auto-acknowledgement, to avoid the interference which could arise if the user employs another UA which, by registration, has activated the IPM-auto-acknowledgement auto-action.

19.5.2 MS-Bind-Result extensions

The **Assembly Capability** extension indicates by its presence in the MS-bind-result that the IPMS-MS supports IPM-assembly-instructions (see 19.5.3.1) within IPM-submission-options (see 19.5.3). It shall be present when requested by means of the MS-configuration-request parameter of MS-bind.

```
assembly-capability MS-EXTENSION ::= {
       NULL IDENTIFIED BY id-mst-assembly-capability }
```

19.5.3 IPM -submission options

The submission-options parameter of the MS-message-submission abstract-operation defined in 8.3.1.1 and 8.1.6 of ITU-T Rec. X.413 | ISO/IEC 10021-5 allows for the specification of IPMS-specific extensions.

The IPM-submission-options information object set comprises the MS-message-submission-argument extensions defined for the IPMS-MS:

```
IPMSubmissionOptions MS-EXTENSION ::= {
       ipm-assembly-instructions |
       originator-body-part-encryption-token |
       originator-forwarded-content-token,
       . . .  -- For future extension additions --  }
```

19.5.3.1 IPM assembly instructions

This IPM submission option is used to request the incorporation of stored IPMs and stored body parts in a submitted IPM.

The **IPM-assembly-instructions-** information object is defined as follows:

```
ipm-assembly-instructions MS-EXTENSION ::= {
       IPMAssemblyInstructions  IDENTIFIED BY  id-mst-assembly-instructions }

IPMAssemblyInstructions ::= SET {
       assembly-instructions  [0]  BodyPartReferences }

BodyPartReferences ::= SEQUENCE OF BodyPartReference

BodyPartReference ::= CHOICE {
       stored-entry           [0]  SequenceNumber,
       stored-content         [1]  SequenceNumber,
       submitted-body-part    [2]  INTEGER (1..MAX),
       stored-body-part       [3]  SEQUENCE {
            message-entry          SequenceNumber,
            body-part-number       INTEGER (1..MAX) } }
```

The single component of IPM-assembly-instructions has the following meaning:

Assembly-instructions (M): This component instructs the IPMS-MS to assemble stored body parts or stored entries with the present submitted IPM, before submitting the resulting IPM to the MTS (or storing it as a draft-message entry). The IPMS-MS shall construct the new Body by assembling body parts in the order specified in the argument, i.e. the sequence of body parts which forms the new Body is determined by the sequence of body-part-references. If **stored-entry** is specified, it may identify an IPM, IPN, or Report. The new body part constructed from the stored-entry will be, respectively, a Message body part, a Notification body part, or a Report body part. If **stored-content** is specified, the new body part constructed from the identified entry will be a Forwarded Content body part. If **submitted-body-part** is specified, the new body part is a body part of the present submitted IPM (identified by number). If **stored-body-part** is specified, the new body part is copied from the entry identified by message-entry, with the body-part-number indicated. Body parts are numbered starting at '1'.

In a Message body part constructed from a stored IPM that represents a delivered-message entry, the Parameters component shall contain delivery-time and delivery-envelope.

In a Message body part constructed from a stored IPM that represents a submitted-message entry, the Parameters component shall not contain delivery-time and shall contain delivery-envelope. This simulated delivery-envelope shall not contain originally-intended-recipient-name, converted-encoded-information-types, nor any extension whose presence is not defined in both a Message Submission envelope and a Message Delivery envelope. The originator-name and this-recipient-name components of this delivery-envelope each contain the OR-address of the IPM's originator.

In a Message body part constructed from a stored IPM that represents a draft-message entry, the Parameters component shall not contain delivery-time or delivery-envelope.

In a Forwarded Content body part constructed from a stored IPM, the Parameters component shall contain delivery-time and delivery-envelope as prescribed above for a Message body part, and shall also contain MTS-identifier except where the stored IPM represents a draft-message entry. In a Forwarded Content body part constructed from a stored IPM that represents a submitted-message entry which has a proof-of-submission and the associated originating-MTA-certificate, the Parameters component shall contain submission-proof.

When the IPMS-MS has assembled the new Body, it shall update the original-encoded-information-types in the message-submission-envelope as necessary, such that the complete message still satisfies the requirements of 20.4.

If the present submitted IPM contains a *Body Part Signature* (see A.1.4), a *Body Part Encryption Token* (see B.6.1), or a *Forwarded Content Token* (see B.6.2) then the body-part-numbers within these shall refer to the parts of the Body as they will appear after the assembly-instructions have been performed. If the entry from which a stored-body-part is copied has a body-part-signature containing one or more signatures for the identified body part, then the IPMS-MS shall append those signatures to the body-part-signature in the submitted IPM (creating a new body-part-signature if necessary) ensuring that the appropriate Certificate or Directory name is included in the *originator-certificates* component of each such copied body-part-signature.

If the present submitted IPM contains an *IPM Security Label* (see A.1.5) which contains *body-part-security-labels* then the order of components within this shall refer to the parts of the Body as they will appear after the assembly-instructions have been performed. If a referenced stored entry has an *IPM Security Label* or a *Message Security Label* then the IPMS-MS shall insert the *content-security-label* from the *IPM Security Label* (or if this is not present then the *Message Security Label*) into the appropriate component of the *body-part-security-labels* in the submitted IPM (if necessary creating a new *body-part-security-labels* or replacing a submitted component value). If the entry from which a stored-body-part is copied has an associated *body-part-security-label* then the IPMS-MS shall insert that *body-part-security-label* into the appropriate component of the *body-part-security-labels* in the submitted IPM (if necessary creating a new *body-part-security-labels* or replacing a submitted component value). If *body-part-security-labels* are inserted then the IPMS-MS shall ensure that the *content-security-label* component contains the appropriate aggregate value, or if it is unable to aggregate any of these labels then the IPMS-MS shall generate a submission error specifying a security-error of *unable-to-aggregate-security-labels*.

The IPMS-MS shall generate a submission error specifying a security-error of *assembly-instructions-conflict-with-security-services* if assembly-instructions are present with the submitted IPM together with any of: content-confidentiality-algorithm-identifier, content-integrity-check, or message-origin-authentication-check.

NOTE 1 – The presence of delivery-envelope in the Parameters component of a Message body part does not imply that the body part was derived from a delivered-message. This derivation is implied (but not verified) by the presence of delivery-time.

NOTE 2 – The assembly of body-parts from entries with content-type other than IPM is possible only for body parts whose definition is compatible with IPM (as stated in the relevant content-type Specification), or for which rules of conversion into IPM body parts are defined.

NOTE 3 – In the case where the MS-user wishes to submit a message comprising entirely body parts assembled by the MS, the argument to MS-message-submission will contain a zero-length Body.

NOTE 4 – If any of the assembled body parts contain data that has been encrypted with a symmetric encryption algorithm where the session-key for that algorithm is itself encrypted in an associated token, it is the responsibility of the IPMS-MS-user to generate appropriate tokens for each recipient of the forwarding IPM. This does not require the IPMS-MS-user to retrieve or decrypt the encrypted data in these body parts, but just to retrieve, decrypt and re-encrypt the associated tokens.

The actions performed by an IPMS-MS when the IPM-assembly-instructions parameter is present in an MS-message-submission argument are defined in 19.9.2.

19.5.3.2 Originator body part encryption token

The **Originator-body-part-encryption-token** MS-submission-extension is used where the submitted message contains a *Body Part Encryption Token* (see B.6.1) that has been encrypted such that it cannot subsequently be decrypted by the originator. This extension enables the originator to supply a *Body Part Encryption Token* constructed as if the originator were a recipient of the message, to be stored in the submitted-message entry but not submitted to the MTS. Subsequently, the originator may retrieve this information and use it to recover the encrypted body part.

```
originator-body-part-encryption-token MS-EXTENSION ::= {
        BodyPartTokens IDENTIFIED BY id-mst-originator-body-part-encryption-token }
```

19.5.3.3 Originator forwarded content token

The **Originator forwarded content token** MS-submission-extension is used where the submitted message contains a *Forwarded Content Token* (see B.6.2) that has been encrypted such that it cannot subsequently be decrypted by the originator. This extension enables the originator to supply a *Forwarded Content Token* constructed as if the originator were a recipient of the message, to be stored in the submitted-message entry but not submitted to the MTS. Subsequently, the originator may retrieve this information and use it to recover the Forwarded Content body part.

```
originator-forwarded-content-token MS-EXTENSION ::= {
        ForwardedContentToken IDENTIFIED BY id-mst-originator-forwarded-content-token }
```

19.5.4 IPM submission errors

When an IPMS-MS performs the MS-message-submission abstract-operation of ITU-T Rec. X.413 | ISO/IEC 10021-5, the IPMS-specific errors defined below may be reported. These are reported as MS-extension-errors, as defined in 9.12 of ITU-T Rec. X.413 | ISO/IEC 10021-5.

The **IPM-submission-errors** information object set comprises the submission errors defined for the IPMS-MS:

```
IPMSubmissionErrors MS-EXTENSION ::= {
        invalid-assembly-instructions |
        invalid-ipn,
        . . .  -- For future extension additions --  }
```

The **invalid-assembly-instructions** error shall be reported where the IPM assembly instructions component of IPM-submission-options is present, but the message submitted is not an IPM, or the IPM assembly instructions component contains a reference to an entry whose content-type is not compatible with IPM, or contains a reference to a body part not present in the submitted or stored message. The invalid body-part-references are reported in the error.

```
invalid-assembly-instructions MS-EXTENSION ::= {
        BodyPartReferences  IDENTIFIED BY  id-mst-invalid-assembly-instructions }
```

The **invalid-IPN error** shall be reported if the UA submits an IPN concerning a message for which an IPN has already been sent, except that an RN may be generated for an auto-forwarded IPM where an NRN indicating IPM-auto-forwarded has already been sent.

```
invalid-ipn MS-EXTENSION ::= {
        NULL  IDENTIFIED BY  id-mst-invalid-ipn }
```

19.5.5 Forwarding-request extension

The IPMS-MS provides two methods for incorporating stored IPMs in the body of a submitted IPM. If a 1994 Application Context is in use, the IPM assembly instructions extension defined in 19.5.3.1 is used for this purpose. If a 1988 Application Context is in use, the forwarding-request extension defined below is used.

If a 1988 Application Context is in use (see 5.7 of ITU-T Rec. X.413 | ISO/IEC 10021-5) then an IPMS-MS shall support the forwarding-request extension as specified in 8.3.1.1 of ITU-T Rec. X.413 | ISO/IEC 10021-5. The IPMS-MS-user may submit an IPM, including Heading and Body, using the MS-message-submission abstract-operation, and identify by means of the forwarding-request extension, a message that is already stored in the IPMS-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.

19.5.6 Delete extensions

The IPMS-MS-user may make use of the delete-extensions parameter of the Delete abstract-operation (see 8.2.4.1 of ITU-T Rec. X.413 | ISO/IEC 10021-5) to prevent the generation of an NRN when an IPM is deleted, as defined in 19.4. The prevent-NRN-generation extension is defined as follows:

```
prevent-nrn-generation MS-EXTENSION ::= {
        NULL  IDENTIFIED BY  id-mst-prevent-nrn-generation }
```

NOTE – This may be used to prevent the automatic generation of NRNs where a UA implementation itself generates NRNs.

19.6 IPMS-MS Attributes

As described in ITU-T Rec. X.413 | ISO/IEC 10021-5, an MS maintains and provides access to certain attributes 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 two).

This clause defines the MS attributes specific to Interpersonal Messaging. Each IPMS-MS attribute is defined as an instance of the ATTRIBUTE information object class (see 6.3.3.3 of ITU-T Rec. X.413 | ISO/IEC 10021-5).

All the IPMS-MS attributes defined in this Specification, except those corresponding to Extended body part types (which cannot be enumerated; see 19.6.3.3), are listed alphabetically, for reference, in the first column of Table 5 in 19.6.7. Table 3 indicates their presence in IPM, NRN, RN and ON entries of the Stored-message, Submission-log, and Delivery-log entry-classes of the MS. For entries of the Submission-log and Delivery-log entry-classes the Body attributes (see 19.6.3) shall not be present. Table 3 also indicates whether the attribute is single-valued or multi-valued, and whether it is available for retrieval by the List and Summarize abstract-operations. Rules for the presence and maintenance of general-attributes in the IPMS-MS are defined in 19.2 and 19.3. No requirements are placed on the IPMS-MS-user for the support of any of the IPMS-MS attributes.

Where a delivery report contains returned content, the child-entry so created shall possess the attributes indicated for an IPM, NRN, RN or ON as appropriate. Where an NRN contains a returned IPM, the child-entry so created shall possess the attributes indicated for an IPM. Where an IPM (whether submitted, delivered, in the returned content of a delivery report, or present in an NRN), contains a Message body part, the child-entry so created shall possess the attributes indicated for an IPM. Where an IPM (whether submitted, delivered, in the returned content of a delivery report, or present in an NRN), contains a Report body part which contains returned content, the report child-entry shall itself have a returned content child-entry which shall possess the attributes indicated for an IPM, NRN, RN or ON as appropriate. Where an IPM (whether submitted, delivered, in the returned content of a delivery report, or present in an NRN), contains a Notification body part, the child-entry so created shall possess the attributes indicated for an IPN; where the IPN is an NRN which contains a returned IPM, the notification child-entry shall itself have a returned IPM child-entry which shall possess the attributes indicated for an IPM. Where an IPM (whether submitted, delivered, in the returned content of a delivery report, or present in an NRN), contains a Forwarded Content body part which contains an IPM (which is not encrypted), the child-entry so created shall possess the attributes indicated for an IPM.

Table 3 applies to all entries except those of the Auto-action-log entry-class. There are no IPMS-specific attributes defined for the Auto-action-log entry-class. See 5.2 of ITU-T Rec. X.413 | ISO/IEC 10021-5 for an elaboration of the table's legend.

Table 3 – Summary of IPMS-specific common attributes

Attribute	S/M	Support			Presence					
	V	Sm	DI	SI	IPM	NRN	RN	ON	L	S
Acknowledgment Mode	S	O	O	O	–	–	P	–	Y	Y
Authorization Time	S	O	O	O	C	–	–	–	Y	N
Authorizing Users	M	O	M	M	C	–	–	–	Y	N
Auto-forward Comment	S	O	O	O	–	C	–	–	Y	N
Auto-forwarded	S	O	O	O	P	–	–	–	Y	Y
Auto-submitted	S	O	O	O	C	–	–	–	Y	N
Bilaterally Defined Body Parts	M	O	–	–	C	–	–	–	N	N
Blind Copy Recipients	M	O	O	M	C	–	–	–	Y	N
Body	S	M	–	–	P	–	–	–	N	N
Body Part Encryption Token	S	O	O	O	C	–	–	–	Y	N
Body Part Security Label	M	O	O	O	C	–	–	–	Y	N
Body Part Signatures Verification Status	S	O	O	–	C	–	–	–	Y	N
Body Part Signatures	S	O	O	O	C	–	–	–	Y	N
Body Part Summary*	M	O	O	O	P	–	–	–	Y	N
Circulation List Recipients	M	O	O	O	C	–	–	–	Y	N
Conversion EITs	M	O	O	O	–	C	C	C	Y	N
Copy Recipients	M	O	O	M	C	–	–	–	Y	N
Discard Reason	S	O	O	O	–	C	–	–	Y	Y
Distribution Codes	M	O	O	O	C	–	–	–	Y	N
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	O	O	C	–	–	–	Y	N
Extended Body Part Types	M	O	–	–	C	–	–	–	Y	Y
Extended Subject	S	O	O	O	C	–	–	–	Y	N
Forwarded Content Token	S	O	O	O	C	–	–	–	Y	N
Forwarding Token	S	O	–	–	C	–	–	–	Y	N
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

Table 3 – Summary of IPMS-specific common attributes (continued)

Attribute	S/M	Support			Presence				L	S
	V	Sm	DI	SI	IPM	NRN	RN	ON		
G4 Class 1 Body Parts	M	O	–	–	C	–	–	–	N	N
Heading	S	M	–	–	P	–	–	–	N	N
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	O	O	P	–	–	–	Y	Y
Incomplete Copy	S	O	O	O	C	–	–	–	Y	N
Information Category	M	O	O	O	C	–	–	–	Y	N
IPM Auto-discarded*	S	–	O	–	C	–	–	–	Y	N
IPM Entry Type	S	M	M	M	P	P	P	P	Y	Y
IPM Intended Recipient	S	O	O	O	–	C	C	C	Y	N
IPM Security Label	S	O	O	O	C	–	–	–	Y	N
IPM Synopsis	S	O	O	O	P	–	–	–	N	N
IPN Originator	S	O	O	O	–	C	C	C	Y	N
Languages	M	O	O	O	C	–	–	–	Y	N
Manual Handling Instructions	M	O	O	O	C	–	–	–	Y	N
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
Nationally Defined Body Parts	M	O	–	–	C	–	–	–	N	N
Non-receipt Reason	S	O	O	O	–	P	–	–	Y	Y
Notification Extensions*	M	O	O	O	–	C	C	C	Y	N
NRN Extensions*	M	O	O	O	–	C	–	–	Y	N
NRN Requestors	M	O	O	–	C	–	–	–	Y	N
Obsoleted IPMs	M	O	O	O	C	–	–	–	Y	N
Originator	S	O	M	O	C	–	–	–	Y	N
Originator's Reference	S	O	O	O	C	–	–	–	Y	N
Other Notification Type Fields*	M	O	O	O	–	–	–	P	Y	N
Precedence	S	O	O	–	C	–	–	–	Y	Y
Precedence Policy Identifier	S	O	O	O	C	–	–	–	Y	N
Primary Recipients	M	O	O	M	C	–	–	–	Y	N
Receipt Time	S	O	O	O	–	–	P	–	Y	N
Related IPMs	M	O	O	O	C	–	–	–	Y	N
Replied-to IPM	S	O	O	O	C	–	–	–	Y	N
Reply Recipients	M	O	O	O	C	–	–	–	Y	N
Reply Requestors	M	O	O	–	C	–	–	–	Y	N
Reply Time	S	O	O	O	C	–	–	–	Y	N
Returned IPM	S	O	O	O	–	C	–	–	Y	N
RN Extensions*	M	O	O	O	–	–	C	–	Y	N
RN Requestors	M	O	O	–	C	–	–	–	Y	N
Sensitivity	S	O	O	O	C	–	–	–	Y	Y
Subject	S	O	M	M	C	–	–	–	Y	N
Subject IPM	S	M	M	M	–	P	P	P	Y	N
Suppl Receipt Info	S	O	O	O	–	–	C	–	Y	N
Teletex Body Parts	M	O	–	–	C	–	–	–	N	N

Table 3 – Summary of IPMS-specific common attributes (concluded)

Attribute	S/M	Support			Presence				L	S
	V	Sm	DI	SI	IPM	NRN	RN	ON		
Teletex Data	M	O	–	–	C	–	–	–	N	N
Teletex Parameters	M	O	–	–	C	–	–	–	N	N
This IPM	S	M	M	M	P	–	–	–	Y	N
Videotex Body Parts	M	O	–	–	C	–	–	–	N	N
Videotex Data	M	O	–	–	C	–	–	–	N	N
Videotex Parameters	M	O	–	–	C	–	–	–	N	N

Legend:

S/M V	Single/multi valued
Support	Support level by IPMS-MS
Sm	for Stored-message entry-class
Dl	for Delivery-log entry-class
Sl	for Submission-log entry-class
Presence	Presence in each IPM-entry-type
L	Available for List, Alert
S	Available for Summarize
*	Not defined for 1988 Application Contexts

Table 4 indicates the presence of the correlation attributes (see 19.6.5) in IPM entries of the Stored-message and Message-log entry-classes. None of the attributes listed in Table 4 are defined for 1988 Application Contexts. These attributes shall not be present in entries representing IPNs.

19.6.1 Summary Attributes

Some attributes summarize an Interpersonal Messaging information object. These attributes are defined and described below.

19.6.1.1 IPM Entry Type

The **IPM Entry Type** attribute identifies an information object's type.

```
ipm-entry-type ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     IPMEntryType,
       EQUALITY MATCHING-RULE    integerMatch,
       NUMERATION                single-valued,
       ID                        id-sat-ipm-entry-type }

IPMEntryType ::= ENUMERATED {
       ipm       (0),
       rn        (1),
       nrn       (2),
       on        (3)}
```

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.

d) *on*: The information object is an ON.

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the Message-log entry for such an object) if, and only if, that object is a message whose content is an IPM or IPN.

Table 4 – Summary of IPMS-specific correlation attributes

Attribute	S/M V	Sup	Presence in			L	S
			Del IPM	Sub IPM	IPN		
AC Correlated Delivered IPNs	M	O	–	C	–	Y	N
AC Correlated Delivered Replies	M	O	–	C	–	Y	N
AC Delivered IPN Summary	M	O	–	C	–	Y	Y
AC Delivered Replies Summary	M	O	–	C	–	Y	Y
AC Forwarded IPMs	M	O	C	C	–	Y	N
AC Forwarding IPMs	M	O	C	C	–	Y	N
AC IPM Recipients	M	O	–	C	–	Y	N
AC Obsoleted IPMs	M	O	C	C	–	Y	N
AC Obsoleting IPMs	M	O	C	C	–	Y	N
AC Related IPMs	M	O	C	C	–	Y	N
AC Relating IPMs	M	O	C	C	–	Y	N
AC Replied-to IPM	M	O	C	C	–	Y	N
AC Replying IPMs	M	O	C	C	–	Y	N
AC Subject IPM	S	O	–	–	C	Y	N
AC Submitted IPN Status	S	O	C	–	–	Y	Y
AC Submitted IPNs	S	O	C	–	–	Y	N
AC Submitted Reply Status	S	O	C	–	–	Y	Y
Recipient Category	S	O	C	–	–	Y	N
Revised Reply Time	S	O	C	–	–	Y	N

Legend:

S/M V	Single/multi valued
Sup	Support level by IPMS-MS
Del IPM	Presence in delivered IPM
Sub IPM	Presence in submitted IPM
IPN	Presence in IPN
L	Available for List
S	Available for Summarize

19.6.1.2 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,
       NUMERATION                 single-valued,
       ID                         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 one 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]   INSTANCE OF TYPE-IDENTIFIER OPTIONAL,
       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 IPMS-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 the purposes of this synopsis, the body part is considered to be of type Extended, whether or not it was so conveyed to the IPMS-MS (see 7.4 for definitions of Extended body part types equivalent to the basic body part types):

a) **Type** (M): The body part's Extended type, i.e., the &id field of the body part's Data component. An Object Identifier.

b) **Parameters** (C): The body part's format and control parameters, i.e., the body part's Parameters component. An instance of TYPE-IDENTIFIER. This conditional component shall be present if, and only if, a Parameters component is defined for this type of body part.

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 ITU-T Rec. X.690 | ISO/IEC 8825-1 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 IPMS-MS's Fetch abstract operation, or has been processed by an auto-action whose definition causes the body part to become *processed*. A Boolean.

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the Message-log entry for such an object) 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.

19.6.1.3 Body Parts Summary

The **Body Parts Summary** attribute, which is multi-valued, provides a summary of the body parts present in an IPM. One value of body-part-descriptor shall be present for each body part present in the IPM. Body-part-descriptors shall appear in the same order in the attribute as body parts appear in the IPM.

```
body-parts-summary ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     BodyPartDescriptor,
       NUMERATION                multi-valued,
       ID                        id-sat-body-parts-summary }

BodyPartDescriptor ::= SEQUENCE {
       data                 [0]   OBJECT IDENTIFIER,
       parameters           [1]   OBJECT IDENTIFIER OPTIONAL,
       this-child-entry     [2]   SequenceNumber OPTIONAL,
       position             [3]   INTEGER,
       size                 [4]   INTEGER,
       processed            [5]   BOOLEAN DEFAULT FALSE }
```

For the purpose of this summary, body parts are considered to be of type Extended, whether or not they were so conveyed to the IPMS-MS. See 7.4 for definitions of the Extended body part types equivalent to the basic body part types.

The body-part-descriptor has the following components:

a) **Data** (M): The body part's Extended type, i.e. the &id field of the body part's Data component (see 19.6.3.3). This Object Identifier identifies the attribute generated for the Data component of the body part.

b) **Parameters** (C): The Extended type of the body part's Parameters, i.e. the &id field of the body part's Parameters component (see 19.6.3.3). This Object Identifier identifies the attribute generated for the Parameters component of the body part. This conditional component shall be present if, and only if, a Parameters component is defined for this type of body part.

c) **This-child-entry** (C): Identifies the sequence-number of the child-entry that constitutes this body part. This shall be present for body part types for which child-entries are generated (e.g., Message body parts).

d) **Position** (M): Indicates the position of this body part within the sequence of values that constitutes the Extended Body Part attributes (Data and Parameters) for this Extended body part type.

e) **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 ITU-T Rec. X.690 | ISO/IEC 8825-1 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.

f) **Processed** (D *false*): An indication of whether the body part (as a single attribute or the Data component only) has been conveyed to the UA by means of the IPMS-MS's Fetch abstract-operation, or has been processed by an auto-action whose definition causes the body part to become *processed*. A Boolean.

NOTE 1 – As a consequence of its variability, the value of the Size component should be considered only an estimate of the body part's size.

NOTE 2 – This attribute may be used in preference to the IPM Synopsis attribute where the IPM comprises many recursive levels of forwarded IPM, or where the Parameters components are large, or where a summary of only a single body part is required.

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the Message-log entry for such an object) if, and only if, that object is a message whose content is an IPM. An IPMS-MS that supports this attribute shall also support the IPM Synopsis attribute.

19.6.1.4 IPM Auto-discarded

The **IPM Auto-discarded** attribute may be present in entries of the Delivery-log entry-class, and indicates whether the corresponding entry of the Delivery entry-class was deleted as a result of the performance of the IPM auto-discard auto-action.

```
ipm-auto-discarded ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     BOOLEAN,
       EQUALITY MATCHING-RULE    booleanMatch,
       NUMERATION                single-valued,
       ID                        id-sat-ipm-auto-discarded }
```

An IPMS-MS that supports this attribute shall maintain it for an information object if, and only if, that object is a Delivery-log entry corresponding to a delivered IPM which was deleted by the performance of the IPM auto-discard auto-action.

19.6.1.5 Body Part Signature Verification Status

The-**Body Part Signature Verification Status** attribute reports the outcome of verification by the IPMS-MS that body parts of a delivered IPM have not been modified in the MTS, if the IPMS-MS has the capability to perform such verification and a signature of the body part was present in the IPM. It is generated by the IPMS-MS. Trust in this indication implies trust in the IPMS-MS.

```
body-part-signature-verification-status  ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     BodyPartSignatureVerification,
       NUMERATION                single-valued,
       ID                        id-hat-body-part-signature-verification-status }

BodyPartSignatureVerification ::= SET OF SET {
       body-part-sequence-number BodyPartNumber,
       body-part-signature       SignatureStatus }
```

The signature status values are defined in 11.2.77 of ITU-T Rec. X.413 | ISO/IEC 10021-5.

19.6.2 Heading Attributes

Some attributes are derived from the Heading of an IPM. These attributes are defined and described below.

19.6.2.1 Heading

The **Heading** attribute is the (entire) Heading of an IPM.

```
heading ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     Heading,
       NUMERATION                single-valued,
       ID                        id-hat-heading }
```

An IPMS-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.

19.6.2.2 Heading Analyses

Some attributes have as their values OR-descriptors selected after analysis of the Heading. They identify the "primary", "copy", "blind copy", and "circulation list" recipients of an IPM of whom an RN, NRN, or reply is requested.

```
rn-requestors ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX      ORDescriptor,
       EQUALITY MATCHING-RULE     oRDescriptorMatch,
       NUMERATION                 multi-valued,
       ID                         id-hat-rn-requestors }

nrn-requestors ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX      ORDescriptor,
       EQUALITY MATCHING-RULE     oRDescriptorMatch,
       NUMERATION                 multi-valued,
       ID                         id-hat-nrn-requestors }

reply-requestors ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX      ORDescriptor,
       EQUALITY MATCHING-RULE     oRDescriptorMatch,
       NUMERATION                 multi-valued,
       ID                         id-hat-reply-requestors }
```

An IPMS-MS that supports one of these attributes shall maintain it for an information object that it holds (and the Message-log entry for such an object) if, and only if, that object is a delivered 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, Blind Copy, or Circulation List 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 that a reply is requested (in the case of the third). The value shall be the recipient specifier's Recipient component.

Note – These attributes exist for historical reasons, predating those defined to support IPM auto-correlation.

19.6.2.3 Heading Fields

Some attributes bear the names of heading fields and have those fields as their values.

```
this-ipm ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX      ThisIPMField,
       EQUALITY MATCHING-RULE     iPMIdentifierMatch,
       NUMERATION                 single-valued,
       ID                         id-hat-this-ipm }

originator ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX      OriginatorField,
       EQUALITY MATCHING-RULE     oRDescriptorMatch,
       OTHER MATCHING-RULES       {oRDescriptorElementsMatch |
                                   oRDescriptorSubstringElementsMatch |
                                   oRDescriptorSingleElementMatch, ...},
       NUMERATION                 single-valued,
       ID                         id-hat-originator }

replied-to-IPM ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX      RepliedToIPMField,
       EQUALITY MATCHING-RULE     iPMIdentifierMatch,
       NUMERATION                 single-valued,
       ID                         id-hat-replied-to-IPM }

subject ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX      SubjectField,
       EQUALITY MATCHING-RULE     mSStringMatch,
       SUBSTRINGS MATCHING-RULE   mSSubstringsMatch,
       NUMERATION                 single-valued,
       ID                         id-hat-subject }

expiry-time ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX      ExpiryTimeField,
       EQUALITY MATCHING-RULE     uTCTimeMatch,
       ORDERING MATCHING-RULE     uTCTimeOrderingMatch,
       NUMERATION                 single-valued,
       ID                         id-hat-expiry-time }

reply-time ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX      ReplyTimeField,
       EQUALITY MATCHING-RULE     uTCTimeMatch,
       ORDERING MATCHING-RULE     uTCTimeOrderingMatch,
       NUMERATION                 single-valued,
       ID                         id-hat-reply-time }
```

```
importance ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     ImportanceField,
       EQUALITY MATCHING-RULE    integerMatch,
       ORDERING MATCHING-RULE    integerOrderingMatch, -- not defined for 1988 Application Contexts
       NUMERATION                single-valued,
       ID                        id-hat-importance }

sensitivity ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     SensitivityField,
       EQUALITY MATCHING-RULE    integerMatch,
       ORDERING MATCHING-RULE    integerOrderingMatch, -- not defined for 1988 Application Contexts
       NUMERATION                single-valued,
       ID                        id-hat-sensitivity }

auto-forwarded ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     AutoForwardedField,
       EQUALITY MATCHING-RULE    booleanMatch,
       NUMERATION                single-valued,
       ID                        id-hat-auto-forwarded }
```

An IPMS-MS that supports one of these attributes shall maintain it for an information object that it holds (and the Message-log entry for such an object) if, and only if, that object is a message whose content is an IPM whose Heading contains the field whose name the attribute bears.

19.6.2.4 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,
       EQUALITY MATCHING-RULE    oRDescriptorMatch,
       OTHER MATCHING-RULES      {oRDescriptorElementsMatch |
                                  oRDescriptorSubstringElementsMatch |
                                  oRDescriptorSingleElementMatch, ...},
       NUMERATION                multi-valued,
       ID                        id-hat-authorizing-users }

primary-recipients ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     PrimaryRecipientsSubfield,
       EQUALITY MATCHING-RULE    recipientSpecifierMatch,
       OTHER MATCHING-RULES      {recipientSpecifierElementsMatch |
                                  recipientSpecifierSubstringElementsMatch |
                                  recipientSpecifierSingleElementMatch, ...},
       NUMERATION                multi-valued,
       ID                        id-hat-primary-recipients }

copy-recipients ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     CopyRecipientsSubfield,
       EQUALITY MATCHING-RULE    recipientSpecifierMatch,
       OTHER MATCHING-RULES      {recipientSpecifierElementsMatch |
                                  recipientSpecifierSubstringElementsMatch |
                                  recipientSpecifierSingleElementMatch, ...},
       NUMERATION                multi-valued,
       ID                        id-hat-copy-recipients }

blind-copy-recipients ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     BlindCopyRecipientsSubfield,
       EQUALITY MATCHING-RULE    recipientSpecifierMatch,
       OTHER MATCHING-RULES      {recipientSpecifierElementsMatch |
                                  recipientSpecifierSubstringElementsMatch |
                                  recipientSpecifierSingleElementMatch, ...},
       NUMERATION                multi-valued,
       ID                        id-hat-blind-copy-recipients }
```

NOTE – If the blind-copy-recipients envelope field is present then the heading field of the same name will be absent, and this attribute contains subfields of the envelope field as its values.

```
obsoleted-IPMs ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     ObsoletedIPMsSubfield,
       EQUALITY MATCHING-RULE    iPMIdentifierMatch,
       NUMERATION                multi-valued,
       ID                        id-hat-obsoleted-IPMs }

related-IPMs ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     RelatedIPMsSubfield,
       EQUALITY MATCHING-RULE    iPMIdentifierMatch,
       NUMERATION                multi-valued,
       ID                        id-hat-related-IPMs }
```

```
reply-recipients ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     ReplyRecipientsSubfield,
       EQUALITY MATCHING-RULE    oRDescriptorMatch,
       OTHER MATCHING-RULES      {oRDescriptorElementsMatch |
                                  oRDescriptorSubstringElementsMatch |
                                  oRDescriptorSingleElementMatch, ...},
       NUMERATION                multi-valued,
       ID                        id-hat-reply-recipients }
```

An IPMS-MS that supports one of these attributes shall maintain it for an information object that it holds (and the Message-log entry for such an object) 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.

19.6.2.5 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,
       NUMERATION                single-valued,   -- An equality match is specified for 1988
                                                  -- Application Contexts --
       ID                        id-hat-incomplete-copy }

languages ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     Language,
       EQUALITY MATCHING-RULE    mSStringMatch,
       SUBSTRINGS MATCHING-RULE  mSSubstringsMatch, -- Not defined for 1988 Application
Contexts
       NUMERATION                multi-valued,
       ID                        id-hat-languages }

auto-submitted ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     AutoSubmitted,
       EQUALITY MATCHING-RULE    integerMatch,
       NUMERATION                single-valued,
       ID                        id-hat-auto-submitted }

body-part-signatures  ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     BodyPartSignatures,
       NUMERATION                single-valued,
       ID                        id-hat-body-part-signatures }

ipm-security-label  ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     IPMSecurityLabel,
       NUMERATION                single-valued,
       ID                        id-hat-ipm-security-label }

body-part-security-label  ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     BodyPartSecurityLabel,
       NUMERATION                multi-valued,
       ID                        id-hat-body-part-security-label }

authorization-time ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX           AuthorizationTime,
       EQUALITY MATCHING-RULE    generalizedTimeMatch,
       ORDERING MATCHING-RULE    generalizedTimeOrderingMatch,
       NUMERATION                single-valued,
       ID                        id-hat-authorization-time }

circulation-list-recipients ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX           CirculationMember,
       EQUALITY MATCHING-RULE    circulationMemberMatch,
       OTHER MATCHING-RULES     { circulationMemberElementsMatch |
                                 circulationMemberSubstringElementsMatch |
                                 circulationMemberSingleElementMatch |
                                 circulationMemberCheckmarkMatch,... },
       NUMERATION                multi-valued,
       ID                        id-hat-circulation-list-recipients }

distribution-codes ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX           DistributionCode,
       EQUALITY MATCHING-RULE    distributionCodeMatch,
       NUMERATION                multi-valued,
       ID                        id-hat-distribution-codes }

extended-subject ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX           ExtendedSubject,
       EQUALITY MATCHING-RULE    mSStringMatch,
       SUBSTRINGS MATCHING-RULE    mSSubstringsMatch,
       NUMERATION                single-valued,
       ID                        id-hat-extended-subject }
```

```
information-category ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX           InformationCategory,
       EQUALITY MATCHING-RULE    informationCategoryMatch,
       NUMERATION                multi-valued,
       ID                        id-hat-information-category }

manual-handling-instructions ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX           ManualHandlingInstruction,
       EQUALITY MATCHING-RULE    mSStringMatch,
       NUMERATION                multi-valued,
       ID                        id-hat-manual-handling-instructions }

originators-reference ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX           OriginatorsReference,
       EQUALITY MATCHING-RULE    mSStringMatch,
       NUMERATION                single-valued,
       ID                        id-hat-originators-reference }

precedence-policy-identifier ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX           PrecedencePolicyIdentifier,
       EQUALITY MATCHING-RULE    objectIdentifierMatch,
       NUMERATION                single-valued,
       ID                        id-hat-precedence-policy-id }
```

An IPMS-MS that supports one of these attributes shall maintain it for an information object that it holds (and the Message-log entry for such an object) 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 IPMS-MS shall maintain one attribute value for each language the extension identifies.

NOTE – Each value of Languages comprises a language code, followed optionally by a country code. The Languages attribute's substrings matching rule may be used to select values of a given language code, regardless of the presence, or value of the optional country code; see A.1.2.

19.6.2.6 Recipient Extensions

One attribute bears the name of a recipient extension contained in the recipient-specifier which identifies this IPMS-MS-user and has as its value the values of that recipient extension.

```
precedence ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX           Precedence,
       EQUALITY MATCHING-RULE    integerMatch,
       NUMERATION                single-valued,
       ID                        id-hat-precedence }
```

An IPMS-MS that supports this attributes shall maintain it for an information object that it holds (and the Message-log entry for such an object) if, and only if, that object is a delivered-message whose content is an IPM whose Heading contains a recipient-specifier which identifies this IPMS-MS-user, and that recipient-specifier contains the recipient extension whose name the attribute bears.

19.6.2.7 Envelope Extensions

Some attributes bear the names of extensions that are logically part of the IPM, but to facilitate efficient processing are envelope extensions, and have as their values the values of those extensions or a part thereof.

```
body-part-encryption-token  ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     BodyPartTokens,
       NUMERATION                single-valued,
       ID                        id-hat-body-part-encryption-token }

forwarded-content-token ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     ForwardedContentToken,
       NUMERATION                single-valued,
       ID                        id-hat-forwarded-content-token }

forwarding-token ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     MessageToken,
       NUMERATION                single-valued,
       ID                        id-hat-forwarding-token }
```

An IPMS-MS that supports the Forwarded Content Token attribute shall maintain it for an information object that it holds (and the Message-log entry for such an object) if that object is a message whose content is an IPM whose Body contains a Forwarded Content. An IPMS-MS that supports the Forwarding Token attribute shall maintain it for an information object that it holds if, and only if, that object is a child-entry which represents a Forwarded Content body part, where that content originally had an associated message-token containing encrypted-data.

19.6.3 Body Attributes

Some attributes are derived from the Body of an IPM. These attributes are defined and described below.

NOTE – The attributes defined in 19.6.3.4, 19.6.3.5, and 19.6.3.6 exist for historical reasons. When using a 1994 Application Context only the Extended body parts attributes should be used.

19.6.3.1 Body

The **Body** attribute is the (entire) Body of an IPM.

```
body ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX     Body,
        NUMERATION                single-valued,
        ID                        id-bat-body }
```

An IPMS-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.

19.6.3.2 Extended Body Part Types

The **Extended Body Part Types** attribute identifies the Extended body part types represented in an IPM. For the purposes of this attribute, all body parts of an IPM are considered to be of type Extended regardless of whether they were so conveyed to the IPMS-MS.

```
extended-body-part-types ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX     OBJECT IDENTIFIER,
        EQUALITY MATCHING-RULE    objectIdentifierMatch,
        NUMERATION                multi-valued,
        ID                        id-bat-extended-body-part-types }
```

An IPMS-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 body parts. It shall maintain one attribute value for every type of body part present. The value shall denote the type as specified in 7.3.2.

NOTE – Each value of this attribute identifies both an Extended body part type represented in the IPM and the attribute generated for the Data component(s) of the body part(s) of that body part type, as specified in 19.6.3.3.

19.6.3.3 Extended Body Parts

Some attributes, unnamed, have as their values the value components of the Sequence type associated with the instance-of type that constitutes the Data (and Parameters) components of Extended body parts. See 7.3.2 above, and C.7 of ITU-T Rec. X.681 | ISO/IEC 8824-2. For the purposes of these attributes, all body parts of an IPM are considered to be of type Extended regardless of whether they were so conveyed to the IPMS-MS; see 19.3, item (b).

To each Extended body part type there corresponds two attributes. The first attribute is denoted by the Object Identifier that is the &id field of the TYPE-IDENTIFIER object that constitutes the Data component of a body part of that type. The content of this first attribute is the value component of the Sequence type associated with the instance-of type for that Data component. The second attribute is denoted by the Object Identifier that is the &id field of the TYPE-IDENTIFIER object that constitutes the Parameters component of a body part of that type. The content of this second attribute is the value component of the Sequence type associated with the instance-of type for that Parameters component.

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 IPMS-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. The order of values shall reflect the order of appearance of body parts of that type in the IPM.

The Data and Parameters attributes of Extended body parts may be derived from the following parameterized object assignments:

```
extended-body-part-data-attribute{EXTENDED-BODY-PART-TYPE:ebpt} ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX     [0] EXPLICIT ebpt.&data.&Type,
        NUMERATION                multi-valued,
        ID                        ebpt.&data.&id }
```

```
extended-body-part-parameters-attribute{EXTENDED-BODY-PART-TYPE:ebpt} ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX       [0] EXPLICIT ebpt.&parameters.&Type,
        NUMERATION                  multi-valued,
        ID                          ebpt.&parameters.&id }
```

NOTE 1 – In editions of this Specification published before 1994, the Parameters and Data attributes of extended body parts were derived from Externals. When the single-ASN1-type encoding alternative of the External is used, the value of the encoding of an External is identical with that of Type-Identifier. However, the IPMS-MS may generate Parameters and Data attributes whose ATTRIBUTE-SYNTAX also supports the octet-aligned encoding alternative. The following associated types indicate the ATTRIBUTE-SYNTAX of the Parameters and Data attributes respectively, that may be generated by the IPMS-MS, and should be anticipated by the IPMS-MS-user:

```
CHOICE {
      single-ASN1-type    [0]   EXPLICIT ebpt.&parameters.&Type,
      octet-aligned       [1]   IMPLICIT OCTET STRING }

CHOICE {
      single-ASN1-type    [0]   EXPLICIT ebpt.&data.&Type,
      octet-aligned       [1]   IMPLICIT OCTET STRING }
```

The constraint specified in 7.3.2 on the choice of encoding applies.

EXAMPLES

For example, the Data attribute for the IA5 Text Extended body part type may be derived as follows:

```
ia5-text-extended-data-attribute ATTRIBUTE ::=

extended-body-part-data-attribute {ia5-text-body-part}
```

This is equivalent to the following assignment:

```
ia5-text-extended-data-attribute ATTRIBUTE ::=
        WITH ATTRIBUTE-SYNTAX       [0] EXPLICIT IA5TextData,
        NUMERATION                  multi-valued,
        ID                          id-et-ia5-text }
```

The Data attribute for the 'Simple Spreadsheet' Extended body part type used as an example in 7.3.1 may be derived as follows:

```
simple-spreadsheet-extended-data-attribute ATTRIBUTE ::=

extended-body-part-data-attribute {simple-spreadsheet-body-part}
```

This is equivalent to the following assignment:

```
simple-spreadsheet-extended-data-attribute ATTRIBUTE ::=
        WITH ATTRIBUTE-SYNTAX       [0] EXPLICIT OCTET STRING,
        NUMERATION                  multi-valued,
        ID                          {local-object-identifier 1} }
```

NOTE 1 – The Extended body parts attributes cannot be enumerated because the Extended body part types cannot be so enumerated.

NOTE 2 – The Extended Body Part Types attribute (see 19.6.3.2) indicates the types of the Extended body part attributes present in the entry representing an IPM.

19.6.3.4 Basic Body Parts

Some attributes bear the names of basic body part types and have, with one exception, such body parts as their values.

NOTE – The attributes defined below exist for historical reasons. When using a 1994 Application Context, only the (Extended) body parts attributes of 19.6.3.3 should be used.

An IPMS-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 is a message whose content is an IPM. The Message Body Parts attribute below has as its values the sequence-numbers the IPMS-MS assigns to those messages.

```
ia5-text-body-parts ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX       IA5TextBodyPart,
        NUMERATION                  multi-valued,
        ID                          id-bat-ia5-text-body-parts }

g3-facsimile-body-parts ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX       G3FacsimileBodyPart,
        NUMERATION                  multi-valued,
        ID                          id-bat-g3-facsimile-body-parts }

g4-class1-body-parts ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX       G4Class1BodyPart,
        NUMERATION                  multi-valued,
        ID                          id-bat-g4-class1-body-parts }
```

```
teletex-body-parts ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX    TeletexBodyPart,
        NUMERATION               multi-valued,
        ID                       id-bat-teletex-body-parts }

videotex-body-parts ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX    VideotexBodyPart,
        NUMERATION               multi-valued,
        ID                       id-bat-videotex-body-parts }

encrypted-body-parts ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX    EncryptedBodyPart,
        NUMERATION               multi-valued,
        ID                       id-bat-encrypted-body-parts }

message-body-parts ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX    SequenceNumber,
        NUMERATION               multi-valued,
        ID                       id-bat-message-body-parts }

mixed-mode-body-parts ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX    MixedModeBodyPart,
        NUMERATION               multi-valued,
        ID                       id-bat-mixed-mode-body-parts }

bilaterally-defined-body-parts ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX    BilaterallyDefinedBodyPart,
        NUMERATION               multi-valued,
        ID                       id-bat-bilaterally-defined-body-parts }

nationally-defined-body-parts ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX    NationallyDefinedBodyPart,
        NUMERATION               multi-valued,
        ID                       id-bat-nationally-defined-body-parts }
```

An IPMS-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. The order of values shall reflect the order of appearance of body parts of that type in the IPM.

19.6.3.5 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.

NOTE – The attributes defined below exist for historical reasons. When using a 1994 Application Context, only the (Extended) body parts attributes of 19.6.3.3 should be used.

```
ia5-text-parameters ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX    IA5TextParameters,
        NUMERATION               multi-valued,
        ID                       id-bat-ia5-text-parameters }

g3-facsimile-parameters ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX    G3FacsimileParameters,
        NUMERATION               multi-valued,
        ID                       id-bat-g3-facsimile-parameters }

teletex-parameters ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX    TeletexParameters,
        NUMERATION               multi-valued,
        ID                       id-bat-teletex-parameters }

videotex-parameters ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX    VideotexParameters,
        NUMERATION               multi-valued,
        ID                       id-bat-videotex-parameters }

encrypted-parameters ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX    EncryptedParameters,
        NUMERATION               multi-valued,
        ID                       id-bat-encrypted-parameters }

message-parameters ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX    MessageParameters,
        NUMERATION               multi-valued,
        ID                       id-bat-message-parameters }
```

An IPMS-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. The order of values shall reflect the order of appearance of body parts of that type in the IPM.

19.6.3.6 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.

NOTE – The attributes defined below exist for historical reasons. When using a 1994 Application Context, only the (Extended) body parts attributes of 19.6.3.3 should be used.

```
ia5-text-data ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     IA5TextData,
       NUMERATION                multi-valued,
       ID                        id-bat-ia5-text-data }

g3-facsimile-data ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     G3FacsimileData,
       NUMERATION                multi-valued,
       ID                        id-bat-g3-facsimile-data }

teletex-data ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     TeletexData,
       NUMERATION                multi-valued,
       ID                        id-bat-teletex-data }

videotex-data ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     VideotexData,
       NUMERATION                multi-valued,
       ID                        id-bat-videotex-data }

encrypted-data ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     EncryptedData,
       NUMERATION                multi-valued,
       ID                        id-bat-encrypted-data }

message-data ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     MessageData,
       NUMERATION                multi-valued,
       ID                        id-bat-message-data }
```

An IPMS-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. The order of values shall reflect the order of appearance of body parts of that type in the IPM.

19.6.4 Notification Attributes

Some attributes are derived from an IPN. These attributes are defined and described below.

19.6.4.1 Common Fields

Some attributes bear the names of common fields and have those fields as their values.

```
subject-ipm ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     SubjectIPMField,
       EQUALITY MATCHING-RULE    iPMIdentifierMatch,
       NUMERATION                single-valued,
       ID                        id-nat-subject-ipm }

ipn-originator ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     IPNOriginatorField,
       EQUALITY MATCHING-RULE    oRDescriptorMatch,
       OTHER MATCHING-RULES      {oRDescriptorElementsMatch |
                                  oRDescriptorSubstringElementsMatch |
                                  oRDescriptorSingleElementMatch, ...},
       NUMERATION                single-valued,
       ID                        id-nat-ipn-originator }

ipm-intended-recipient ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     IPMIntendedRecipientField,
       EQUALITY MATCHING-RULE    oRDescriptorMatch,
       OTHER MATCHING-RULES      {oRDescriptorElementsMatch |
                                  oRDescriptorSubstringElementsMatch |
                                  oRDescriptorSingleElementMatch, ...},
       NUMERATION                single-valued,
       ID                        id-nat-ipm-intended-recipient }
```

```
conversion-eits ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     MS-EIT,
       EQUALITY MATCHING-RULE    objectIdentifierMatch,
       NUMERATION                multi-valued,
       ID                        id-nat-conversion-eits }

notification-extensions ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     IPMSExtension {{ NotificationExtensions }},
       NUMERATION                multi-valued,
       ID                        id-nat-notification-extensions }
```

An IPMS-MS that supports one of these attributes shall maintain it for an information object that it holds (and the Message-log entry for such an object) if, and only if, that object is a message whose content is an IPN that contains the field whose name the attribute bears.

19.6.4.2 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,
       EQUALITY MATCHING-RULE    integerMatch,
       NUMERATION                single-valued,
       ID                        id-nat-non-receipt-reason }

discard-reason ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     DiscardReasonField,
       EQUALITY MATCHING-RULE    integerMatch,
       NUMERATION                single-valued,
       ID                        id-nat-discard-reason }

auto-forward-comment ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     AutoForwardCommentField,
       EQUALITY MATCHING-RULE    mSStringMatch,
       SUBSTRINGS MATCHING-RULE  mSSubstringsMatch,
       NUMERATION                single-valued,
       ID                        id-nat-auto-forward-comment }

returned-ipm ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     ReturnedIPMField,
       NUMERATION                single-valued,
       ID                        id-nat-returned-ipm }

nrn-extensions ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     IPMSExtension {{ NRNExtensions }},
       NUMERATION                multi-valued,
       ID                        id-nat-nrn-extensions }
```

An IPMS-MS that supports one of these attributes shall maintain it for an information object that it holds (and the Message-log entry for such an object) if, and only if, that object is a message whose content is an NRN that contains the field whose name the attribute bears. The Returned IPM attribute shall not be present in entries of either the submission-log or delivery-log.

19.6.4.3 Receipt Fields

Some attributes bear the names of receipt fields and have those fields as their values.

```
receipt-time ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     ReceiptTimeField,
       EQUALITY MATCHING-RULE    uTCTimeMatch,
       ORDERING MATCHING-RULE    uTCTimeOrderingMatch,
       NUMERATION                single-valued,
       ID                        id-nat-receipt-time }

acknowledgment-mode ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     AcknowledgmentModeField,
       EQUALITY MATCHING-RULE    integerMatch,
       NUMERATION                single-valued,
       ID                        id-nat-acknowledgment-mode }

suppl-receipt-info ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     SupplReceiptInfoField,
       EQUALITY MATCHING-RULE    mSStringMatch,
       SUBSTRINGS MATCHING-RULE  mSSubstringsMatch,
       NUMERATION                single-valued,
       ID                        id-nat-suppl-receipt-info }

rn-extensions ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     IPMSExtension {{ RNExtensions }},
       NUMERATION                multi-valued,
       ID                        id-nat-rn-extensions }
```

An IPMS-MS that supports one of these attributes shall maintain it for an information object that it holds (and the Message-log entry for such an object) if, and only if, that object is a message whose content is an RN that contains the field whose name the attribute bears.

19.6.4.4 Other Notification Type Fields

The **Other Notification Type Fields** attribute contains values of the field whose name it bears.

```
other-notification-type-fields ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     IPMSExtension,
       NUMERATION                multi-valued,
       ID                        id-nat-other-notification-type-fields }
```

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the Message-log entry for such an object) if, and only if, that object is a message whose content is an IPN whose content is an ON.

19.6.5 Correlation Attributes

Some attributes correlate messages which are interrelated in various ways:

a) an IPM and the IPMs generated in reply;

b) an IPM and the IPNs notifying receipt or non-receipt;

c) an IPM and the IPMs which subsequently forward it, or obsolete it, or are related to it.

Some attributes are relevant to all submitted and delivered messages which contain replies and notifications; others are specific to submitted-message entries, and correlate the delivered replies and IPNs received in response to a submitted message; others still are specific to delivered-message entries, and correlate the replies and IPNs generated by this IPMS-MS-user in response to a delivered message. All the Correlation attributes defined in this clause are generated by the IPMS-MS. With the exception of AC Forwarded IPMs, all other Correlation attributes defined in this clause are applicable only to main entries.

The deletion of an entry referred to by one of the correlation attributes has no effect on the value of that attribute.

19.6.5.1 Common attributes

Some attributes, common to the entries of both the Submission and Delivery entry-classes (and corresponding entries of the Message-log entry-class), correlate an IPM with the IPMs which indicate in their Heading fields that they render it obsolete, or are related to it, or are replies to it, or have forwarded it. A further common attribute correlates an IPN with the IPM entry to which it refers.

19.6.5.1.1 AC Forwarding IPMs

The **AC Forwarding IPMs** attribute, which is multi-valued, contains the sequence-numbers of the IPM entries that bear the present IPM as a Message or Forwarded Content body-part. One value of the attribute shall be generated for each forwarding IPM containing the message. The attribute values are stored in ascending order of the forwarding entries' creation times.

```
ac-forwarding-ipms ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     SequenceNumber,
       EQUALITY MATCHING-RULE    integerMatch,
       ORDERING MATCHING-RULE    integerOrderingMatch,
       NUMERATION                multi-valued,
       ID                        id-cat-forwarding-ipms }
```

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the Message-log entry for such an object) if, and only if, that object is a message whose content is an IPM, which has been the subject of forwarding by, or on behalf of, this IPMS-MS-user.

19.6.5.1.2 AC Forwarded IPMs

The **AC Forwarded IPMs** attribute, which is multi-valued, contains the sequence-numbers of the stored IPMs that correspond to the Message or Forwarded Content body-part that the present child-entry represents. One value of the attribute shall be generated for each stored IPM that corresponds to the body part that the present entry represents. Each value indicates the sequence-number of the corresponding IPM. The attribute is absent unless at least one IPM entry exists that corresponds to the IPM body part that the present entry represents.

```
ac-forwarded-ipms ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     SequenceNumber,
       EQUALITY MATCHING-RULE    integerMatch,
       ORDERING MATCHING-RULE    integerOrderingMatch,
       NUMERATION                multi-valued,
       ID                        id-cat-forwarded-ipms }
```

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the Message-log entry for such an object) if, and only if, that object is a child-entry that represents a Message body part or Forwarded Content IPM body part that corresponds to at least one IPM present in the IPMS-MS.

19.6.5.1.3 AC Obsoleting IPMs

The **AC Obsoleting IPMs** attribute, which is multi-valued, contains the sequence-numbers of the IPM entries that indicate, by means of their Obsoleted IPMs heading field, that they render obsolete the present IPM. The attribute values are stored in ascending order of the obsoleting entries' creation times.

```
ac-obsoleting-ipms ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX      SequenceNumber,
       EQUALITY MATCHING-RULE     integerMatch,
       ORDERING MATCHING-RULE     integerOrderingMatch,
       NUMERATION                 multi-valued,
       ID                         id-cat-obsoleting-ipms }
```

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the Message-log entry for such an object) if, and only if, that object is a message whose content is an IPM, to which at least one IPM refers in its Obsoleted IPMs heading field.

19.6.5.1.4 AC Obsoleted IPMs

The **AC Obsoleted IPMs** attribute, which is multi-valued, contains the sequence-numbers of the IPM entries identified by the Obsoleted IPMs heading field of the present IPM. One value of the attribute shall be generated for each subfield of the Obsoleted IPMs heading field. The value *stored* indicates the one or more IPM entries identified by a given subfield. The value *absent* indicates that no IPM entry corresponds to a given subfield. The attribute values are stored in the same order as the corresponding values of the heading field. The attribute is absent unless at least one IPM entry exists that is identified by a subfield of the Obsoleted IPMs heading field.

```
ac-obsoleted-ipms ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX      IPMLocation,
       OTHER MATCHING-RULES       {iPMLocationMatch, ...},
       NUMERATION                 multi-valued,
       ID                         id-cat-obsoleted-ipms }

IPMLocation ::= CHOICE {
       stored     SET OF SequenceNumber,
       absent     NULL,
       ... }
```

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the Message-log entry for such an object) if, and only if, that object is a message whose content is an IPM, which identifies in its Obsoleted IPMs heading field one or more IPMs present in the IPMS-MS.

19.6.5.1.5 AC Relating IPMs

The **AC Relating IPMs** attribute, which is multi-valued, contains the sequence-numbers of the IPM entries which indicate, by means of their Related IPMs heading field, that they are related to the present IPM. The attribute values are stored in ascending order of the relating entries' creation times.

```
ac-relating-ipms ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX      SequenceNumber,
       EQUALITY MATCHING-RULE     integerMatch,
       ORDERING MATCHING-RULE     integerOrderingMatch,
       NUMERATION                 multi-valued,
       ID                         id-cat-relating-ipms }
```

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the Message-log entry for such an object) if, and only if, that object is a message whose content is an IPM, to which at least one IPM refers in its Related IPMs heading field.

19.6.5.1.6 AC Related IPMs

The **AC Related IPMs** attribute, which is multi-valued, contains the sequence-numbers of the IPM entries identified by the Related IPMs heading field of the present IPM. One value of the attribute shall be generated for each subfield of the Related IPMs heading field. The value *stored* indicates the one or more IPM entries identified by a given subfield. The value *absent* indicates that no IPM entry corresponds to a given subfield. The attribute values are stored in the same order as the corresponding values of the heading field. The attribute is absent unless at least one IPM entry exists that is identified by a subfield of the Related IPMs heading field.

```
        ac-related-ipms ATTRIBUTE ::= {
                WITH ATTRIBUTE-SYNTAX       IPMLocation,
                OTHER MATCHING-RULES        {iPMLocationMatch, ...},
                NUMERATION                  multi-valued,
                ID                          id-cat-related-ipms }
```

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the Message-log entry for such an object) if, and only if, that object is a message whose content is an IPM, which identifies in its Related IPMs heading field one or more IPMs present in the IPMS-MS.

19.6.5.1.7 AC Replied-to IPM

The **AC Replied-to IPM** attribute, which is multi-valued, contains the sequence-numbers of each instance of the IPM entry to which the present IPM is a reply.

```
        ac-replied-to-ipm ATTRIBUTE ::= {
                WITH ATTRIBUTE-SYNTAX       SequenceNumber,
                EQUALITY MATCHING-RULE      integerMatch,
                ORDERING MATCHING-RULE      integerOrderingMatch,
                NUMERATION                  multi-valued,
                ID                          id-cat-replied-to-ipm }
```

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the Message-log entry for such an object) if, and only if, that object is a message whose content is an IPM, and which contains a Replied-to IPM heading field. Its value shall be the sequence-number of each entry identified by the Replied-to IPM heading field.

19.6.5.1.8 AC Replying IPMs

The **AC Replying IPMs** attribute, which is multi-valued, contains the sequence-numbers of the IPM entries which indicate, by means of their Replied-to IPM heading field, that they were sent in reply to the present IPM. The attribute values are stored in ascending order of the replying entries' creation times.

```
        ac-replying-ipms ATTRIBUTE ::= {
                WITH ATTRIBUTE-SYNTAX       SequenceNumber,
                EQUALITY MATCHING-RULE      integerMatch,
                ORDERING MATCHING-RULE      integerOrderingMatch,
                NUMERATION                  multi-valued,
                ID                          id-cat-replying-ipms }
```

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the Message-log entry for such an object) if, and only if, that object is a message whose content is an IPM, and at least one reply to that IPM has been submitted or delivered.

19.6.5.1.9 AC Subject IPM

The **AC Subject IPM** attribute, which is multi-valued, contains the sequence-numbers of each instance of the IPM entry to which the present IPN refers.

```
        ac-subject-ipm ATTRIBUTE ::= {
                WITH ATTRIBUTE-SYNTAX       SequenceNumber,
                EQUALITY MATCHING-RULE      integerMatch,
                ORDERING MATCHING-RULE      integerOrderingMatch,
                NUMERATION                  multi-valued,
                ID                          id-cat-subject-ipm }
```

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the Message-log entry for such an object) if, and only if, that object is a message whose content is an IPN, and whose Subject IPM common field identifies an IPM entry. Its value shall be the sequence-number of each entry identified by the Subject IPM common field.

19.6.5.2 Submitted message correlation

Some attributes correlate IPMs submitted by this IPMS-MS-user with the replies and notifications subsequently delivered in response, and provide a summary of the responses requested and those received.

> NOTE – If a non-delivery-report is received concerning an intended recipient of a submitted message then no reply or IPN will be generated by that recipient. The UA can determine when this applies by examining the MS AC-report-summary attribute and correlating the recipients identified in the MS Recipient-names and AC IPM Recipients attributes.

For any given entry, the number of values of each of the submitted message correlation attributes, defined below, shall be the same. The ordering of values in the attributes is aligned, such that all values at a given order position within the sequence of attribute-values refer to one of the intended recipients of the IPM. The AC IPM Recipients attribute shall be supported if any of the other attributes defined in 19.6.5.2 are supported.

19.6.5.2.1 AC IPM Recipients

The **AC IPM Recipients** attribute, which is multi-valued, contains one value for each distinct Primary, Copy, Blind Copy, and Circulation List Recipient of the submitted IPM. Where two or more recipient specifiers contain the same value of OR-name (i.e. the Formal-name component of the Recipient component of recipient specifier) a value is generated only for the first of these. Fields are inspected in the order Primary, Copy, Blind Copy, and Circulation List Recipients.

The order of values in this attribute is aligned with the order of values in the other attributes defined in 19.6.5.2.

> NOTE – For example, the value containing the OR-descriptor for a given recipient occupies the same position in this attribute as the position occupied by the value containing the delivered-reply-status for the same recipient in the AC Delivered Replies Summary attribute.

The initial values of this attribute are drawn from the corresponding recipient specifiers, as determined by the procedure indicated above.

```
ac-ipm-recipients ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     ORDescriptor,
       EQUALITY MATCHING-RULE    oRDescriptorMatch,
       OTHER MATCHING-RULES      {oRDescriptorElementsMatch |
                                  oRDescriptorSubstringElementsMatch |
                                  oRDescriptorSingleElementMatch, ...},
       NUMERATION                multi-valued,
       ID                        id-cat-ipm-recipients }
```

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the submission-log entry for such an object) if, and only if, that object is a submitted message whose content is an IPM. It shall maintain one attribute value for each distinct recipient specifier in the IPM's Primary, Copy, Blind Copy, and Circulation List Recipients fields. The values shall be the recipient specifiers' Recipient components.

19.6.5.2.2 AC Delivered Replies Summary

The **AC Delivered Replies Summary** attribute, which is multi-valued, records whether replies have been solicited from each of the AC IPM Recipients of a submitted IPM and whether replies have been received. The initial value of the attribute is set for each of the AC IPM Recipients according to whether a reply was requested of that recipient. The order of values in this attribute is aligned with the order of values in the other attributes defined in 19.6.5.2.

```
ac-delivered-replies-summary ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     DeliveredReplyStatus,
       EQUALITY MATCHING-RULE    integerMatch,
       ORDERING MATCHING-RULE    integerOrderingMatch,
       NUMERATION                multi-valued,
       ID                        id-cat-delivered-replies-summary }

DeliveredReplyStatus ::= INTEGER {
       no-reply-requested        (0)   -- reply not requested --,
       reply-outstanding         (1)   -- reply requested --  ,
       reply-received            (2) }
```

For each value present in the AC IPM Recipients attribute, this attribute may assume any one of the following values:

a) *no-reply-requested*: No reply was requested of this recipient and none has been received. This is a possible initial value of the attribute.

b) *reply-outstanding*: A reply was requested of this recipient and none has been received. This is a possible initial value of the attribute.

c) *reply-received*: One or more replies originated by this recipient have been received.

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the submission-log entry for such an object) if, and only if, that object is a submitted message whose content is an IPM. It shall maintain one attribute value for each value of the AC IPM Recipients attribute. The initial values shall reflect the values of the recipient specifiers' Reply-requested components.

19.6.5.2.3 AC Correlated Delivered Replies

The **AC Correlated Delivered Replies** attribute, which is multi-valued, identifies the delivered IPMs which were originated by an intended recipient of a submitted IPM, and which indicate, by means of their Replied-to IPM heading field, that they were sent in reply to the submitted IPM. The initial value of the attribute is set for each of the AC IPM Recipients and indicates that no reply has been received. The order of values in this attribute is aligned with the order of values in the other attributes defined in 19.6.5.2.

```
ac-correlated-delivered-replies ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     CorrelatedDeliveredReplies,
       NUMERATION                multi-valued,
       ID                        id-cat-correlated-delivered-replies }

CorrelatedDeliveredReplies ::= CHOICE {
       no-reply-received    [0]  NULL,
       received-replies     [1]  SEQUENCE OF SequenceNumber }
```

The components of correlated-delivered-replies have the following meaning:

a) **No-reply-received** (C): No reply has been received from this intended recipient of the submitted IPM. This is the initial value of the attribute.

b) **Received-replies** (C): This identifies the sequence-numbers of the one or more IPMs received by this IPMS-MS-user in reply to a previously submitted IPM. The sequence-numbers are present in ascending order of the corresponding entries' creation times.

Each value of this attribute indicates that no reply has been received, or identifies each delivered-message entry which contains a reply, whose subject is the originally submitted IPM, and whose originator was an intended recipient of the IPM.

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the submission-log entry for such an object) if, and only if, that object is a submitted message whose content is an IPM. It shall maintain one attribute value for each value of the AC IPM Recipients attribute. The initial values shall be *no-reply-received*.

19.6.5.2.4 AC Delivered IPN Summary

The **AC Delivered IPN Summary** attribute, which is multi-valued, contains a summary of the IPNs requested of, and generated by, or on behalf of, the AC IPM Recipients of a submitted message. The initial values of AC Delivered IPN Summary are set according to the values of the Notification-requests component of each of the IPM Recipient's recipient specifier, and each value is updated as each IPN is received (with the condition that the new value is greater than the value it replaces). The order of values in this attribute is aligned with the order of values in the other attributes defined in 19.6.5.2.

```
ac-delivered-ipn-summary  ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     DeliveredIPNStatus,
       EQUALITY MATCHING-RULE    integerMatch,
       ORDERING MATCHING-RULE    integerOrderingMatch,
       NUMERATION                multi-valued,
       ID                        id-cat-delivered-ipn-summary }

DeliveredIPNStatus ::= INTEGER {
       no-ipn-requested           (0),
       an-requested               (3),
       nrn-requested              (5),
       rn-requested               (10),
       an-received                (13),
       ipm-auto-forwarded         (15),
       ipm-discarded              (20),
       rn-received                (25) }
```

For each value present in the AC IPM Recipients attribute, this attribute may assume any one of the following values:

a) *no-ipn-requested*: No notification-requests were made of this recipient. This is a possible initial value of the attribute.

b) *an-requested*: An advice notification was requested of this recipient, and no such notification has been received. This is a possible initial value of the attribute.

c) *nrn-requested*: Non-receipt-notification was requested of this recipient and no such notification has been received. This is a possible initial value of the attribute.

d) *rn-requested*: Receipt notification was requested of this recipient and no such notification has been received. This is a possible initial value of the attribute.

e) *an-received*: An advice notification generated by, or on behalf of, this recipient has been received which contains additional advice concerning receipt of the IPM; see 8.4.

f) *ipm-auto-forwarded*: A non-receipt-notification generated by, or on behalf of, this recipient has been received which indicates that the IPM was auto-forwarded; see 8.2.1.

g) *ipm-discarded*: A non-receipt-notification generated by, or on behalf of, this recipient has been received which indicates that the IPM was discarded; see 8.2.1.

h) *rn-received*: An IPN generated by, or on behalf of, this recipient has been received which confirms the receipt of the submitted IPM.

NOTE – A future version of this Specification may specify additional values of AC Delivered IPN Summary for ONs.

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the submission-log entry for such an object) if, and only if, that object is a submitted message whose content is an IPM. It shall maintain one attribute value for each value of the AC IPM Recipients attribute. The initial values shall reflect the values of the recipient specifiers' Notification-requests components.

19.6.5.2.5 AC Correlated Delivered IPNs

The **AC Correlated Delivered IPNs** attribute, which is multi-valued, identifies the delivered IPNs that have been correlated with each of the AC IPM Recipients of a submitted IPM. The initial value of the attribute is set for each of the AC IPM Recipients and indicates that no IPNs have been received. The order of values in this attribute is aligned with the order of values in the other attributes defined in 19.6.5.2.

```
ac-correlated-delivered-ipns ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     CorrelatedDeliveredIPNs,
       NUMERATION                multi-valued,
       ID                        id-cat-correlated-delivered-ipns }

CorrelatedDeliveredIPNs ::= CHOICE {
       no-ipn-received    [0]  NULL,
       ipns-received      [1]  SEQUENCE OF SequenceNumber }
```

The components of **correlated-delivered-IPNs** have the following meaning:

a) **No-IPN-received** (C): No IPN has been received from this intended recipient or from an actual recipient acting on his behalf. This is the initial value of the attribute.

b) **IPNs-received** (C): This identifies the sequence-numbers of the IPN entries received from this intended recipient or the actual recipient acting on his behalf. The sequence-numbers are present in ascending order of the corresponding entries' creation times.

Each value of this attribute indicates that no IPN has been received, or identifies each delivered-message entry which contains an IPN, whose subject is the originally submitted IPM, and whose originator received the IPM as, or on behalf of, an intended recipient.

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the submission-log entry for such an object) if, and only if, that object is a submitted message whose content is an IPM. It shall maintain one attribute value for each value of the AC IPM Recipients attribute. The initial values shall be *no-IPN-received*.

19.6.5.3 Delivered message correlation

Some attributes correlate IPMs delivered to this IPMS-MS-user with the replying IPMs and IPNs subsequently submitted by the IPMS-MS-user, and provide a summary of the responses requested and those generated.

19.6.5.3.1 AC Submitted Reply Status

The **AC Submitted Reply Status attribute** indicates whether a reply to a delivered IPM was requested and whether one has been sent.

```
ac-submitted-reply-status ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     SubmittedReplyStatus,
       EQUALITY MATCHING-RULE    integerMatch,
       ORDERING MATCHING-RULE    integerOrderingMatch,
       NUMERATION                single-valued,
       ID                        id-cat-submitted-reply-status }

SubmittedReplyStatus ::= INTEGER {
       no-reply-requested        (0),
       no-reply-intended         (1),
       reply-pending             (2),
       reply-sent                (3) }
```

This attribute may assume any one of the following values:

a) *no-reply-requested*: The delivered IPM does not contain a request for this recipient to generate a reply. This is a possible initial value of the attribute. If the subject recipient specifier of the delivered IPM cannot be identified, this value is assumed.

b) *no-reply-intended*: This recipient has determined not to generate a reply to the delivered IPM (regardless of whether one was requested).

b) *reply-pending*: The delivered message contains a request for this recipient to generate a reply, or this recipient has declared an intention to generate a reply. The action is still outstanding. This is a possible initial value of the attribute.

c) *reply-sent*: A reply to the delivered IPM has been generated.

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the delivery-log entry for such an object) if, and only if, that object is a delivered message whose content is an IPM. The initial value of the attribute is set in accordance with the reply-requested component of the subject recipient specifier. The attribute is subject to modification by means of the Modify abstract-operation and the Auto-modify auto-action.

19.6.5.3.2 AC Submitted IPN Status

The **AC Submitted IPN Status** attribute indicates whether an interpersonal notification for a delivered message was requested and whether one has been sent.

```
ac-submitted-ipn-status ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     SubmittedIPNStatus,
       EQUALITY MATCHING-RULE    integerMatch,
       ORDERING MATCHING-RULE    integerOrderingMatch,
       NUMERATION                single-valued,
       ID                        id-cat-submitted-ipn-status }

SubmittedIPNStatus ::= INTEGER{
       no-ipn-requested               (0),
       nrn-requested                  (5),
       nrn-with-ipm-return-requested  (10),
       rn-requested                   (15),
       rn-with-ipm-return-requested   (20),
       ipm-auto-forwarded             (25),
       ipm-discarded                  (30),
       rn-sent                        (35) }
```

This attribute may assume any one of the following values:

a) *no-ipn-requested*: The delivered message contains no notification-requests for this IPMS-MS-user. This is a possible initial value of the attribute. If the subject recipient specifier of the delivered IPM cannot be identified, this value is assumed.

b) *nrn-requested*: The delivered message contains a non-receipt notification request for this IPMS-MS-user. This is a possible initial value of the attribute.

c) *nrn-with-ipm-return-requested*: The delivered message contains a request for non-receipt notification with return of IPM for this IPMS-MS-user. This is a possible initial value of the attribute.

d) *rn-requested*: The delivered message contains a receipt notification request for this IPMS-MS-user. This is a possible initial value of the attribute.

e) *rn-with-ipm-return-requested*: The delivered message contains a request for receipt notification with return of IPM in the event of non-receipt by this IPMS-MS-user. This is a possible initial value of the attribute.

f) *ipm-auto-forwarded*: A non-receipt-notification which indicates that the IPM was auto-forwarded has been sent; see 8.2.1.

g) *ipm-discarded*: A non-receipt-notification which indicates that the IPM was discarded has been sent; see 8.2.1.

h) *rn-sent*: A receipt notification has been sent.

NOTE – The specification of additional values of AC Submitted IPN Status for ONs may be the subject of future standardization.

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the delivery-log entry for such an object) if, and only if, that object is a delivered message whose content is an IPM. The initial value of the attribute is set in accordance with Notification-requests component of the subject recipient specifier.

19.6.5.3.3 AC Submitted IPNs

The **AC Submitted IPNs** attribute, which is multi-valued, identifies the IPNs submitted in response to a delivered IPM. The IPMS-MS shall record, by means of this attribute, any IPN submitted by the IPMS-MS-user, and any NRN caused by the invocation of the Delete abstract-operation (see 19.4), or caused as a consequence of auto-actions performed by the IPMS-MS. The attribute values are stored in ascending order of the IPN entries' creation times.

```
ac-submitted-ipns ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     SequenceNumber,
       EQUALITY MATCHING-RULE    integerMatch,
       ORDERING MATCHING-RULE    integerOrderingMatch,
       NUMERATION                multi-valued,
       ID                        id-cat-submitted-ipns }
```

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the delivery-log entry for such an object) if, and only if, that object is a delivered message whose content is an IPM.

19.6.5.3.4 Recipient Category

The **Recipient Category** attribute indicates the category of recipient (primary, copy, blind copy, or circulation list), if known, in which this IPMS-MS-user was placed by the originator of a delivered IPM. If this recipient is represented in more than one category, the attribute assumes the lowest applicable value.

```
recipient-category ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     RecipientCategory,
       EQUALITY MATCHING-RULE    integerMatch,
       ORDERING MATCHING-RULE    integerOrderingMatch,
       NUMERATION                single-valued,
       ID                        id-cat-recipient-category }

RecipientCategory ::= INTEGER {
       primary-recipient    (0),
       copy-recipient       (1),
       blind-copy-recipient (2),
       category-unknown     (3),
       circulation-list     (4) }
```

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the delivery-log entry for such an object) if, and only if, that object is a delivered message whose content is an IPM.

19.6.5.3.5 Revised Reply Time

The **Revised Reply Time** attribute enables the IPMS-MS-user to maintain a modified version of the Reply Time attribute. The IPMS-MS-user may generate this attribute using the Modify abstract-operation.

```
revised-reply-time ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX     ReplyTimeField,
       EQUALITY MATCHING-RULE    uTCTimeMatch,
       ORDERING MATCHING-RULE    uTCTimeOrderingMatch,
       NUMERATION                single-valued,
       ID                        id-cat-revised-reply-time }
```

An IPMS-MS that supports this attribute shall maintain it for an information object that it holds (and the Delivery-log entry for such an object) if, and only if, that object is a delivered message whose content is an IPM.

19.6.6 The IPMS-attribute-table information object class

For the IPMS-MS, the members of the **IPMS-attribute-table** information object set are regarded as objects populating the Content-specific-attributes information object set defined in 11.3 of ITU-T Rec. X.413 | ISO/IEC 10021-5. It is defined as follows:

```
IPMSAttributeTable ATTRIBUTE ::= {
        acknowledgment-mode | authorizing-users | auto-forward-comment |
        auto-forwarded | auto-submitted | bilaterally-defined-body-parts |
        blind-copy-recipients | body | conversion-eits | copy-recipients |
        discard-reason | encrypted-body-parts | encrypted-data | encrypted-parameters |
        expiry-time | extended-body-part-types | g3-facsimile-body-parts |
        g3-facsimile-data | g3-facsimile-parameters | g4-class1-body-parts | heading |
        ia5-text-body-parts | ia5-text-data | ia5-text-parameters | importance |
        incomplete-copy | ipm-entry-type | ipm-intended-recipient | ipm-synopsis |
        ipn-originator | languages | message-body-parts | message-data |
        message-parameters | mixed-mode-body-parts | nationally-defined-body-parts |
        non-receipt-reason | nrn-requestors | obsoleted-IPMs | originator |
        primary-recipients | receipt-time | related-IPMs | replied-to-IPM |
        reply-recipients | reply-requestors | reply-time | returned-ipm |
        rn-requestors | sensitivity | subject | subject-ipm | suppl-receipt-info |
        teletex-body-parts | teletex-data | teletex-parameters | this-ipm |
        videotex-body-parts | videotex-data | videotex-parameters,
        ...   -- 1994 extension additions -- ,
        ac-correlated-delivered-ipns | ac-correlated-delivered-replies |
        ac-delivered-ipn-summary | ac-delivered-replies-summary | ac-forwarded-ipms |
        ac-forwarding-ipms | ac-ipm-recipients | ac-obsoleted-ipms |
        ac-obsoleting-ipms | ac-related-ipms | ac-relating-ipms | ac-replied-to-ipm |
        ac-replying-ipms | ac-subject-ipm | ac-submitted-ipn-status |
        ac-submitted-ipns | ac-submitted-reply-status | authorization-time |
        body-part-encryption-token | body-part-security-label |
        body-part-signature-verification-status | body-part-signatures |
        body-parts-summary | circulation-list-recipients | distribution-codes |
        extended-subject | forwarded-content-token | forwarding-token |
        information-category | ipm-auto-discarded | ipm-security-label |
        manual-handling-instructions | notification-extensions | nrn-extensions |
        originators-reference | other-notification-type-fields | precedence |
        precedence-policy-identifier | recipient-category | revised-reply-time |
        rn-extensions }
```

19.6.7 Generation of the IPMS-specific Attributes

Table 5 summarizes the rules governing the generation of the IPMS-specific attributes. See 5.4 for a description of the classifications used. The definitive rules for the generation of attributes is given in 19.6 and 19.9.

Table 5 – Generation of the IPMS attribute-types

Attribute-type name	Single/ multi valued	Source	Generation rules
AC Correlated Delivered IPNs	M	MS	A value is generated for each delivered IPN which has been correlated with an intended recipient of the submitted IPM.
AC Correlated Delivered Replies	M	MS	A value is generated for each delivered reply which has been correlated with an intended recipient of the submitted IPM.
AC Delivered IPN Summary	M	MS	Summarizes the notification requests in a submitted IPM, correlated with the IPNs received. One value is generated for each of the IPM recipients.
AC Delivered Replies Summary	M	MS	Summarizes the reply requests in a submitted IPM, correlated with the replies received. One value is generated for each of the IPM recipients.
AC Forwarded IPMs	M	MS	The attribute-values are the sequence-numbers of the stored IPMs hat correspond to the body-part that the present child-entry represents.
AC Forwarding IPMs	M	MS	The attribute-values are the sequence-numbers of the IPMs which bear the present IPM as a Message body part.
AC IPM Recipients	M	MS	A value is generated for each recipient of the submitted IPM.
AC Obsoleted IPMs	M	MS	A value is generated for each IPM referred to in the Obsoleted IPMs Heading field of the present IPM.
AC Obsoleting IPMs	M	MS	A value is generated for each IPM which refers to the present IPM in its Obsoleted IPMs Heading field.
AC Related IPMs	M	MS	A value is generated for each IPM referred to in the Related IPMs Heading field of the present IPM.
AC Relating IPMs	M	MS	A value is generated for each IPM which refers to the present IPM in its Related IPMs Heading field.

Table 5 – Generation of the IPMS attribute-types (continued)

Attribute-type name	Single/ multi valued	Source	Generation rules
AC Replied-to IPM	S	MS	The attribute-value is the sequence-number of the IPM to which the present IPM refers in its Replied-to IPM Heading field.
AC Replying IPMs	M	MS	The attribute-values are the sequence-numbers of IPMs which refer to the present IPM in their Replied-to IPM Heading fields.
AC Subject IPM	S	MS	The attribute-value is the sequence-numbers of IPM to which the present IPM refers in its Subject IPM common field.
AC Submitted IPN Status	S	MS	The attribute-value is initially set to the value of notification-requests pertaining to this IPMS-MS-user. Assigned by auto-correlation.
AC Submitted IPNs	M	MS	The attribute-values are the sequence-numbers of IPNs whose subject IPM is this entry. Assigned by auto-correlation.
AC Submitted Reply Status	S	MS, Mod	The attribute-value is initially set to the value of reply-requested pertaining to this IPMS-MS-user. The IPMS-MS updates this value if the IPMS-MS-user generates a reply.
Acknowledgment Mode	S	RN	The attribute-value is the value of the Receipt field of the same name.
Authorization Time	S	IPM	The attribute-value is the value of the Heading Extension of the same name.
Authorizing Users	M	IPM	The attribute-values are the values of the sub-fields of the Heading field of the same name. One value is generated for each sub-field present.
Auto-forward Comment	S	NRN	The attribute-value is the value of the Non-receipt field of the same name.
Auto-forwarded	S	IPM	The attribute-value is the value of the Heading field of the same name.
Auto-submitted	S	IPM	The attribute-value is the value of the Heading Extension of the same name.
Bilaterally Defined Body Parts	M	IPM	The attribute-values are the values of the Bilaterally Defined basic body parts present in the IPM. One value is generated for each such body part.
Blind Copy Recipients	M	IPM	The attribute-values are the values of the sub-fields of the Envelope field, if present, otherwise of the Heading field of the same name. One value is generated for each sub-field present.
Body	S	IPM	The attribute-value is generated from the Body of the IPM.
Body Part Encryption Token	S	IPM	For a delivered-message main-entry the attribute-value is the value of the Delivery Envelope parameter of the same name. For a submitted-message main-entry the attribute-value is the value of the Originator-body-part-encryption-token MS-submission-extension. For a child-entry the attribute-value is the appropriate message-or-content-body-part component from this attribute-value in its parent-entry.
Body Part Security Label	M	IPM	The attribute-values are the values of the body-part-security-label sub-fields of the IPM Security Label heading extension field. One value is generated for each sub-field present.
Body Part Signature Verification Status	S	MS	The attribute-value is generated by the MS when performing body part signature verification.
Body Part Signatures	S	IPM	The attribute-value is the value of the Heading Extension of the same name.
Body Part Summary	M	MS	The value is generated for each body part present in the IPM.
Circulation List Recipients	M	IPM	The attribute-values are the values of the Heading Extension of the same name.
Conversion EITs	M	RN, ON, NRN	The attribute-values are derived from the values of the Notification common field of the same name.

Table 5 – Generation of the IPMS attribute-types (continued)

Attribute-type name	Single/ multi valued	Source	Generation rules
Copy Recipients	M	IPM	The attribute-values are the values of the sub-fields of the Heading field of the same name. One value is generated for each sub-field present.
Discard Reason	S	NRN	The attribute-value is the value of the Non-receipt field of the same name.
Distribution Codes	M	IPM	The attribute-values are the values of the Heading Extension of the same name.
Encrypted Body Parts	M	IPM	The attribute-values are the values of the Encrypted basic body parts present in the IPM. One value is generated for each such body part.
Encrypted Data	M	IPM	The attribute-values are the values of the Data components of the Encrypted basic body parts present in the IPM. One value is generated for each such body part.
Encrypted Parameters	M	IPM	The attribute-values are the values of the Parameters components of the Encrypted basic body parts present in the IPM. One value is generated for each such body part.
Expiry Time	S	IPM	The attribute-value is the value of the Heading field of the same name.
Extended Body Part Types	M	IPM	The attribute-values identify the Extended body part types represented in the IPM. (All body part types are considered to be of type Extended, regardless of whether they were so conveyed to the IPMS-MS.). One value is generated for each such type present.
Extended Subject	S	IPM	The attribute-value is the value of the Heading Extension of the same name.
Forwarded Content Token	S	IPM	For a delivered-message main-entry the attribute-value is the value of the Delivery Envelope parameter of the same name. For a submitted-message main-entry the attribute-value is the value of the Originator-forwarded-content-token MS-submission-extension. For a child-entry the attribute-value is the appropriate message-or-content-body-part component from this attribute-value in its parent-entry.
Forwarding Token	S	IPM	This attribute may only be present in a child-entry which represents a Forwarded Content body part, where that content originally had an associated message-token containing encrypted-data. The attribute-value is the value of the Forwarding-token component of the Forwarded Content Token that is associated with this Forwarded Content body part.
G3 Facsimile Body Parts	M	IPM	The attribute-values are the values of the G3 Facsimile basic body parts present in the IPM. One value is generated for each such body part.
G3 Facsimile Data	M	IPM	The attribute-values are the values of the Data components of the G3 Facsimile basic body parts present in the IPM. One value is generated for each such body part.
G3 Facsimile Parameters	M	IPM	The attribute-values are the values of the Parameters components of the G3 Facsimile basic body parts present in the IPM. One value is generated for each such body part.
G4 Class 1 Body Parts	M	IPM	The attribute-values are the values of the G4 Class 1 basic body parts present in the IPM. One value is generated for each such body part.
Heading	S	IPM	The attribute-value is the value of the Heading of the IPM.
IA5 Text Body Parts	M	IPM	The attribute-values are the values of the IA5 Text basic body parts present in the IPM. One value is generated for each such body part.
IA5 Text Data	M	IPM	The attribute-values are the values of the Data components of the IA5 Text basic body parts present in the IPM. One value is generated for each such body part.

Table 5 – Generation of the IPMS attribute-types (continued)

Attribute-type name	Single/ multi valued	Source	Generation rules
IA5 Text Parameters	M	IPM	The attribute-values are the values of the Parameters components of the IA5 Text basic body parts present in the IPM. One value is generated for each such body part.
Importance	S	IPM	The attribute-value is the value of the Heading field of the same name.
Incomplete Copy	S	IPM	The attribute-value is the value of the Heading Extension of the same name.
Information Category	M	IPM	The attribute-values are the values of the Heading Extension of the same name.
IPM Auto-discarded	S	MS	Set *false* when the entry is created; set *true* if the IPM is subsequently auto-discarded.
IPM Entry-type	S	IPM, RN, ON, NRN	The attribute-value is generated according to the abstract-operation which caused the entry to be created (see 12.1 and 12.2).
IPM Intended Recipient	S	RN, ON, NRN	The attribute-value is the value of the Notification common field of the same name.
IPM Security Label	S	IPM	The attribute-value is the value of the Heading Extension of the same name.
IPM Synopsis	S	MS	The value is generated from an analysis of the structure and types of the body parts present in the IPM.
IPN Originator	S	RN, ON, NRN	The attribute-value is the value of the Notification common field of the same name.
Languages	M	IPM	The attribute-values are the values of the Heading Extension of the same name.
Manual Handling Instructions	M	IPM	The attribute-values are the values of the Heading Extension of the same name.
Message Body Parts	M	IPM	The attribute-values are the sequence-numbers of the child-entries containing Message Body Parts. One such value is generated for each such body part in the IPM.
Message Data	M	IPM	The attribute-values are the values of the Data components of the Message basic body parts present in the IPM. One value is generated for each such body part.
Message Parameters	M	IPM	The attribute-values are the values of the Parameters components of the Message basic body parts present in the IPM. One value is generated for each such body part.
Mixed-mode Body Parts	M	IPM	The attribute-values are the values of the Mixed-mode basic body parts present in the IPM. One value is generated for each such body part.
Nationally Defined Body Parts	M	IPM	The attribute-values are the values of the Nationally Defined basic body parts present in the IPM. One value is generated for each such body part.
Non-receipt Reason	S	NRN	The attribute-value is the value of the Non-receipt field of the same name.
Notification Extensions	M	RN, ON, NRN	The attribute-values are the values of the Notification common field of the same name.
NRN Extensions	M	NRN	The attribute-values are the values of the Non-receipt field of the same name.
NRN Requestors	M	MS	The attribute-values are generated from Heading analysis.
Obsoleted IPMs	M	IPM	The attribute-values are the values of the sub-fields of the Heading field of the same name. One value is generated for each sub-field present.
Originator	S	IPM	The attribute-value is the value of the Heading field of the same name.

Table 5 – Generation of the IPMS attribute-types (continued)

Attribute-type name	Single/ multi valued	Source	Generation rules
Originator's Reference	S	IPM	The attribute-value is the value of the Heading Extension of the same name.
Other Notification Type Fields	M	ON	The attribute-values are the values of the Notification common field of the same name.
Precedence	S	IPM	The attribute-value is the value of the Recipient Extension of the same name.
Precedence Policy Identifier	S	IPM	The attribute-value is the value of the Heading Extension of the same name.
Primary Recipients	M	IPM	The attribute-values are the values of the sub-fields of the Heading field of the same name. One value is generated for each sub-field present.
Receipt Time	S	RN	The attribute-value is the value of the Receipt field of the same name.
Recipient Category	S	IPM	The attribute-value is generated from an analysis of the Primary, Copy, Blind Copy, and Circulation List Recipients Heading fields.
Related IPMs	M	IPM	The attribute-values are the values of the sub-fields of the Heading field of the same name. One value is generated for each sub-field present.
Replied-to IPM	S	IPM	The attribute-value is the value of the Heading field of the same name.
Reply Recipients	M	IPM	The attribute-values are the values of the sub-fields of the Heading field of the same name. One value is generated for each sub-field present.
Reply Requestors	M	MS	The attribute-values are generated from Heading analysis.
Reply Time	S	IPM	The attribute-value is the value of the Heading field of the same name.
Returned IPM	S	NRN	The attribute-value is the value of the Non-receipt field of the same name.
Revised Reply Time	S	Mod	The IPMS-MS-user may create, and subsequently modify the contents of this attribute.
RN Extensions	M	RN	The attribute-values are the values of the Receipt field of the same name.
RN Requestors	M	MS	The attribute-values are generated from Heading analysis.
Sensitivity	S	IPM	The attribute-value is the value of the Heading field of the same name.
Subject	S	IPM	The attribute-value is the value of the Heading field of the same name.
Subject IPM	S	RN, ON, NRN	The attribute-value is the value of the Notification common field of the same name.
Suppl Receipt Info	S	RN	The attribute-value is the value of the Receipt field of the same name.
Teletex Body Parts	M	IPM	The attribute-values are the values of the Teletex basic body parts present in the IPM. One value is generated for each such body part.
Teletex Data	M	IPM	The attribute-values are the values of the Data components of the Teletex basic body parts present in the IPM. One value is generated for each such body part.
Teletex Parameters	M	IPM	The attribute-values are the values of the Parameters components of the Teletex basic body parts present in the IPM. One value is generated for each such body part.
This IPM	S	IPM	The attribute-value is the value of the Heading field of the same name.

Table 5 – Generation of the IPMS attribute-types (concluded)

Attribute-type name	Single/ multi valued	Source	Generation rules
Videotex Body Parts	M	IPM	The attribute-values are the values of the Videotex basic body parts present in the IPM. One value is generated for each such body part.
Videotex Data	M	IPM	The attribute-values are the values of the Data components of theVideotex basic body parts present in the IPM. One value is generated for each such body part.
Videotex Parameters	M	IPM	The attribute-values are the values of the Parameters components of the Videotex basic body parts present in the IPM. One value is generated for each such body part.

19.6.8 Attributes Subject to Modification

Of the IPMS-specific attribute-types, only those listed below are subject to modification by the Modify abstract-operation and Auto-modify auto-action. Where an IPMS-MS supports one of these attributes, it shall support its modification by the Modify abstract-operation.

a) AC Submitted Reply Status

b) Revised Reply Time

19.7 IPMS-MS matching rules

A matching-rule allows entries to be selected by making an assertion about their attribute-values. Each attribute definition indicates which matching-rules (if any) can be used to make assertions about values of that attribute-type. A number of matching rules used in this Specification are defined in ITU-T Rec. X.413 | ISO/IEC 10021-5, ITU-T Rec. X.501 | ISO/IEC 9594-2, and ITU-T Rec. X.520 | ISO/IEC 9594-6. In addition to these general matching-rules which may apply to attributes of any content-type, some matching-rules are defined for use with the IPMS-specific attributes. These are defined as instances of the MATCHING-RULE information object class; see 6.3.9.3 of ITU-T Rec. X.413 | ISO/IEC 10021-5. For the IPMS-MS, the elements of the **IPM-matching-rule-table** information object set are regarded as objects populating the Content-specific-matching-rules information object set defined in 12.6 of ITU-T Rec. X.413 | ISO/IEC 10021-5. It is defined as follows:

```
IPMMatchingRuleTable MATCHING-RULE ::= {
       iPMIdentifierMatch | oRDescriptorMatch | recipientSpecifierMatch,
       . . .   -- 1994 extension additions --,
       circulationMemberCheckmarkMatch | circulationMemberElementsMatch |
       circulationMemberMatch | circulationMemberSingleElementMatch |
       circulationMemberSubstringElementsMatch | distributionCodeMatch |
       informationCategoryMatch | iPMLocationMatch | oRDescriptorElementsMatch |
       oRDescriptorSingleElementMatch | oRDescriptorSubstringElementsMatch |
       recipientSpecifierElementsMatch | recipientSpecifierSingleElementMatch |
       recipientSpecifierSubstringElementsMatch }
```

19.7.1 IPM-identifier-match

The **IPM-identifier-match** compares for equality a presented value with attribute-values of type IPM-identifier.

```
iPMIdentifierMatch MATCHING-RULE ::= {
       SYNTAX    IPMIdentifier
       ID        id-mr-ipm-identifier }
```

The rule returns *true* if, and only if, the user component of IPM identifier is present in both or absent in both values, and corresponding components match. The user component matches according to the OR-name-match rule, and the user-relative-identifier matches according to the MS-string-match rule.

19.7.2 IPM-location-match

The **IPM-location-match** compares for equality a presented value with an element of the *stored* alternative of attribute-values of type IPM Location.

```
iPMLocationMatch MATCHING-RULE ::= {
       SYNTAX    SequenceNumber
       ID        id-mr-ipm-location }
```

The rule returns *true*, if and only if, the presented value matches an element of the *stored* alternative of at least one value of the attribute according to the Integer Match rule.

19.7.3 OR-descriptor-match

The **OR-descriptor-match** compares for equality a presented value with attribute-values of type OR-descriptor.

```
oRDescriptorMatch MATCHING-RULE ::= {
       SYNTAX    ORDescriptor
       ID        id-mr-or-descriptor }
```

The rule returns *true* if, and only if, one of the following conditions is fulfilled for the presented value and at least one value of the attribute:

a) the formal-name component of OR-descriptor is present in both values, and matches according to the OR-name-match rule;

b) formal-name component of OR-descriptor is absent in either (or both) values, but free-form-name is present in both, and matches according to the MS-string-match rule.

c) the formal name component of OR-descriptor is absent in either (or both) values, but telephone-number is present in both and matches according to the MS-string-match rule.

Otherwise, the rule returns *false*.

19.7.4 OR-descriptor-elements-match

The **OR-descriptor-elements-match** determines whether a presented value is a subset of the elements present in some value of an attribute of type OR-descriptor.

```
oRDescriptorElementsMatch MATCHING-RULE ::= {
       SYNTAX    ORDescriptor
       ID        id-mr-or-descriptor-elements }
```

The rule is identical to the OR-descriptor-match rule except that the formal-name component of OR-descriptor matches using the OR-name-elements-match rule rather than the OR-name-match rule.

19.7.5 OR-descriptor-substring-elements-match

The **OR-descriptor-substring-elements-match** rule determines whether a presented value is a subset of the elements present in some value of an attribute of type OR-descriptor, where each presented string value is a substring of the corresponding stored value.

```
oRDescriptorSubstringElementsMatch MATCHING-RULE ::= {
       SYNTAX    ORDescriptor
       ID        id-mr-or-descriptor-substring-elements }
```

This rule is identical to the OR-descriptor-elements-match rule except that:

- the formal-name component matches using the OR-name-substring-elements-match rule;
- the free-form-name component matches using the MS-single-substring-match rule;
- the telephone-number matches according to the MS-single-substring-match rule.

19.7.6 OR-descriptor-single-element-match

The **OR-descriptor-single-element-match** rule determines whether a presented string and some element present in the formal-name, free-form-name, or telephone-number component of a value of an attribute of type OR-descriptor match for equality.

```
oRDescriptorSingleElementMatch MATCHING-RULE ::= {
       SYNTAX    MSString {ub-msstring-match}
       ID        id-mr-or-descriptor-single-element }
```

The rule returns true if, and only if, the stored OR-descriptor contains at least one element that matches the presented value according to the MS-string-match rule. The terminal-type and extended form of network address elements are not considered when evaluating the OR-descriptor-single-element-match rule.

19.7.7 Recipient-specifier-match

The **Recipient-specifier-match** compares for equality a presented value with the OR-descriptor components of attribute-values of type recipient specifier.

```
recipientSpecifierMatch MATCHING-RULE ::= {
       SYNTAX    RecipientSpecifier
       ID        id-mr-recipient-specifier }
```

The rule is identical to the OR-descriptor-match rule as applied to the recipient component of the presented and stored values of recipient specifier. The other components of recipient specifier are not considered.

19.7.8 Recipient-specifier-elements-match

The **Recipient-specifier-elements-match** determines whether a presented value is a subset of the elements present in some value of an attribute of type recipient specifier.

```
recipientSpecifierElementsMatch MATCHING-RULE ::= {
       SYNTAX     RecipientSpecifier
       ID         id-mr-recipient-specifier-elements }
```

The rule is identical to the OR-descriptor-elements-match rule as applied to the recipient component of the presented and stored values of recipient specifier. The other components of recipient specifier are not considered.

19.7.9 Recipient-specifier-substring-elements-match

The **Recipient-specifier-substring-elements-match** determines whether a presented value is a subset of the elements present in some value of an attribute of type recipient specifier, where each presented string value is a substring of the corresponding stored value.

```
recipientSpecifierSubstringElementsMatch MATCHING-RULE ::= {
       SYNTAX     RecipientSpecifier
       ID         id-mr-recipient-specifier-substring-elements }
```

The rule is identical to the OR-descriptor-substring-elements-match rule as applied to the recipient component of the presented and stored values of recipient specifier. The other components of recipient specifier are not considered.

19.7.10 Recipient-specifier-single-element-match

The **Recipient-specifier-single-element-match** rule determines whether a presented string and some element present in the recipient component of some value of an attribute of type recipient specifier match for equality.

```
recipientSpecifierSingleElementMatch MATCHING-RULE ::= {
       SYNTAX     MSString {ub-msstring-match}
       ID         id-mr-recipient-specifier-single-element }
```

The rule is identical to the OR-descriptor-single-element-match rule as applied to the recipient component of an attribute of type recipient specifier.

19.7.11 Circulation-member-match

The **Circulation-member-match** compares for equality a presented value with the OR-descriptor components of attribute-values of type *Circulation Member*.

```
circulationMemberMatch MATCHING-RULE ::= {
       SYNTAX     CirculationMember
       ID         id-mr-circulation-member }
```

The rule is identical to the recipient-specifier-match rule as applied to the *circulation-recipient* component of the presented and stored values of *Circulation Member*. The other components of *Circulation Member* are not considered.

19.7.12 Circulation-member-elements-match

The **Circulation-member-elements-match** determines whether a presented value is a subset of the elements present in some value of an attribute of type *Circulation Member*.

```
circulationMemberElementsMatch MATCHING-RULE ::= {
       SYNTAX     CirculationMember
       ID         id-mr-circulation-member-elements }
```

The rule is identical to the recipient-specifier-elements-match rule as applied to the *circulation-recipient* component of the presented and stored values of *Circulation Member*. The other components of *Circulation Member* are not considered.

19.7.13 Circulation-member-substring-elements-match

The **Circulation-member-substring-elements-match** determines whether a presented value is a subset of the elements present in some value of an attribute of type *Circulation Member*, where each presented string value is a substring of the corresponding stored value.

```
circulationMemberSubstringElementsMatch MATCHING-RULE ::= {
       SYNTAX     CirculationMember
       ID         id-mr-circulation-member-substring-elements }
```

The rule is identical to the recipient-specifier-substring-elements-match rule as applied to the *circulation-recipient* component of the presented and stored values of *Circulation Member*. The other components of *Circulation Member* are not considered.

19.7.14 Circulation-member-single-element-match

The **Circulation-member-single-element-match** rule determines whether a presented string and some element present in the *circulation-recipient* component of a value of an attribute of type *Circulation Member* match for equality.

```
circulationMemberSingleElementMatch MATCHING-RULE ::= {
       SYNTAX    MSString {ub-msstring-match}
       ID        id-mr-circulation-member-single-element }
```

The rule is identical to the recipient-specifier-single-element-match rule as applied to the *circulation-recipient* component of an attribute of type *Circulation Member*. The other components of *Circulation Member* are not considered.

19.7.15 Circulation-member-checkmark-match

The **Circulation-member-checkmark-match** rule compares a presented value with attribute-values of type *Circulation Member* for the presence of the checked component value.

```
circulationMemberCheckmarkMatch MATCHING-RULE ::= {
       SYNTAX    CirculationMember
       ID        id-mr-circulation-member-checkmark }
```

The rule returns *true* if, and only if, the checked component is present. Otherwise, the rule returns *false*. The other components of *Circulation Member* are not considered.

19.7.16 Distribution-code-match

The **Distribution-code-match** rule compares for equality a presented value with attribute-values of type *Distribution Code* as defined in A.1.8.

```
distributionCodeMatch MATCHING-RULE ::= {
       SYNTAX    DistributionCode
       ID        id-mr-distribution-code }
```

The rule returns *true* if, and only if, at least one of the following conditions is fulfilled for the presented value and at least one value of the attribute:

a) the *OID-code* component of the *Distribution Code* is present in both values, and matches according to the object-identifier-match rule; and either:

   i) the *alphanumeric-code* component is present in both values, and matches according to the MS-string-match rule; or

   ii) the *alphanumeric-code* component is absent from both values.

b) the *OID-code* component of the *Distribution Code* is absent from at least one of the values; and the *alphanumeric-code* component is present in both values, and matches according to the MS-string-match rule.

Otherwise, the rule returns *false*.

NOTE – The *OR-descriptor* component of the *Distribution Code* is not considered by this matching-rule.

19.7.17 Information-category-match

The **Information-category-match** rule compares for equality a presented value with attribute-values of type *Information Category* as defined in A.1.10.

```
informationCategoryMatch MATCHING-RULE ::= {
       SYNTAX    InformationCategory
       ID        id-mr-information-category }
```

The rule returns *true* if, and only if, one of the following conditions is fulfilled for the presented value and at least one value of the attribute:

a) the reference component of the *Information Category* is present in both values, and matches according to the object-identifier-match rule; and either:

   i) the description component of the *Information Category* is present in both values, and matches according to the MS-string-match rule;

   ii) the description component of the *Information Category* is absent from at least one of the values.

b) the reference component of the *Information Category* is absent from both values; and the description component of the *Information Category* is present in both values, and matches according to the MS-string-match rule.

Otherwise, the rule returns *false*.

19.8 IPMS-MS auto-actions

The IPMS-MS shall perform the general-auto-actions as specified in clause 13 of ITU-T Rec. X.413 | ISO/IEC 10021-5. In addition, this Specification defines five auto-actions that are specific to the IPMS-MS:

a) IPM auto-forward;

b) IPM auto-acknowledgement;

c) IPM auto-correlate;

d) IPM auto-discard;

e) IPM auto-advise.

Each IPMS-MS auto-action is defined as an instance of the AUTO-ACTION information object class (see 6.5.1 of ITU-T Rec. X.413 | ISO/IEC 10021-5). For the IPMS-MS, the elements of the **IPM-auto-actions** information object set are regarded as objects populating the Content-specific-auto-actions information object set defined in clause 13 of ITU-T Rec. X.413 | ISO/IEC 10021-5. It is defined as follows:

```
IPMAutoActions AUTO-ACTION ::= {
       ipm-auto-forward,
       ...   -- 1994 extension additions --  ,
       ipm-auto-acknowledgement |
       ipm-auto-correlate |
       ipm-auto-discard |
       ipm-auto-advise }
```

Each auto-action-error that may be generated by the IPM auto-actions is defined as an instance of the AUTO-ACTION-ERROR information object class. For the IPMS-MS, the elements of the **IPM-auto-action-error-table** information object set are regarded as objects populating the Content-specific-auto-action-errors information object set defined in clause 13 of ITU-T Rec. X.413 | ISO/IEC 10021-5. It is defined as follows:

```
IPMAutoActionErrorTable AUTO-ACTION-ERROR ::= {
       ...   -- 1994 extension additions --  ,
       submission-control-violated |
       element-of-service-not-subscribed |
       originator-invalid |
       recipient-improperly-specified |
       inconsistent-request |
       security-error |
       unsupported-critical-function |
       remote-bind-error |
       auto-forwarding-loop |
       duplicate-ipn |
       ipm-auto-discard-error }
```

The IPMS-MS-user may register and deregister auto-actions by subscription, or, in certain cases, by means of the Register-MS abstract-operation as described in 8.2.5 of ITU-T Rec. X.413 | ISO/IEC 10021-5. An auto-action-registration-parameter is associated with the registration of an auto-action and contains the parameters required by the IPMS-MS to perform the registered auto-action.

The operation of IPM auto-actions may be affected by the implementation of a security policy.

Table 6 summarises the registration and log generation capabilities of each of the IPM-auto-actions in the following respects:

– whether the auto-action may be registered by means of the Register-MS abstract-operation;

– whether multiple registrations are permitted;

– whether a registration parameter is defined for the auto-action;

– whether the execution of the auto-action may cause the generation of an entry in the Auto-action-log.

NOTE – If the register-MS-argument contains a registration-status-request for auto-action-registrations, then all auto-actions currently in effect (whether registered by means of Register-MS or by subscription) are reported in the registered-information argument of register-MS-result.

Table 6 – Summary of IPM-auto-action registration and logging capabilities

Auto-action-type	Register-MS	Multiple registrations	Registration parameter	Auto-action-log
IPM auto-advise	Y	Y	Y	Y
IPM auto-forward	Y	Y	Y	Y
IPM auto-acknowledgement	Y	N	Y	Y
IPM auto-correlate	N	N	N	N
IPM auto-discard	Y	Y	Y	Y

19.8.1 Auto-action performance

Table 7 shows the various events which may cause the creation of an entry in the IPMS-MS, and indicates which general and IPM-specific auto-actions are performed consequently for each type of event, and their order of execution (reading left to right).

Table 7 – Order of auto-action execution

Event	Auto-action						
	Auto-correlate-reports	IPM auto-correlate	Auto-modify	IPM auto-forward	IPM auto-advise	IPM auto-discard	Auto-alert
IPM delivery	–	Y	Y	Y	Y	Y	Y
IPN delivery	–	Y	Y	Y	–	–	Y
Report delivery	Y	–	Y	Y	–	–	Y
Other delivery	–	–	Y	Y	–	–	Y
Submit IPM	Y	Y	Y	–	–	–	–
Submit IPN	Y	Y	Y	–	–	–	–
Submit probe	Y	–	Y	–	–	–	–
Create draft	–	–	Y	–	–	–	–

Legend

Y	auto-action performed
–	not performed

NOTE 1 – The Submit IPN event arises both from the submission of an IPN by the IPMS-MS-user and from the submission of an IPN by the IPMS-MS as a secondary consequence of the performance of an abstract-operation (Fetch, Modify, Delete) or an auto-action (IPM auto-forward, IPM auto-acknowledgement, IPM auto-discard, Auto-delete).

NOTE 2 – The Submit IPM event arises both from the submission of an IPM by the IPMS-MS-user and from the submission of an auto-forwarding IPM by the IPMS-MS.

NOTE 3 – The IPM auto-discard auto-action is present in Table 7 only in regard to its effect when an obsoleting IPM is delivered; the effect of the auto-action on expired IPMs is not directly reflected in Table 7 (except that it may result in a Submit IPN event).

NOTE 4 – The IPM auto-acknowledgement, IPM auto-discard (of expired IPMs), and Auto-delete auto-actions are not performed as a consequence of the events recorded in Table 7, and are not part of the sequence of auto-action executions shown in the Table. However, all may give rise to a Submit IPN event.

NOTE 5 – The Other delivery table row indicates the performance of the IPM auto-forward auto-action when a Message whose content-type is not IPM is delivered to the IPMS-MS.

19.8.2 IPM Auto-forward

The **IPM auto-forward** auto-action enables the IPMS-MS-user to instruct the IPMS-MS to forward, automatically, any subsequently delivered information object (i.e., message or report) to another recipient or recipients. The auto-action is performed when an information object is delivered to the IPMS-MS.

One or more **IPM-auto-forward-registration-parameters** may be registered with the IPMS-MS, each identified by its registration-identifier. The IPM-auto-forward-registration-parameter specifies criteria, by means of its filter component, that determine whether this registration applies to a particular delivered object. If so, the object is auto-forwarded using the MS-message-submission abstract-operation (see 8.3.1 of ITU-T Rec. X.413 | ISO/IEC 10021-5). If the delivered object matches the criteria of more than one IPM auto-forward registration, it is auto-forwarded for each such registration.

Each IPM-auto-forward-registration-parameter contains a Message Submission Envelope argument, an IPM Heading, and, optionally, an IPM body part. These components are combined with components of the delivered object to construct the forwarding IPM.

Each IPM-auto-forward-registration-parameter specifies whether the delivered object shall be deleted after auto-forwarding. If any of the registered parameters acted upon indicates no deletion, or if any one of the submissions fails, then the entry shall not be deleted.

If the delivered object is an IPM whose originator requested notification of non-receipt, the IPMS-MS shall submit an NRN, unless the MS retrieval-status attribute retains the value new after the IPM is auto-forwarded.

When a 1994 Application Context is in use, auto-action registration is performed using the IPM-auto-forward-registration-parameter defined below. When a 1988 Application Context is in use, auto-action registration is performed using the auto-forward-registration-parameter-88 defined in Annex J.

```
ipm-auto-forward AUTO-ACTION ::= {
       REGISTRATION PARAMETER IS  CHOICE {
            ipm-auto-forward-registration-parameter IPMAutoForwardRegistrationParameter
                                                       -- used in 1994 Application Contexts only --
       ,
            auto-forward-registration-parameter-88  AutoForwardRegistrationParameter88
                                                       -- used in 1988 Application Contexts only --
       }
       ERRORS            { auto-forwarding-loop | element-of-service-not-subscribed |
                           inconsistent-request | ms-extension-error |
                           originator-invalid | recipient-improperly-specified |
                           remote-bind-error | security-error | service-error |
                           submission-control-violated | unsupported-critical-function,
                           ... }
       IDENTIFIED BY     id-act-ipm-auto-forward }

IPMAutoForwardRegistrationParameter ::= SEQUENCE {
       filter                       [0] Filter OPTIONAL,
       forwarding-envelope          [1] MessageSubmissionEnvelope,
       forwarding-heading           [2] Heading,
       forwarding-cover-note        [3] BodyPart OPTIONAL,
       submission-options           [4] MSSubmissionOptions OPTIONAL,
       nrn-comment                  [5] AutoForwardComment OPTIONAL,
       ipm-auto-forward-options     [6] IPMAutoForwardOptions DEFAULT { } }
```

The components of the **IPM-auto-forward-registration-parameter** have the following meaning:

a) **Filter** (O): This specifies a set of criteria that a newly delivered object shall satisfy before the IPMS-MS will auto-forward it using this registration.

   If this component is absent, then all newly delivered objects are auto-forwarded by this registration.

b) **Forwarding-envelope** (M): This component specifies a Message Submission Envelope for the forwarding IPM; see Figure 2 (Part 11) of ITU-T Rec. X.411 | ISO/IEC 10021-4. Before the forwarding IPM is submitted, the Envelope may be modified according to the values of the delivered object's Message Delivery (or Report Delivery) arguments. The following Message Submission arguments shall be absent from forwarding-envelope: deferred-delivery-time, latest-delivery-time, forwarding-request, and those security arguments whose values are derived from message Content. The original-encoded-information-types argument shall identify the encoded-information-types represented in the forwarding-cover-note component (if present).

c) **Forwarding-heading** (M): This component specifies a Heading for the forwarding IPM. If the delivered object is an IPM, the Heading may be modified according to the values of the delivered Heading. The following fields shall be absent from forwarding-heading, and, except for the Auto-submitted field, will assume the values of the corresponding fields (if present) in the delivered Heading: Replied-to IPM, Obsoleted IPMs, Related IPMs, Expiry Time, Reply Time, Auto-submitted, Incomplete Copy, and Languages.

d) **Forwarding-cover-note** (O): The forwarding-cover-note, if present, shall form the forwarding IPM's first body part.

e) **Submission-options** (O): This component contains submission-options that shall apply to the forwarding IPM and to the submitted NRN (if one is requested). Submission-options are specific to MS operation (see 8.1.6 of ITU-T Rec. X.413 | ISO/IEC 10021-5). The value *draft* is not permitted for the object-entry-class component. The assembly-instructions component of IPM-submission-options shall be absent from the MS-submission-extensions component. If submission-options is omitted, it assumes the value of submission-defaults, as registered by means of Register-MS; see 8.2.5.1 of ITU-T Rec. X.413 | ISO/IEC 10021-5.

f) **NRN-comment** (O): This component contains the Auto-forward Comment field of the NRN that may be returned to the object's originator, if the delivered object is an IPM. An NRN is generated if the delivered object is auto-forwarded successfully, an NRN was requested by the delivered object's originator, and the performance of the IPM-auto-forward auto-action causes the deletion of the delivered IPM or a change in the value of its retrieval-status attribute from *new* to *processe*d.

g) **IPM-auto-forward-options** (O): This component specifies IPM-auto-forward-options affecting the performance of the IPM auto-forward auto-action.

```
IPMAutoForwardOptions ::= BIT STRING {
       forward-all-object-types     (0), -- forward-all-object-types 'one', forward IPMs only
'zero' --
       include-returned-content     (1), -- include-returned-content 'one', exclude 'zero' --
       include-returned-ipm         (2), -- include-returned-ipm 'one', exclude 'zero' --
       forwarded-content-prohibited (3), -- forwarded-content-prohibited 'one', allowed 'zero'
--
       preserve-retrieval-status    (4), -- preserve-retrieval-status 'one', change 'zero' --
       delete-delivered-object      (5)  -- delete-delivered-object 'one', no deletion 'zero' --
        }
```

If forward-all-object-types is set, there is no restriction on the types of delivered object that will be auto-forwarded by this registration. If not set, then only IPMs will be auto-forwarded.

If include-returned-content is set, and the delivered object is a Report containing returned-content, this registration requests inclusion of the returned-content in the Report body part of the forwarding IPM. If not set, returned-content shall not be included.

If include-returned-IPM is set and the delivered object is an NRN containing returned-IPM, this registration requests inclusion of the returned-IPM in the Notification body part of the forwarding IPM. If not set, returned-IPM shall not be included.

If forwarded-content-prohibited is set, then the IPMS-MS shall not use the forwarded-content body part type to auto-forward a delivered message. This may be set where the recipient of the auto-forwarding message is known not to be capable of handling this body part type.

If preserve-retrieval-status is set, this registration requests that the retrieval-status of the message is left unchanged. If not set, and retrieval-status is *new* (i.e. has not been altered by a previous auto-action), then this registration requests that retrieval-status is set to *processed*.

If delete-delivered-object is set, this registration requests deletion of the delivered object after successful auto-forwarding. If not set, the delivered object shall not be deleted after auto-forwarding. If both preserve-retrieval-status and delete-delivered-object are set, then the IPMS-MS shall not accept the registration and returns a Register-MS-error.

NOTE 1 – The IPMS-MS may check the consistency of the IPM-auto-forward-registration-parameter, and may reject an inconsistent registration. For example, the IPMS-MS may verify that the recipients specified in the forwarding-envelope correspond to those specified in the forwarding-heading, and it may verify that the originator specified in the forwarding-envelope (and forwarding-heading) corresponds to an OR-name of the IPMS-MS-user. (This latter check may not be possible at registration time unless the IPMS-MS and MTA are co-located.)

The procedure for the performance this auto-action is defined in 19.9.1.2.

The performance of the IPM auto-forward auto-action may cause the creation of one or more entries in the Auto-action-log entry-class (subject to subscription to the Auto-action-log entry-class). The auto-action-errors associated with the IPM auto-forward auto-action correspond to the abstract-errors of the MS-message-submission abstract-operation; see 8.3.1 of ITU-T Rec. X.413 | ISO/IEC 10021-5.

NOTE 2 – The entry-class-error and message-group-error abstract-errors of the MS-message-submission abstract-operation are not defined as auto-action-errors of the IPM auto-forward auto-action, since both errors are detected at registration time rather than execution time of the auto-action.

In addition, the following auto-action-error is generated if an auto-forwarding loop is detected (see 18.5.3.1):

```
auto-forwarding-loop AUTO-ACTION-ERROR ::= {
       CODE    global:id-aae-auto-forwarding-loop }
```

Support of the IPM auto-forward auto-action by an IPMS-MS or an IPMS-MS-user requires that it supports the registration of the IPM-auto-forward-registration-parameter by means of the Register-MS abstract-operation.

19.8.3 IPM Auto-acknowledgement

The **IPM auto-acknowledgement** auto-action enables the IPMS-MS-user to instruct the IPMS-MS to automatically originate RNs on the user's behalf. The auto-action is performed on the first occasion on which the retrieval-status of an

IPM entry in the Delivery entry-class changes to *processed* (the same change may occur on subsequent occasions if the IPMS-MS-user uses the Modify abstract-operation to change retrieval-status back to *listed*; the auto-action is not performed in these cases. The RN is originated only if RN was requested of this user for this IPM by means of the Notification-requests component of the subject recipient specifier. The RN shall not be generated if suspend-auto-acknowledgement was specified in the bind-extensions parameter of the MS-bind abstract-operation which established the present abstract-association (see 19.5.1). The RN shall have the common fields and receipt fields as prescribed in 18.5.2.1 and shall be submitted as prescribed in 18.5.2.2.

NOTE 1 – No RN should be generated for an IPM which has been the subject of DL-expansion.

NOTE 2 – An RN is not originated for an entry whose retrieval-status changes to *processed* as a consequence of IPM auto-forwarding.

NOTE 3 – An abstract-association might terminate abnormally after an RN has been submitted by the IPMS-MS, but before the fetch-result that caused the IPM auto-acknowledgement has been received by the IPMS-MS-user. The IPMS-MS has no detection or recovery mechanisms for this case.

The IPM-auto-acknowledgement-registration-parameter may specify the Suppl Receipt Info field of each RN generated by IPM auto-acknowledgement, and the submission-options that shall apply (see 8.1.6 of ITU-T Rec. X.413 | ISO/IEC 10021-5). In submission-options, the value *draft* is not permitted for the object-entry-class component; the MS-submission-extensions component shall be absent. If the submission-options parameter is omitted, it assumes the value of submission-defaults, as registered by means of Register-MS; see 8.2.5.1 item (h) of ITU-T Rec. X.413 | ISO/IEC 10021-5.

```
ipm-auto-acknowledgement AUTO-ACTION ::= {
       REGISTRATION PARAMETER IS    IPMAutoAcknowledgementRegistrationParameter
       ERRORS                 { recipient-improperly-specified | inconsistent-request |
                                element-of-service-not-subscribed | originator-invalid |
                                unsupported-critical-function | remote-bind-error |
                                submission-control-violated | security-error |
                                duplicate-ipn}
       IDENTIFIED BY          id-aa-ipm-auto-acknowledgement }

IPMAutoAcknowledgementRegistrationParameter ::= SET {
       auto-acknowledge-suppl-receipt-info   [0] SupplReceiptInfoField OPTIONAL,
       submission-options                    [1] MSSubmissionOptions OPTIONAL }
```

The performance of the IPM auto-acknowledgement auto-action may cause the creation of an entry in the Auto-action-log entry-class, subject to subscription to the Auto-action-log entry-class. Where an IPN has already been generated for a delivered IPM, (except for an auto-forwarded IPM where an NRN indicating IPM-auto-forwarded has already been sent), IPM auto-acknowledgement fails and generates the following error:

```
duplicate-ipn AUTO-ACTION-ERROR ::= {
       CODE    global:id-aae-duplicate-ipn }
```

Support of the IPM auto-acknowledgement auto-action by an IPMS-MS, or an IPMS-MS-user, requires that it supports a single registration of the IPM-auto-acknowledgement-registration-parameter by means of the Register-MS abstract-operation. The registration-identifier component of auto-action-registration shall be absent when registration is requested. The IPM auto-acknowledgement auto-action shall not be subscribed to unless the AC Submitted-IPN-Status attribute is also subscribed to.

An IPMS-MS which supports the IPM auto-acknowledgement auto-action shall support the suspend-auto-acknowledgement extension defined in 19.5.1.

19.8.4 IPM Auto-correlate

The **IPM auto-correlate** auto-action correlates IPMs and IPNs related in the following ways:

a) an IPM and the IPMs generated in reply;

b) an IPM and the IPNs notifying receipt or non-receipt;

c) an IPM and the IPMs which subsequently forward it, or obsolete it, or are related to it.

The auto-action also correlates response requests made of the IPMS-MS-user with any corresponding replies or IPNs subsequently generated by this user, or by the IPMS-MS in the performance of some other auto-action. The auto-action is performed whenever an IPM or IPN is submitted or delivered. The IPM auto-correlate auto-action is provided by subscription only, and not by registration using the Register-MS abstract-operation of ITU-T Rec. X.413 | ISO/IEC 10021-5.

```
ipm-auto-correlate AUTO-ACTION ::= {
       IDENTIFIED BY    id-aa-ipm-auto-correlate }
```

The IPMS-specific attributes which support IPM auto-correlate are defined in 19.6.5, and the additional procedures necessary for the support of this auto-action are defined in 19.9.1.1. The performance of the IPM auto-correlate auto-action shall not cause the creation of an entry in the Auto-action-log entry-class.

19.8.5 IPM Auto-discard

The **IPM auto-discard** auto-action enables the IPMS-MS-user to instruct the IPMS-MS to automatically delete a delivered IPM entry (and any child-entries associated with it) when the date and time denoted by its Expiry Time field has passed or when a subsequently delivered IPM renders it obsolete. IPM auto-discard shall not be performed while an abstract-association exists between the IPMS-MS and the IPMS-MS-user. When it auto-discards an IPM, the IPMS-MS originates an NRN on the user's behalf if, and only if, an NRN is requested of this user by means of the Notification-requests component of the subject recipient specifier and the retrieval-status does not have the value processed. The NRN shall have the common fields and non-receipt fields as prescribed in 18.5.1.2 and shall be submitted as prescribed in 18.5.1.3.

```
ipm-auto-discard AUTO-ACTION ::= {
       REGISTRATION PARAMETER IS    IPMAutoDiscardRegistrationParameter
       ERRORS                  { submission-control-violated | inconsistent-request |
                                 originator-invalid | recipient-improperly-specified |
                                 unsupported-critical-function | security-error |
                                 element-of-service-not-subscribed |
                                 remote-bind-error | ipm-auto-discard-error}
       IDENTIFIED BY           id-aa-ipm-auto-discard }

IPMAutoDiscardRegistrationParameter ::= SET {
       filter                            [0]  Filter OPTIONAL,
       submission-options                [1]  MSSubmissionOptions OPTIONAL,
       auto-discard-expired-ipms         [2]  BOOLEAN,
       auto-discard-obsoleted-ipms       [3]  BOOLEAN,
       restrict-obsoleting-to-originator [4]  BOOLEAN }
```

The components of IPM-auto-discard-registration-parameter have the following meaning:

a) **Filter** (O): This specifies a filter which an expired or obsoleted message shall satisfy before IPM auto-discard is performed; if omitted, all expired and obsoleted IPMs are auto-discarded (subject to Items c, d and e).

   NOTE – The IPMS-MS-user may prevent IPM auto-discard from acting on entries with MS retrieval-status new, by constructing a Filter which excludes such entries.

b) **Submission-options** (O): This specifies submission requests for the submitted NRN (see 8.1.6 of ITU-T Rec. X.413 | ISO/IEC 10021-5). The value *draft* is not permitted for the object-entry-class component; the MS-submission-extensions component shall be absent. If the submission-options is omitted, it assumes the value of general-submission-defaults, as registered by means of Register-MS; see 8.2.5.1 of ITU-T Rec. X.413 | ISO/IEC 10021-5.

c) **Auto-discard-expired-IPMs** (M): If *true*, and if the filter is satisfied, then expired IPMs shall be auto-discarded.

d) **Auto-discard-obsoleted-IPMs** (M): If *true*, and if the filter is satisfied, then obsoleted IPMs shall be auto-discarded.

e) **Restrict-obsoleting-to-originator** (M): If *true*, then an IPM shall be deemed to be obsolete only if the obsoleted IPM and the IPM carrying the obsoleting-indication were originated by the same user.

The performance of the IPM auto-discard auto-action may cause the creation of an entry in the Auto-action-log entry-class, subject to subscription to the Auto-action-log entry-class. Where an IPM is not auto-discarded because the restrict-obsoleting-to-originator component is *true*, the IPMS-MS shall assign an IPM-auto-discard-error with the value *not-obsoleted-by-originator* to the auto-action-error attribute in the corresponding Auto-action-log entry:

```
ipm-auto-discard-error AUTO-ACTION-ERROR ::= {
       PARAMETER      SET {
                           problem   [0]  AutoDiscardProblem }
       CODE           global:id-aae-auto-discard-error }

AutoDiscardProblem ::= INTEGER {
       not-obsoleted-by-originator  (0) }
```

If an IPM is auto-discarded and the Message-log entry-class is subscribed to then an IPM-auto-discarded attribute is added to the Message-log entry and assigned the value *true*.

Support for the IPM auto-discard auto-action by an IPMS-MS, or an IPMS-MS-user, requires that it supports the registration of the IPM-auto-discard-registration-parameter by means of the Register-MS abstract-operation.

19.8.6 IPM auto-advise

The **IPM auto-advise** auto-action enables the IPMS-MS-user to instruct the IPMS-MS to generate ANs automatically that will convey additional advice to the originators of subsequently delivered IPMs. The auto-action is performed when an IPM is delivered to the IPMS-MS.

One or more **IPM auto-advise-registration-parameters** may be registered with the IPMS-MS, each identified by its registration-identifier, that specifies criteria which determine whether this registration applies to a particular delivered IPM. If so, an AN is generated to inform the IPM's originator of the AN originator's absence.

```
ipm-auto-advise AUTO-ACTION ::= {
       REGISTRATION PARAMETER IS  IPMAutoAdviseRegistrationParameter
       ERRORS                     {inconsistent-request |
                                   element-of-service-not-subscribed |
                                   originator-invalid |
                                   recipient-improperly-specified |
                                   remote-bind-error | security-error |
                                   submission-control-violated |
                                   unsupported-critical-function }
       IDENTIFIED BY              id-aa-ipm-auto-advise }

IPMAutoAdviseRegistrationParameter ::= SET {
       enabled                           [0] BOOLEAN DEFAULT TRUE,
       filter                            [1] Filter OPTIONAL,
       advice-notifications              [2] SET OF IPMSExtension
                                                 {{AdviceNotifications}},
       suppress-subsequent-notifications [3] BOOLEAN DEFAULT TRUE,
       use-ipm-if-an-not-supported       [4] BOOLEAN DEFAULT FALSE,
       submission-options                [5] MSSubmissionOptions OPTIONAL }
```

The components of IPM-auto-advise-registration-parameter have the following meaning:

a) **Enabled** (D *true*): This specifies whether each registered IPM auto-advise auto-action is enabled or disabled. If *true*, then the registered IPM auto-advise auto-action is enabled, and remains so until a subsequent registration disables it. If *false*, then the registered IPM auto-advise auto-action is disabled, and remains so until a subsequent registration enables it, but its other parameters remain registered so that it may be re-enabled easily.

b) **Filter** (O): This specifies a filter that the delivered IPM shall satisfy before IPM auto-advise is performed using this registration. If this component is absent then all delivered IPMs are subject to this registered IPM auto-advise auto-action.

c) **Advice-notifications** (M): This contains the value that is to be supplied as the other notification type fields of each AN conveyed to the originator of the delivered IPM; see 8.4. This may convey absence advice or change of address advice (or both).

d) **Suppress-subsequent-notifications** (D *true*): If *true*, then the delivered IPM's originator will receive only a single AN. Subsequently delivered IPMs from the same originator will not be subject to the IPM auto-advise auto-action. If *false*, then IPM auto-advise shall apply to subsequently delivered IPMs regardless of whether an AN was previously sent to the same IPM originator. For any given originator, the suppression of subsequent notifications is cancelled when the auto-action is next disabled (and re-enabled).

e) **Use-IPM-if-AN-not-supported** (D *false*): If *true*, and the Notification-requests component of the subject recipient specifier does not indicate that ANs are supported (i.e. *an-supported* is not set), then the MS may send an IPM instead of the AN if the MS has the capability to render into a text body part of the IPM the information that would have been conveyed by the AN. The IPM shall include the auto-submitted indication.

   NOTE – The facility to render an AN into an IPM is a transition aid to enable the contents of auto-advise notifications to be conveyed to UAs which do not have the capability to render ANs. Use of ANs is preferred, where available, to provide a more appropriately unobtrusive service and to permit ANs to be correlated. The precise manner in which the information in the AN is rendered into the IPM is implementation dependent.

f) **Submission-options** (O): This specifies submission requests for the submitted AN (see 8.1.6 of ITU-T Rec. X.413 | ISO/IEC 10021-5). The value *draft* is not permitted for the object-entry-class component; the MS-submission-extensions component shall be absent. If the submission-options is omitted, it assumes the value of general-submission-defaults, as registered by means of Register-MS; see 8.2.5.1 of ITU-T Rec. X.413 | ISO/IEC 10021-5.

The additional procedures necessary for the support of this auto-action are defined in 19.9.1.4. The performance of the IPM auto-advise auto-action shall cause the creation of an entry in the Auto-action-log entry-class. The value of retrieval-status is not affected by the execution of this auto-action.

Support for the IPM auto-advise auto-action by an IPMS-MS, or IPMS-MS-user, requires that it supports the registration of the IPM-auto-advise-registration-parameter by means of the Register-MS abstract-operation.

19.9 Procedures for the IPMS-MS

The procedures for a general (content-independent) MS are described in clauses 15 and 16 of ITU-T Rec. X.413 | ISO/IEC 10021-5. Additional procedures required to support the operation of an IPMS-MS are defined here.

19.9.1 Additional procedures for Message-delivery and Report-delivery

The performance of the Message-delivery and Report-delivery abstract-operations is described in 15.1.1 and 15.1.2, respectively, of ITU-T Rec. X.413 | ISO/IEC 10021-5. Additions to item (c) in 15.1.1, required for the support of Message-delivery in Interpersonal Messaging, are described in 19.9.1.1 – 19.9.1.3 below. Additions to item (c) in 15.1.2, required for the support of Report-delivery in Interpersonal Messaging, are described in 19.9.1.2 below.

19.9.1.1 Additional procedures for IPM auto-correlate

If the IPM auto-correlate auto-action is subscribed to, the IPMS-MS performs the following actions:

a) If the delivered message contains an IPM whose Replied-to IPM heading field is present, the IPMS-MS shall attempt to locate the entry identified by the Replied-to IPM field by searching main-entries of all the entry-classes except the Draft and Auto-action-log entry-classes. If this entry is found (the replied-to entry), its sequence-number is recorded in the AC Replied-to IPM attribute of the present entry. Similarly, the replied-to entry has its AC Replying IPMs attribute updated to reference the present entry.

   If the replied-to entry described above is located in the Submission or Submission-log entry-class, the following additional actions are taken. The Originator of the present IPM is matched against the list of recipients held in the replied-to entry's AC IPM Recipients attribute. If a match is found, then the IPMS-MS shall update the value corresponding to that recipient in the replied-to entry's AC Correlated Delivered Replies attribute to cause it to reference the present IPM. In addition, the value corresponding to the same recipient in the replied-to entry's AC Delivered Replies Summary attribute shall be updated to record that a reply was received from that recipient.

   NOTE – If no match is found for the Originator, the IPMS-MS may attempt to match one of the OR-names in DL-expansion-history, if present in the delivered IPM, against the list of recipients held in the replied-to entry's AC IPM Recipients attribute.

   If the delivered message contains an IPM whose This IPM heading field matches a subfield of the Replied-to IPM heading field of a stored IPM, then the sequence-number of each such stored IPM is recorded in the AC Replying IPMs attribute of the present entry. In addition, the AC Replied-to IPM attribute of each such stored IPM is updated to reference the present entry.

b) If the delivered message contains an IPM, then each of the recipient specifier values present in its Primary, Copy, Blind Copy, and Circulation List Recipients fields is matched against the originally-intended-recipient-name argument, or if that argument is not present, the this-recipient-name argument. If any recipient specifier matches the intended recipient, then the following actions are taken. If the recipient specifier's reply-requested component indicates that a reply is requested, then the IPMS-MS shall create an AC Submitted Reply Status attribute containing the value *reply-pending*; otherwise the value *no-reply-requested* is assigned. If the recipient specifier's Notification-requests component indicates that an IPN is requested, and the message contains no DL-expansion-history, then the IPMS-MS shall create an AC Submitted IPN Status attribute containing the value *nrn-requested*, *nrn-with-ipm-return-requested*, *rn-requested*, or *rn-with-ipm-return-requested* as appropriate; otherwise the value *no-ipn-requested* is assigned. The IPMS-MS shall create a Recipient Category attribute set according to whether a recipient specifier was found which matches the intended recipient, and if so, the category of recipient field in which the recipient specifier was found.

   NOTE – This procedure attempts to identify which recipient specifier caused the delivery of this copy of the message by comparing the recipient specifiers against information taken from the message-delivery-envelope. At the time of submission, the OR-names of the recipients in the message submission envelope will comprise all of the OR-names from the recipient specifiers in the IPM heading. At the time of delivery, the this-recipient-name component of the message delivery envelope contains the OR-name which caused delivery of this copy of the message. If the message has not been subject to DL-expansion or Redirection, the value of this-recipient-name will be one of the values from the original message-submission-envelope, and can be expected to match one of the recipient specifiers in the heading. If DL-expansion or Redirection has occurred, this-recipient-name will not contain a value supplied by the originator, but in this case the originally-intended-recipient-name will be present in the envelope, containing the value supplied by the originator which was replaced by the first Redirection or DL-expansion. Hence the procedure will identify the relevant recipient specifier regardless of DL-expansion or Redirection. This is desirable behaviour in the case of the Recipient Category and AC Submitted Reply Status attributes, but the DL-expansion service requires that receipt notifications are not generated in response to messages received as a result of DL-expansion (in order that the membership of the DL may be kept confidential) and hence the AC Submitted IPN Status attribute has to be set explicitly to the value no-ipn-requested in this case.

   If the delivered message contains an IPM whose This IPM heading field is identical with that of a (previously delivered) stored IPM, then the value of the delivered IPM's AC Submitted Reply Status attribute and that of the corresponding attribute of the stored IPM shall be made identical, the higher value taking precedence.

c) If the delivered message contains one or more forwarded IPMs, either as Message or Forwarded Content body parts, the IPMS-MS attempts to locate any corresponding stored IPM entries by searching main-entries of all the entry-classes, except the Draft and Auto-action-log entry-classes, and matching on IPM identifier. For each forwarded IPM, all matching stored IPM entries found have their AC Forwarding IPMs attribute updated to record the sequence-number of the delivered message entry. In addition, for each child-entry of the delivered message which contains a forwarded IPM, the AC Forwarded IPMs attribute is updated to record the sequence-numbers of the matching stored IPM entries.

d) If the delivered message contains an IPM whose Related IPMs heading field is present, the IPMS-MS shall attempt to locate all the entries identified by each subfield of the Related IPMs field by searching entries of all the entry-classes except the Draft and Auto-action-log entry-classes. If any such entries are found, each has its AC Relating IPMs attribute updated to record the sequence-number of the delivered message entry. In addition, the AC Related IPMs attribute of the delivered message entry is updated to record the sequence-numbers of the related IPMs.

   If the delivered message contains an IPM whose This IPM heading field matches a subfield of the Related IPMs heading field of a stored IPM, then the AC Related IPMs attribute of each such stored IPM is updated to record the sequence-number of the delivered IPM. In addition the AC Relating IPMs attribute of the delivered IPM is updated to record the sequence-numbers of the stored IPMs.

e) If the delivered message contains an IPM whose Obsoleted IPMs heading field is present, the IPMS-MS shall attempt to locate all the entries identified by each subfield of the Obsoleted IPMs field by searching entries of all the entry-classes except the Draft and Auto-action-log entry-classes. If any such entries are found, each has its AC Obsoleting IPMs attribute updated to record the sequence-number of the delivered message entry. In addition, the AC Obsoleted IPMs attribute of the delivered message entry is updated to record the sequence-numbers of the obsoleted IPMs.

   If the delivered message contains an IPM whose This IPM heading field matches a subfield of the Obsoleted IPMs heading field of a stored IPM, then the AC Obsoleted IPMs attribute of each such stored IPM is updated to record the sequence-number of the delivered IPM. In addition the AC Obsoleting IPMs attribute of the delivered IPM is updated to record the sequence-numbers of the stored IPMs.

f) If the delivered message contains an IPN, the IPMS-MS shall attempt to locate all entries which may correspond to the IPN's subject IPM by searching the Submission (and Submission-log) entry-classes. If any such entries are found, the IPMS-MS shall perform the following actions on each such entry in turn. The sequence-number of the subject IPM entry is recorded in the AC Subject IPM attribute of the present entry. The IPN's IPM Intended Recipient field (or, if this field is absent, the IPN's IPN Originator field) is compared against the list of recipients recorded in the subject IPM's AC IPM Recipients attribute. If a match is found, then the IPMS-MS shall update the value corresponding to that recipient in the subject IPM's AC Correlated Delivered IPNs attribute to cause it to reference the present entry. In addition, the corresponding value in the subject IPM's AC Delivered IPN Summary attribute is updated to indicate that an IPN has been received from that recipient (or from the IPN originator to whom delivery of the subject IPM occurred as a consequence of its being addressed to that recipient).

19.9.1.2 Additional procedures for IPM auto-forward

If the IPM auto-forward auto-action is subscribed to the IPMS-MS performs the following actions:

a) The delivered object is matched against the filter specified in each IPM-auto-forward-registration-parameter, in turn. For each registration in which the delivered object satisfies the filter, steps (b) to (i) are followed.

b) If the forward-all-object-types option is registered, or if the delivered object is an IPM, the object is subject to auto-forwarding. Otherwise, the object will not be auto-forwarded and the IPMS-MS resumes processing the next registered IPM-auto-forward auto-action.

   NOTE – This option ensures that, by default, only IPMs will be auto-forwarded. The same condition could be set using filter, which enables the specification of general selection criteria, but the option provides a simple mechanism for this common requirement.

c) An entry is created in the Auto-action-log entry-class. If an error in processing this registered auto-action occurs, the error is recorded by attaching an auto-action-error attribute to the auto-action-event entry (see 6.5.3 of ITU-T Rec. X.413 | ISO/IEC 10021-5).

d) If the delivered object is an IPM and an auto-forwarding loop is detected (see 18.5.3.1), the auto-forwarding-loop auto-action-error is recorded. The procedure then terminates, and no further IPM auto-forward registrations are processed.

e) The forwarding IPM is constructed as follows:

1) The Body is constructed from the registered forwarding-cover-note (if present), and the delivered object. The registered forwarding-cover-note, if present, forms the first body part. The second body part (or, if the forwarding-cover-note is absent, the only body part) is one of the following, depending on the type of the delivered object:

i) if the delivered object is an IPN, then a Notification body part (if the include-returned-IPM option is registered, and a returned-IPM is present in the IPN it shall also be present in the body part); or,

ii) if the delivered object is a Report, then a Report body part (if the include-returned-content option is registered, and a returned-content is present in the Report it shall also be present in the body part); or,

iii) if the delivered object is an IPM, and the message-origin-authentication-check, content-confidentiality-algorithm-identifier, content-integrity-check, and message-token arguments are absent from the message envelope, or any of the arguments is present, but the forwarded-content-prohibited option is registered, then a Message body part; or,

iv) if the forwarded-content-prohibited option is registered, then the message does not satisfy the criteria of this registered auto-action and the IPMS-MS considers the next registered auto-action;

v) a Forwarded Content body part otherwise.

2) The Heading is constructed from the registered forwarding-heading and the delivered Heading (present if the delivered object is an IPM):

i) If the Subject field, Extended Subject heading extension, or Sensitivity field is absent from the forwarding-heading, each assumes the value (if any) present in the delivered Heading.

ii) The Importance field assumes the higher of the values present in the forwarding-heading and delivered Heading. The Precedence recipient extension assumes the higher of the values (if any) present in the forwarding-heading and delivered Heading, irrespective of the Precedence Policy Identifier associated with that value. The Precedence Policy Identifier heading extension assumes the value (if any) present in the forwarding-heading or delivered Heading as appropriate depending on which value was selected for the Precedence.

iii) The following fields of the forwarding IPM's Heading assume the values, if present, of the corresponding fields of the delivered Heading, and are absent otherwise: Replied-to IPM, Obsoleted IPMs, Related IPMs, Expiry Time, Reply Time, Incomplete Copy, Languages, Distribution Codes, Information Category, Manual Handling Instructions, and Originator's Reference.

iv) The Reply Recipients field assumes the value, if present, of the corresponding field of the forwarding-heading, and is absent otherwise.

v) If the delivered object is an IPM, the User-relative-identifier component of its This IPM field is appended to the corresponding component of the forwarding-heading's This IPM field. Otherwise, a value generated by the IPMS-MS is appended.

NOTE – The IPMS-MS may make other modifications to the User-relative-identifier, if necessary, to ensure that the This IPM field uniquely and unambiguously identifies the forwarding IPM (see 7.1.1), and satisfies any size constraints.

vi) The IPMS-MS shall create an Auto-forwarded Heading field with the value *true*.

vii) If the delivered object is an IPN, then the notification-requests component of each recipient-specifier present in Primary, Copy, Blind Copy, and Circulation List Recipients is deleted.

3) The Envelope is constructed from the registered forwarding-envelope and the delivered Envelope (present if the delivered object is a Message):

i) The priority argument assumes the value present in forwarding-envelope, or the delivered Envelope, whichever has the higher priority.

ii) If the value *conversion-with-loss-prohibited* is specified for the conversion-with-loss-prohibited argument either in forwarding-envelope, or in the delivered Envelope, then that value is assumed. If the value *implicit-conversion-prohibited* is specified for the implicit-conversion-prohibited argument either in forwarding-envelope, or in the delivered Envelope, then that value is assumed.

iii) The original-encoded-information-types argument shall be the union of those values specified in the same argument of forwarding-envelope, and one of following dependent on the body part type for the forwarded object selected in item e) 1):

Message	the encoded-information-types specified in the delivered Envelope (from the converted-encoded-information-types argument, if present, or original-encoded-information-types otherwise);
Forwarded Content	the encoded-information-type for the Forwarded Content body part specified in 7.4.16;
Notification	no additional encoded-information-types;
Report	no additional encoded-information-types.

iv) If the delivered object is a Report, then for each recipient-name specified in the Envelope under construction, the originator-report-request is given the value no-report.

NOTE – If the delivered object is a Report, the Envelope is specified solely by the registered forwarding-envelope.

f) The IPMS-MS combines the Body, Heading, and Envelope to form the forwarding IPM. The IPMS-MS then stages a performance of the MS-message-submission abstract operation with the forwarding IPM and the registered submission-options as its arguments, and the procedure defined in 19.9.2 is followed.

g) If the submission fails, the error is recorded by attaching an auto-action-error attribute to the auto-action-event entry.

h) If the submission is successful, the IPMS-MS verifies the following:

1) that an NRN reporting non-receipt of the delivered object has not already been submitted;

2) that the registered IPM-auto-forward-options do not specify preserve-retrieval-status;

3) that the delivered object is an IPM whose originator requested an NRN by means of the notification-requests component of the subject recipient specifier.

If these conditions are fulfilled, then the IPMS-MS shall submit an NRN. The IPMS-MS draws the NRN's Auto-forward Comment field from the registered NRN-comment, if present. Other fields of the NRN are constructed as specified in 18.5.3.4. The IPMS-MS stages a performance of the MS-message-submission abstract-operation with the NRN and the registered submission-options as its arguments, and the procedure defined in 19.9.2 is followed.

i) If an entry for the forwarding IPM was created in the Submission (or Submission-log) entry-class, then the MS general-attribute MS-originated is created and assigned the value *true*. The IPMS-MS then resumes processing the next registered IPM-auto-forward auto-action.

j) Once all registered IPM auto-forward-registration-parameters have been processed, the procedure continues as follows.

If at least one of the IPM auto-forward auto-actions is performed successfully, and at least one of the registered IPM-auto-forward-registration-parameters whose criteria were satisfied by the delivered object did not request preserve-retrieval-status, then the delivered object's MS retrieval-status is set to *processed*. This change in retrieval-status does not cause the performance of the IPM auto-acknowledgement auto-action.

k) If at least one of the IPM auto-forward auto-actions is performed successfully, and all of the registered IPM-auto-forward-registration-parameters whose criteria were satisfied by the delivered object requested delete-delivered-object, then the IPMS-MS shall delete the delivered object.

19.9.1.3 Additional procedures for IPM auto-discard

If the IPM auto-discard auto-action is subscribed to, and the IPMS-MS-user has registered at least one which requests auto-discard of obsoleted IPMs, the IPMS-MS performs the following actions:

a) If the delivered message contains an IPM whose Obsoleted IPMs heading field is present, the IPMS-MS shall attempt to identify an entry corresponding to each Obsoleted IPM by searching entries of the Stored-message entry-class. If any such entries are found, only those which satisfy the IPM auto-discard Filter shall be considered further.

If restrict-obsoleting-to-originator has been set to *true*, then the originator-name attribute of each such entry shall be compared for equality with the originator-name attribute of the delivered message and only those entries which match shall be considered further. For each occasion on which matching fails, the IPMS-MS shall generate an Auto-action-log entry and assign an IPM-auto-discard-error with the value *not-obsoleted-by-originator* to its auto-action-error attribute.

b) For each entry selected in step (a) that has a retrieval-status of *new* or *listed* then the IPMS-MS shall construct an NRN as specified in 18.5.1.2 if, and only if, one is requested by means of the Notification-requests component of the IPM's subject recipient specifier. The NRN is submitted by invoking MS-message-submission, using the submission-options parameter registered for the IPM auto-discard auto-action, and the procedures defined in 19.9.2 are followed.

c) Each of the entries selected in step (a) shall be deleted by the IPMS-MS. If the Message-log entry-class is subscribed to then an IPM-auto-discarded attribute is added to the corresponding Message-log entry and assigned the value *true*.

NOTE – If the IPMS-MS is able to determine that the delivery of an IPM, now auto-discarded, caused an alert condition that is still outstanding, and would not be in effect but for that delivered IPM, it may set the alert-indication to *false*; see 7.1.2 of ITU-T Rec. X.413 | ISO/IEC 10021-5.

d) Auto-discard auto-action processing continues until all the obsoleted IPMs have been deleted, or all the registrations which contain a request to auto-discard obsoleted IPMs have been processed.

19.9.1.4 Additional procedures for IPM auto-advise

If the IPM auto-advise auto-action is subscribed to, the IPMS-MS performs the following actions when an IPM is delivered:

a) If any of the following conditions apply, then the procedure terminates and no further IPM auto-advise registrations are considered:

i) the Notification-requests component of the subject recipient specifier requests suppression of advice notification (*suppress-an*);

ii) an NRN has already been generated as a consequence of the performance of the IPM auto-forward auto-action;

iii) the IPM was delivered as a result of DL-expansion;

iv) a previously delivered IPM from the same originator caused the execution of an IPM auto-advise auto-action whose registration-parameter specified suppress-subsequent-notifications.

NOTE – Suppression only applies to IPMs delivered after that IPM auto-advise registration was last enabled.

b) The delivered IPM is matched against the filter specified in the IPM auto-advise registration, provided that the registration is enabled. If the filter is satisfied, the procedure continues at step (c). Otherwise, the present step is repeated using the next registered IPM auto-advise auto-action. If all IPM auto-advise auto-action registrations have been considered, the procedure terminates.

c) If the Notification-requests component of the subject recipient specifier indicates that ANs are supported (i.e. *an-supported* is *true*) the procedure continues at step (d). Otherwise, if use-IPM-if-AN-not-supported is *false* then step (b) is repeated using the next registered IPM auto-advise auto-action. If use-IPM-if-AN-not-supported is *true* then steps (d) and (e) are followed except that an IPM is constructed and submitted containing the information which would otherwise have been encoded in the AN.

d) The AN is constructed with the common fields prescribed for auto-acknowledgement (see 18.5.2.1). In addition the AN shall contain the advice notification fields drawn from the Registration component of the IPM-auto-advise-registration-parameter.

e) The AN is submitted by invoking MS-message-submission, using the submission-options parameter registered for the IPM auto-advise auto-action, and the procedures defined in 19.9.2 are followed. The procedure then terminates.

19.9.2 Additional Procedures for MS-message-submission

Procedures for the invocation of the Message-submission abstract-operation and for the performance of the MS-message-submission abstract-operation are defined in 15.2.1 and 16.2.1, respectively, of ITU-T Rec. X.413 | ISO/IEC 10021-5.

The following two items are additions to 16.2.1 (a) and (b) of ITU-T Rec. X.413 | ISO/IEC 10021-5, and apply where the IPMS-MS-user invokes MS-message-submission:

a) If a 1994 Application Context is in use and the submitted message contains an IPM, and the submission-options parameter of the MS-message-submission argument contains an IPM assembly instructions parameter, then the IPMS-MS shall verify that each stored-IPM specified in the IPM assembly instructions parameter refers to an IPM entry, and that each body-part-number specified is present either in the associated stored-entry or in the submitted IPM, as indicated. The forwarding IPM's Body is constructed as indicated in 19.5.3.1.

b) If a 1988 Application Context is in use, and the submitted message contains an IPM, the IPMS-MS examines the MS-message-submission argument for the presence of a forwarding-request parameter. If present, the IPMS-MS verifies that the entry to be forwarded is a delivered IPM and constructs the forwarding IPM's Body as indicated in 19.5.5.

If the IPM auto-correlate auto-action is subscribed to, then the following additions are required to 16.2.1 (f) of ITU-T Rec. X.413 | ISO/IEC 10021-5, and apply both in the case where the IPMS-MS-user invokes MS-message-submission, and where the IPMS-MS invokes Message-submission as a consequence of auto-action processing. This procedure is described for the case where an entry is created in the Submission-log (or Submission) entry-class; if the submission options and subscription details are such that no entry is created, the maintenance of the AC Submitted IPN Status and AC Submitted Reply Status attributes shall be performed as described in steps c) and e), but the remaining steps are not applicable.

c) If the submitted message contains an IPN, the IPMS-MS shall attempt to locate the entry identified by the Subject IPM field by searching entries of the Delivery and Delivery-log entry-classes. If such an entry is found, the sequence-number of the submitted IPN is added to the subject IPM's AC Submitted IPNs attribute. Conversely, the sequence-number of the subject IPM is recorded in the AC Subject IPM attribute of the present entry. In addition, the subject IPM's AC Submitted IPN Status attribute is given the value *ipm-discarded*, *ipm-auto-forwarded*, *an-sent* or *rn-sent* as appropriate. If the IPN was submitted as the result of an IPMS-MS auto-action, an MS-originated general-attribute is attached to the entry and assigned the value *true*.

d) If the submitted message contains an IPM, the IPMS-MS shall attach the following attributes to the entries created in the Submission and Submission-log entry-classes:

    1) AC IPM Recipients shall contain one value for each distinct recipient specified in the Primary, Copy, Blind Copy, and Circulation List Recipients fields;

    2) AC Correlated Delivered Replies shall contain the same number of values, each of which shall indicate that no reply has been received from the corresponding member of AC IPM Recipients;

    3) AC Delivered Replies Summary shall contain the same number of values, each of which shall indicate whether a reply was requested of the corresponding member of AC IPM Recipients;

    4) AC Correlated Delivered IPNs shall contain the same number of values, each of which shall indicate that no IPN has been received from the corresponding member of AC IPM Recipients;

    5) AC Delivered IPN Summary shall contain the same number of values, each of which shall indicate whether an RN, NRN, AN (if neither RN or NRN is present), or no notification was requested of the corresponding member of AC IPM Recipients.

e) If the submitted message contains an IPM whose Replied-to IPM heading field is present, the IPMS-MS shall attempt to locate the entry identified by the Replied-to IPM field by searching main-entries of the Delivery and Delivery-log entry-classes. If this entry is found (the replied-to entry), its sequence-number is recorded in the AC Replied-to IPM attribute of the present entry. Similarly, the replied-to entry has its AC Replying IPMs attribute updated to reference the present entry. In addition, that entry's AC Submitted Reply Status attribute is given the value *reply-sent*.

f) If the submission-options parameter inspected in item (a) above identifies one or more IPMs to be forwarded by the submitted message, then for each distinct IPM entry indicated (as a stored-entry or stored-content) the IPMS-MS shall add the sequence-number of the submitted (forwarding) entry to those entries' AC Forwarding IPMs attribute. In addition, the AC Forwarded IPMs attribute of each of the submitted message's child entries, corresponding to one or more stored IPM entries is updated to record the sequence-numbers of these forwarded IPMs.

    If the submitted message contains one or more forwarded IPMs, either as Message or Forwarded Content body parts, not identified in submission-options, the IPMS-MS attempts to locate the stored IPM entries by searching main-entries of all the entry-classes, except the Draft and Auto-action-log entry-classes, and matching on IPM identifier. For each forwarded IPM, all matching stored IPM entries found have their AC Forwarding IPMs attribute updated to record the sequence-number of the submitted message entry. In addition, the AC Forwarded IPMs attribute of the child-entry of the submitted message that corresponds to the matching stored IPM entries is updated to record the sequence-number of each stored IPM entry.

g) If the submitted message contains an IPM whose Related IPMs heading field is present, the IPMS-MS shall attempt to locate all the entries identified by each subfield of the Related IPMs field by searching main-entries of all the entry-classes except the Draft and Auto-action-log entry-classes. If any such entries are found, each has its AC Relating IPMs attribute updated to record the sequence-number of the submitted message entry. In addition, the AC Related IPMs attribute of the submitted message entry is updated to record the sequence-numbers of the related IPMs.

h) If the submitted message contains an IPM whose Obsoleted IPMs heading field is present, the IPMS-MS shall attempt to locate all the entries identified by each subfield of the Obsoleted IPMs field by searching main-entries of the Stored-message and Message-log entry-classes. If any such entries are found, each has its AC Obsoleting IPMs attribute updated to record the sequence-number of the submitted message entry. In addition, the AC Obsoleted IPMs attribute of the submitted message entry is updated to record the sequence-numbers of the obsoleted IPMs.

i) If the submitted message contains an IPM, and the submission-options parameter of the MS-message-submission argument contains an originator-body-part-encryption-token parameter, then the IPMS-MS shall create a body-part-encryption-token attribute in the Submission and Submission-log entry-classes containing that value.

j) If the submitted message contains an IPM, and the submission-options parameter of the MS-message-submission argument contains an originator-forwarded-content-token parameter, then the IPMS-MS shall create a forwarded-content-token attribute in the Submission and Submission-log entry-classes containing that value.

19.9.3 Additional Procedures for Fetch

If the IPM auto-acknowledgement auto-action is subscribed to and the suspend-auto-acknowledgement bind-extensions parameter was not present in the MS-Bind-argument which established the present abstract-association, the IPMS-MS performs the following action:

a) On the first occasion when the performance of Fetch causes the retrieval-status of an IPM to change to *processed*, the IPMS shall construct an RN if, and only if, one is requested by means of the Notification-requests component of the IPM's subject recipient specifier. (If Modify is supported to set retrieval-status back to *listed*, then a subsequent Fetch may cause the IPM to become *processed* for a second time; no RN is generated in this case). The IPMS-MS draws the Suppl Receipt Info field from the auto-acknowledge-suppl-receipt-info field of IPM-auto-acknowledgement-registration-parameter if present, and constructs other RN fields as specified in 18.5.2.1.

NOTE – The AC Submitted IPN Status attribute, if supported, may be used to ensure that only a single notification is ever sent.

b) MS-message-submission is invoked using the submission-options parameter registered for the IPM auto-acknowledgement auto-action, and the procedures defined in 19.9.2 are followed.

c) The performance of the IPM auto-acknowledgement auto-action shall cause the creation of an entry in the Auto-action-log entry-class, if subscribed to. If the performance causes an auto-action-error, the IPMS-MS shall attach an auto-action-error attribute indicating the nature of the error to the Auto-action-log entry, and shall set the auto-action-error-indication, which is reported to the IPMS-MS-user when the next abstract-association is established.

19.9.4 Additional Procedures for Delete and Auto-delete

When performing the Delete abstract-operation or the Auto-delete auto-action, the IPMS-MS shall generate an NRN if the entry contains a delivered IPM whose retrieval-status is *listed*, and an NRN was requested of this user by means of the Notification-requests component of the subject recipient specifier, as specified in 19.4. In the case of the delete abstract-operation, an NRN is not generated if prevent-nrn-generation is specified in the delete-extensions parameter of the Delete abstract-operation which deletes the IPM (see 19.5.6).

MS-message-submission is invoked with the submission-options parameter drawn from general submission-defaults (as registered by means of Register-MS), and the procedures defined in 19.9.2 are followed.

19.9.5 Auto-discard of expired IPMs

If the IPM auto-discard auto-action is subscribed to and the user has registered at least one IPM auto-discard auto-action whose registration-parameter contains an auto-discard-expired-IPMs value of *true*, the IPMS-MS performs the following actions:

a) The IPMS-MS shall identify each entry in the Delivered-message entry-class with an expiry-time attribute containing a date and time which has passed. If any such entries are found, only those which satisfy the IPM auto-discard Filter shall be considered further.

b) If any entries selected in step (a) have a retrieval-status of *new* or *listed* then the IPMS-MS shall construct an NRN as specified in 18.5.1.2 if, and only if, one is requested by means of the Notification-requests component of the IPM's subject recipient specifier.

c) MS-message-submission is invoked using the submission-options parameter registered for the IPM auto-discard auto-action, and the procedures defined in 19.9.2 are followed.

d) Each of the entries selected in step (a) shall be deleted by the IPMS-MS. If the Delivery-log entry-class is subscribed to then an IPM-auto-discarded attribute is added to the corresponding Delivery-log entry and assigned the value *true*.

NOTE – If the IPMS-MS is able to determine that the delivery of an IPM, now auto-discarded, caused an alert condition that is still outstanding, and would not be in effect but for that delivered IPM, it may set the alert-indication to *false*; see 7.1.2 of ITU-T Rec. X.413 | ISO/IEC 10021-5.

e) The performance of the IPM auto-discard auto-action shall cause the creation of an entry in the Auto-action-log entry-class, if subscribed to. If the performance causes an auto-action-error, the IPMS-MS shall attach an auto-action-error attribute indicating the nature of the error to the Auto-action-log entry, and shall set the auto-action-error-indication, which is reported to the IPMS-MS-user when an abstract-association is next established.

20 Message Contents

As has already been seen, various secondary objects (e.g., UAs) have occasion to convey the information objects of section two 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 two in accordance with the Basic Encoding Rules of ITU-T Rec. X.690 | ISO/IEC 8825-1, and the additional encoding rules specified in 7.4.12.7.

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 Extended 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 CCITT Recommendation X.411 (1984).

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

v) The Parameters and Data elements of any Message body part (of an IPM) satisfy 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 CCITT Recommendation X.420 (1984) (as clarified by Version 6 of the CCITT *1984 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 Specification and systems conforming (only) to CCITT Recommendation X.420 (1984).

NOTE 3 – If conversion is performed on a message of content type 2 which results in a message containing an Extended body part, the content type will change to 22.

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 two (a choice of an IPM or an IPN) when the Basic Encoding Rules of ITU-T Rec. X.690 | ISO/IEC 8825-1 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 encoded information types (EITs), see 8.5.6 of ITU-T Rec. X.411|ISO/IEC 10021-4, and non-basic parameters (NBPs) of the message as follows.

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

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

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

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

c) *Standard body part*: The EITs (if any) and NBPs (if any) of an individual Standard body part shall depend upon that body part type as specified in Table 8. A body part type for which the table indicates that the EITs are *unspecified* contributes no EITs to the IPM.

d) *Application-specific body part*: If the definition of an application-specific body part specifies one or more extended EITs, then those appropriate to this instance of the application-specific body part shall be specified. Otherwise, the *undefined* EIT shall be indicated. In either case, no NBPs shall be specified.

e) *Encrypted body part*: The effect of an Encrypted body part upon the EITs and NBPs to be specified may be the subject of future standardization.

Table 8 – Interpersonal Messaging EITs and NBPs

Standard 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	unspecified	-
Message	see 20.4 b)	see 20.4 b)
Mixed-mode	Mixed-mode	G4 Class 1/Mixed-mode
Bilaterally Defined	Undefined	-
Nationally Defined	Undefined	-
General Text	see 7.4.11	-
File Transfer	see 7.4.12.8	-
Voice	see 7.4.13	-
Report	unspecified	-
Notification	unspecified	-
Forwarded Content	see 7.4.16	-

21 Port Realization

How an MS or the MTS concretely realizes the secondary ports it supplies is specified in ITU-T 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 Specification.

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 CCITT 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 Specification 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 Specification, 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 Specification, 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 separate statements shall be made concerning conformance upon origination and conformance upon reception:

a) The heading fields and heading extensions for which conformance is claimed.

b) The standard and application-specific body part types for which conformance is claimed.

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 conformance is claimed.

d) In the case of an IPMS-MS, or an IPMS UA accessing an IPMS-MS, those IPMS-specific auto-actions and matching-rules for which conformance is claimed.

In addition, the implementor of a TLMA or AU shall state whether conformance is claimed 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 standard and application-specific 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 Table 3.

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 ITU-T Rec. X.419 | ISO/IEC 10021-6.

f) An IPMS UA, IPMS-MS, TLMA, or AU which claims support upon reception for any Standard body part type for which both basic and extended representations are defined shall support the reception of both the basic and the extended representation of that body part type.

g) An IPMS UA, IPMS-MS, TLMA, or AU which claims support upon reception for an Application-specific body part type consisting of octet-aligned data, or which claims support upon reception for the File Transfer body part's application-reference encoded as an Object Identifier, shall support both the reception of that application-specific octet-aligned data encoded in the Application-specific body part and the reception of it encoded in the File Transfer body part.

h) 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 clause 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

General IPMS Extensions

(This annex forms an integral part of this Recommendation | International Standard)

This annex defines general extensions for Interpersonal Messaging. Security extensions are defined in Annex B.

A.1 Heading Extensions

This section defines all (presently defined) general heading extensions. Heading extensions are conveyed in the extensions heading field of the IPM (see 7.2.17).

A.1.1 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.

```
incomplete-copy IPMS-EXTENSION ::= {VALUE IncompleteCopy,
                                    IDENTIFIED BY id-hex-incomplete-copy}

IncompleteCopy ::= NULL
```

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

A.1.2 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 clause 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, IDENTIFIED BY 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.1.3 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, or auto-replied.

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,
                                   IDENTIFIED BY id-hex-auto-submitted}

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

The auto-forwarded heading field may also indicate that the message has been submitted without human intervention. The absence of both this heading extension and the auto-forwarded heading field indicates that no information is available as to whether the message-submission involved human control.

A.1.4 Body Part Signature

The **Body Part Signature** heading extension may contain a digital signature for each body part in the IPM to enable the recipient to verify that the body part has not been modified.

The encoding of the body part itself is unchanged in order that it may be understood by all recipients, irrespective of ability to verify these signatures.

```
body-part-signatures IPMS-EXTENSION ::= {VALUE BodyPartSignatures,
                                         IDENTIFIED BY id-hex-body-part-signatures }
```

```
BodyPartSignatures ::= SET OF SET {
       body-part-number                BodyPartNumber,
       body-part-signature             BodyPartSignature,
       originator-certificate-selector CertificateAssertion OPTIONAL,
       originator-certificates         [0] ExtendedCertificates OPTIONAL,
       ... }

BodyPartNumber ::= INTEGER (1..MAX)

BodyPartSignature ::= SIGNATURE { SEQUENCE {
       signature-algorithm-identifier  AlgorithmIdentifier,
       body-part                       BodyPart,
       body-part-security-label        SecurityLabel OPTIONAL } }
```

The Body part signatures comprises the following components for each signed body part:

a) **Body-part-number** (M): Identifies one of the body parts in this IPM, numbered starting at '1'.

b) **Body-part-signature** (M): Identifies an algorithm and contains the signature obtained by applying that algorithm to the Body Part together with its *Body-part-security-label* (see A.1.5) and the algorithm-identifier; the algorithm-identifier specifies five constituent parts:

   i) the ASN.1 encoding rule (CER or DER) to be applied to the body part before hashing;

   ii) the hash function;

   iii) whether or not the hash value is to be encoded within an ASN.1 Bit String prior to encryption;

   iv) the algorithm used to protect the hash value (e.g. an asymmetric encryption algorithm); and

   v) any parameters of the algorithm such as any necessary keys, initialisation values, and padding instructions.

   NOTE – There is no requirement to transmit the CER or DER encoding of the body part, but simply to apply CER or DER encoding before computing the signature.

   Where body-part-number identifies an encrypted body part, the signature applies to the body part before encryption.

c) **Originator-certificate-selector** (C): If the originator has more than one public key for the signature algorithm (and therefore more than one certificate with the same algorithm-identifier), then either the originator-certificate-selector shall be present containing components of certificate-assertion (see 12.7.2 of ITU-T Rec. X.509 | ISO/IEC 9594-8) appropriate to enable the recipient to determine which one of the originator's certificates (and thus which one of the originator's public keys for this algorithm) is applicable, or else precisely one certificate for this algorithm shall be present in originator-certificates.

d) **Originator-certificates** (C): This may be used to convey a verified copy of the public-asymmetric-encryption-key of the originator who signed the body part (i.e. a Certificate), or the name of a Directory entry containing the originator's certificate, or multiple certificates (or Directory names) where the public-asymmetric-encryption-key is verified with different certification paths or by different certification authorities. This component shall be present if the originator who signed the body part is different from the originator of the IPM, but may also be present otherwise. Where the originator certificate is not included within the body part signatures, then the originator certificate which may be present on the envelope shall be used. If no certificates are present, it is assumed that another method (e.g. use of the Directory) is to be used to obtain a verified copy of the originator's public key.

A body part may be signed more than once if it is required to use different algorithms for some recipients, or to include more than one signatory.

A.1.5 IPM Security Label

The **IPM Security Label** heading extension provides the capability to label a complete message content, the message heading, and individual message body parts.

```
ipm-security-label IPMS-EXTENSION ::= {
       VALUE               IPMSecurityLabel,
       IDENTIFIED BY       id-hex-ipm-security-label }

IPMSecurityLabel ::= SEQUENCE{
       content-security-label      [0] SecurityLabel,
       heading-security-label      [1] SecurityLabel OPTIONAL,
       body-part-security-labels   [2] SEQUENCE OF BodyPartSecurityLabel OPTIONAL }

BodyPartSecurityLabel  ::= CHOICE {
       body-part-unlabelled        [0] NULL,
       body-part-security-label    [1] SecurityLabel }
```

The IPM Security Label comprises the following components:

a) **Content-security-label** (M): Identifies the aggregate security classification of all the security labels contained in this heading extension. The relationship between this aggregate security label and the other labels in this heading extension are subject to the local security policy. The content-security-label component and the message-security-label (in the envelope) both provide an indication of the security classification of the message content. However, depending on local architecture and policy, the security policy cited by the two elements may differ. Local policy may also affect the choice of whether to use the content-security-label, the message-security-label, or both as part of the security architecture. One factor influencing this choice is that the content-security-label is within the message content, and will therefore be affected by any encryption of the content; whereas the message-security-label is in the envelope and is therefore independent of content encryption.

b) **Heading-security-label** (O): Identifies the security classification of the information collectively conveyed in the heading of the IPM.

c) **Body-part-security-labels** (O): Identifies the security classification of the information conveyed by individual body parts of the IPM. Labels may be included for any subset of the body parts included in the IPM. One value of body-part-security-label shall be present for each body part present in the IPM, and shall appear in the same order in body-part-security-labels as the body part appears within the body of the IPM. The body-part-security-label comprises either:

   i) *Body-part-unlabelled* (C): Indicates that the body part is not labelled.

   ii) *Body-part-security-label* (C): Identifies the security label (see 8.5.9 of ITU-T Rec. X.411 | ISO/IEC 10021-4) pertaining to the body part.

A.1.6 Authorization Time

The **Authorization Time** heading extension identifies the date and time at which a message was formally authorized by the authorizing users. Depending upon local requirements, this date and time stamp may vary from the date and time when the message was submitted to the MTS. The authorization time can be used to augment the authorizing users heading field (see 7.2.3) to provide additional information about the authorizing event. Either or both of the authorization time or authorizing users may be present according to local policy.

```
authorization-time IPMS-EXTENSION ::= {
       VALUE                AuthorizationTime,
       IDENTIFIED BY        id-hex-authorization-time }

AuthorizationTime ::= GeneralizedTime
```

A.1.7 Circulation List Recipients

The **Circulation List Recipients** heading extension identifies the two or more users to whom the authorizing users request that the IPM be distributed sequentially. It also identifies the responses the authorizing users ask of each of those users. It comprises a Sequence of sub-fields, each a **Circulation Member**, one for each circulation list recipient.

Each circulation member includes an indication of whether that user has already received the IPM, and optionally the date and time when the IPM was received, and optionally a digital signature. In this context, users that have received the message are said to be "checked" in the circulation list.

```
circulation-list-recipients IPMS-EXTENSION       ::= {
       VALUE                CirculationList,
       IDENTIFIED BY        id-hex-circulation-list-recipients }

CirculationList ::= SEQUENCE (SIZE(2..ub-circulation-list-members))
                             OF CirculationMember

CirculationMember ::= SET {
       circulation-recipient RecipientSpecifier (WITH COMPONENTS {
                                   ...,
                                   recipient (WITH COMPONENTS {
                                        ...,
                                        formal-name PRESENT } ) } ),
       checked                     Checkmark OPTIONAL }

Checkmark ::= CHOICE {
       simple               NULL,
       timestamped          CirculationTime,
       signed               CirculationSignature }

CirculationTime ::= GeneralizedTime

CirculationSignature ::= SIGNED { SEQUENCE {
       algorithm-identifier CirculationSignatureAlgorithmIdentifier,
       this-IPM             ThisIPMField,
       timestamp            CirculationTime } }
```

```
CirculationSignatureAlgorithmIdentifier ::= AlgorithmIdentifier
```

A **circulation list** is a set of circulation members, each of which has the following components:

a) **Circulation-recipient** (M): Identifies a member of the circulation list and any information (i.e., notification requests, reply request, and recipient extensions) associated with that member. The formal-name component of the OR-descriptor shall be present. If the recipient is a DL then this should be the last circulation-recipient in the circulation list, as the IPM will not be sent to members of a circulation list subsequent to a DL.

b) **Checked** (C): Identifies whether or not this member has received the IPM as a result of the circulation list.

If a member of the circulation list has not yet received the IPM, then the checked component shall be absent. If a member has received the IPM, then the checked component shall take one of the following values:

i) *simple*: a simple assertion that the circulation recipient has received the IPM;

ii) *timestamped*: indicates the date and time at which the IPM was accessed by the circulation recipient;

iii) *signed*: includes the IPM Identifier, the date and time (as above), and a digital signature of these by the circulation recipient.

The authorizing user should compose the circulation list to indicate the preferred order of circulation. Any notifications requested are returned to the originator. These may be used by the originator to perform correlation and tracking of the progress of the circulation. If the authorizing user elects to receive the completed circulation list at the conclusion of circulation, then the originator should be included as the last member of the circulation list. If the authorizing user desires that replies to the message be received by all members of the circulation list, then this should be reflected in the selection of reply recipients. Recipients other than the circulation list may also be specified at the discretion of the originator.

On reception, the UA examines the circulation list recipients as part of the check of recipient fields. If the recipient is included in the circulation list, the UA determines the receipt requests, reply requests and recipient extensions applicable to that recipient as specified in clause 8. Notifications will be returned to the originator of the circulation list. Replies should be directed to the reply recipients specified in the heading by the authorizing users.

The circulation list should be updated by the UA to add the checked component to it's own entry. It may optionally include additional recipients in the list. Existing elements of the list should not be deleted, and the UA shall ensure that the originator's last entry remains last in the sequence. Other heading fields of the IPM should not be modified. In particular it is vital that the this-IPM field (see 7.2.1) and originator field (see 7.2.2) be preserved intact.

If, after updating, all the members of the list are checked, or if the IPM has been delivered as the result of DL Expansion, then the IPM requires no further sequential circulation. Otherwise, the UA should select the next list member that does not contain the checked component for the next circulation step. After processing and review is complete, the UA should resubmit the modified IPM to the MTS. The submission envelope should address only the next circulation list member.

The UA of each circulation member may append additional body parts (containing annotations or other comments) at the discretion of the user.

UAs that originate circulation list-recipients may optionally generate in addition a single instance of a recipient specifier to include the circulation list indicator recipient extension (see A.2.1) in any one of the other recipient heading fields (primary recipients, copy recipients, or blind copy recipients) for backward compatibility purposes. UAs that support circulation list-recipients should also support the circulation list identifier on reception.

A.1.8 Distribution Codes

The **Distribution Codes** heading extension identifies information to support distribution of the IPM either within the MHS (e.g., auto-forwarding) or external to the MHS (e.g., hard copy distribution). A specific definition of distribution code values and semantics should be mutually supported by the originator and the recipient. The value of this field may be used within filters which select specific auto-actions (e.g., specific auto-forward and auto-alert registrations) and within the attributes in an Alert operation.

```
distribution-codes IPMS-EXTENSION ::= {
       VALUE               DistributionCodes,
       IDENTIFIED BY       id-hex-distribution-codes }

DistributionCodes ::= SEQUENCE (SIZE (1..ub-distribution-codes)) OF DistributionCode
```

```
DistributionCode ::= SEQUENCE {
        oid-code              OBJECT IDENTIFIER OPTIONAL,
        alphanumeric-code     AlphaCode OPTIONAL,
        or-descriptor        ORDescriptor OPTIONAL }

AlphaCode ::= UniversalOrBMPString {ub-alpha-code-length}
```

The Distribution Code has the following components:

a) **OID-code** (O): Identifies a request from the originator for the recipient to redistribute the IPM according to previously agreed semantics. The semantics of the originator's request may be solely conveyed by this component, or may additionally depend on the values of the OR-descriptor or alphanumeric-code components.

b) **Alphanumeric-code** (O): Identifies a request from the originator for the recipient to redistribute the IPM according to previously agreed semantics. The semantics of the originator's request may be solely conveyed by this component, or may additionally depend on the values of the OR-descriptor or OID-code components.

c) **OR-descriptor** (O): Identifies a non-IPMS user whom the originator has requested be part of the post-delivery distribution process for this IPM. The values of alphanumeric-code and OID-code may qualify the distribution role requested by the originator.

At least one component of the DistributionCode shall be present.

A.1.9 Extended Subject

The **Extended Subject** heading extension identifies the subject of the IPM. The extended subject consists of a string that provides both a richer character set and a greater length than is provided by the subject heading field (see 7.2.10).

If the extended subject is supported on reception, and both the subject and the extended subject are present, then the extended subject shall be used in preference to the subject for display to the user.

```
extended-subject IPMS-EXTENSION ::= {
        VALUE                 ExtendedSubject,
        IDENTIFIED BY         id-hex-extended-subject }

ExtendedSubject ::= UniversalOrBMPString {ub-extended-subject-length}
```

On origination, if the text of the subject can be accommodated without loss within the constrained length and character repertoire of the subject heading field (see 7.2.10) then that field should be generated in preference to extended subject. When extended subject is generated, it is recommended that if possible the subject heading field is also generated (for recipient UAs which may not support extended subject) containing the same text converted and truncated as necessary.

A.1.10 Information Category

The **Information Category** heading extension identifies the character of the information contained in the IPM. It can convey a registered identifier for each particular type, or free form information describing the nature of the communication. Recipient UAs may use the information conveyed by this extension to affect the presentation of messages to the recipient, or to affect any other local processing functions. A specific definition of information category values and semantics should be mutually supported by the originator and the recipient. Examples of possible information category values include: draft message, press release, contractual commitment, policy statement.

```
information-category IPMS-EXTENSION      ::= {
        VALUE                 InformationCategories,
        IDENTIFIED BY         id-hex-information-category }

InformationCategories ::= SEQUENCE (SIZE (1..ub-information-categories))
        OF InformationCategory

InformationCategory ::= SEQUENCE {
        reference         [0] OBJECT IDENTIFIER OPTIONAL,
        description       [1] DescriptionString OPTIONAL }

DescriptionString ::= UniversalOrBMPString {ub-information-category-length}
```

The Information Category has the following components:

a) **Reference** (O): Depending on the mutually agreed semantics, this component may identify either:

   i) the category of message; or

   ii) the context in which the description component should be interpreted to determine the category of the message.

   This component may be used either alone or in combination with the description component.

b) **Description** (O): identifies the category of message based on mutually agreed semantics.

This component may be used either alone or in combination with the reference component.

At least one component of the Information Category shall be present. If both components are used, then the category shall be determined primarily by the reference component, as qualified by description component.

A.1.11 Manual Handling Instructions

The **Manual Handling Instructions** heading extension identifies instructions for manual handling of the IPM following its delivery. The extension can convey instructions consisting of free form text. Examples of manual handling instructions include special recipient handling requests (e.g., "Please pass to ...", "Please DO NOT pass to ..."), and instructions for how to process body data.

NOTE – Instructions indicated by this heading extension may apply either to the IPM as a whole or to specific components of the IPM (e.g., specific body parts). Where necessary the content of the instructions should indicate the scope of the instructions or the part(s) of the IPM to which the instruction applies.

```
manual-handling-instructions IPMS-EXTENSION ::= {
       VALUE                 ManualHandlingInstructions,
       IDENTIFIED BY         id-hex-manual-handling-instructions }

ManualHandlingInstructions ::= SEQUENCE (SIZE (1..ub-manual-handling-instructions))
                                        OF ManualHandlingInstruction

ManualHandlingInstruction ::=
       UniversalOrBMPString {ub-manual-handling-instruction-length}
```

A.1.12 Originator's Reference

The **Originator's Reference** heading extension identifies a reference value that is chosen by the authorizing users. The originator's reference may be used within the organization of the originator as an internal reference. Examples of possible originator's references include: file number, claim number, legal case number. This information may be used by the recipient in later communications with the originator, possibly outside the MHS, concerning a particular IPM.

```
originators-reference IPMS-EXTENSION ::= {
       VALUE                 OriginatorsReference,
       IDENTIFIED BY         id-hex-originators-reference }

OriginatorsReference ::= UniversalOrBMPString {ub-originators-reference-length}
```

A.1.13 Precedence Policy Identifier

The **Precedence Policy Identifier** heading extension identifies the semantics of the precedence values assigned to the recipients by the authorizing users. The precedence policy identifier is intended to be used only in conjunction with the precedence recipient extension defined in A.2.2, and applies to all instances of the precedence recipient extension. If the precedence policy identifier is received in the absence of any precedence values, then the precedence policy identifier shall be ignored.

```
precedence-policy-identifier IPMS-EXTENSION ::= {
       VALUE                 PrecedencePolicyIdentifier,
       IDENTIFIED BY         id-hex-precedence-policy-id }

PrecedencePolicyIdentifier ::= OBJECT IDENTIFIER
```

A.2 Recipient Extensions

This section defines all general recipient extensions. Recipient extensions are conveyed in the recipient-extensions field of the recipient specifier (see 7.1.2).

```
RecipientExtensions IPMS-EXTENSION ::= {
       circulation-list-indicator |
       precedence |
       recipient-security-request |
       PrivateIPMSExtensions, ... }
```

A.2.1 Circulation List Indicator

The **Circulation List Indicator** recipient extension identifies a recipient specifier that is associated with the circulation list-recipients heading extension. Use of this extension provides graceful degradation of the circulation list-recipients function for UAs that do not support the circulation list-recipients.

```
circulation-list-indicator  IPMS-EXTENSION ::= {
       VALUE                 NULL,
       IDENTIFIED BY         id-rex-circulation-list-indicator }
```

UAs that originate a circulation list-recipients may optionally generate a single instance of recipient specifier in any one of the recipient heading fields (primary recipients, copy recipients, or blind copy recipients) for backward compatibility purposes. Such a recipient specifier should include the circulation list indicator recipient extension to allow it to be easily identified by recipient UAs that support circulation list. UAs that support circulation list-recipients should also support the circulation list identifier on reception.

Inclusion of this extension in an instance of the recipient specifier constrains the allowed values of the recipient component of that recipient specifier. When this extension is used, the recipient shall be specified using only the value "CIRCULATION LIST" in the free-form-name component of the OR-descriptor. The formal-name and telephone-number components of the OR-descriptor shall be absent. A maximum of one instance of this extension is allowed per IPM.

On reception, a UA or MS that supports circulation list-recipients shall expunge a recipient-specifier if it contains circulation list identifier.

A.2.2 Precedence

The **Precedence** recipient extension identifies the precedence level that the authorizing users attribute to the IPM for each recipient. The precedence level provides an indication of the perceived importance, or relevance, of the IPM to each recipient. The semantics of values that appear in this field are entirely dependent on the value conveyed in the precedence policy identifier extension. If precedence values are present in the absence of the precedence policy identifier, then the precedence values shall be ignored. This extension shall only be originated in conjunction with the precedence policy identifier extension.

```
precedence IPMS-EXTENSION ::= {
        VALUE                   Precedence,
        IDENTIFIED BY           id-rex-precedence }

Precedence ::= INTEGER (0..ub-precedence)
```

The precedence recipient extension affects the priority attribute in the submission envelope as described in 18.2.2(a) bullet (iv).

A.3 Notification Extensions

This section defines all general notification extensions. Notification extensions are conveyed in the notification extensions (see 8.1.5), rn extensions (see 8.3.4), nrn extensions (see 8.2.5), or other notification type (see 8.4) fields of the IPN.

No general notification extensions are defined in this edition of ITU-T Rec. X.420 | ISO/IEC 10021-7.

Annex B

IPMS Security Extensions

(This Annex forms an integral part of this Recommendation | International Standard)

This annex defines security extensions for Interpersonal Messaging. It specifies an optional request which may be included in an IPM, the response which may be included in the resulting IPN, and additional procedures for generating this response. It uses the IPMS Extension of 7.2.17.

B.1 Recipient Security Request

The **Recipient Security Request** is an IPMS extension which may be present in the recipient-extensions field of a recipient specifier. The Recipient Security Request indicates which security function is requested to be applied to an IPN (RN or NRN) from this recipient (whether proof or non-repudiation), and which security function is to be applied to the IPM by this recipient (whether proof or non-repudiation) on receipt of the IPM.

```
recipient-security-request IPMS-EXTENSION ::= {
       VALUE        RecipientSecurityRequest,
       IDENTIFIED BYid-sec-ipm-security-request}

RecipientSecurityRequest ::= BIT STRING {
       content-non-repudiation (0),
       content-proof (1),
       ipn-non-repudiation (2),
       ipn-proof (3)}
```

The Recipient Security Request shall be present only if the notification-requests in the recipient specifier has the value rn or nrn (see 7.1.2), and thus requires that the recipient specifier contains a formal-name (see 7.1.3).

The Recipient Security Request may have the following values:

a) *content-non-repudiation*: An IPN generated in the circumstances prescribed in clause 8 is requested to contain verified security elements possessing non-repudiation properties from the message whose content is the subject IPM.

The security elements to be verified by the recipient shall depend on the security arguments present in the envelope of this message and on the security policy in force.

If the recipient is unable to verify the security arguments, the IPN may contain either the content of the message or an appropriate security diagnostic code.

b) *content-proof*: An IPN generated in the circumstances prescribed in clause 8 is requested to contain verified security elements from the message whose content is the subject IPM.

The security elements to be verified by the recipient shall depend on the security arguments present in the envelope of this message and on the security policy in force.

If the recipient is unable to verify the security arguments, the IPN may contain either the content of the message or an appropriate security diagnostic code.

c) *ipn-non-repudiation*: An IPN generated in the circumstances prescribed in clause 8 is requested to be signed with non-repudiation properties on submission of the IPN to the MTS.

d) *ipn-proof*: An IPN generated in the circumstances prescribed in clause 8 is requested to be signed on submission of the IPN to the MTS.

The absence of the Recipient Security Request implies that no IPM recipient security request is made.

Support of the Recipient Security Request does not imply support of *IPN Security Response* (see B.2). If the recipient UA does not support the *IPN Security Response* then it ignores the Recipient Security Request. If the recipient UA supports the *IPN Security Response* but is unable to generate the required response then it shall generate an appropriate *Security Diagnostic Code* (see B.3).

The Recipient Security Request does not in itself require any security arguments to be present in the subject message on submission. However, the requested notification may contain the *original-content* of the subject message unless one of the following security elements is generated on message submission:

Content-integrity-check,

Message-token (including at least one *content-integrity-check*),

Message-origin-authentication-check.

These security arguments are defined in 8.2.1.1.1.28, 8.2.1.1.1.29 and 8.2.1.1.1.26 of ITU-T Rec. X.411 | ISO/IEC 10021-4 respectively.

NOTE – When requesting content-non-repudiation, it is recommended that the security arguments applied to the message on submission have non-repudiation properties.

B.2 IPN Security Response

The **IPN Security Response** is an IPMS extension which may be present in the notification extensions field of an IPN. It shall be present only if the subject recipient specifier contains a Recipient Security Request. The IPN Security Response may contain the *subject message* content, or security elements of the *subject message*, or a security diagnostic code.

The **subject message** contains the subject IPM; it is the MessageDeliveryEnvelope and Content as defined in ITU-T Rec. X.411 | ISO/IEC 10021-4.

```
ipn-security-response IPMS-EXTENSION ::= {
       VALUE        IpnSecurityResponse,
       IDENTIFIED BYid-sec-security-common-fields}

IpnSecurityResponse ::= SET {
       content-or-arguments CHOICE {
           original-content OriginalContent,
           original-security-arguments SET {
             original-content-integrity-check
                     [0] OriginalContentIntegrityCheck OPTIONAL,
             original-message-origin-authentication-check
                     [1] OriginalMessageOriginAuthenticationCheck OPTIONAL,
             original-message-token [2] OriginalMessageToken OPTIONAL}},
       security-diagnostic-code SecurityDiagnosticCode OPTIONAL }

OriginalContent ::= Content

OriginalContentIntegrityCheck ::= ContentIntegrityCheck

OriginalMessageOriginAuthenticationCheck ::= MessageOriginAuthenticationCheck

OriginalMessageToken ::= MessageToken
```

The setting of these fields and their use will be subject to the originator's request in the Recipient Security Request, the security elements present in the subject message and the security policy in force (in the circumstances prescribed in B.3).

The IPN Security Response may have one or more of the following values:

a) *original-content*: the content of the subject message.

b) *original-content-integrity-check*: the content-integrity-check of the subject message.

c) *original-message-origin-authentication-check*: the message-origin-authentication-check of the subject message.

d) *original-message-token*: the message-token of the subject message.

NOTE – b), c) and d) above are available only if the subject message contains the relevant security arguments in its delivery envelope.

e) *security-diagnostic-code*: the security diagnostic code (see B.3).

Support of the IPN Security Response requires support for reception of the Recipient Security Request.

Unless the IPN Security Response is a security-diagnostic-code, at least one of the following security elements shall be generated on message submission of the IPN in reply to the subject message:

Content-integrity-check,

Message-token (including at least one *content-integrity-check*),

Message-origin-authentication-check.

These security arguments are defined in 8.2.1.1.1.28, 8.2.1.1.1.29 and 8.2.1.1.1.26 of ITU-T Rec. X.411 | ISO/IEC 10021-4 respectively.

B.3 Security Diagnostic Code

A **Security Diagnostic Code** may be generated if a UA cannot support a Recipient Security Request or a security failure is detected.

```
SecurityDiagnosticCode ::= INTEGER {
       integrity-failure-on-subject-message (0),
       integrity-failure-on-forwarded-message (1),
       moac-failure-on-subject-message (2),
       unsupported-security-policy (3),
       unsupported-algorithm-identifier (4),
       decryption-failed (5),
       token-error (6),
       unable-to-sign-notification (7),
       unable-to-sign-message-receipt (8),
       authentication-failure-on-subject-message (9),
       security-context-failure-message (10),
       message-sequence-failure (11),
       message-security-labelling-failure (12),
       repudiation-failure-of-message (13),
       failure-of-proof-of-message (14),
       signature-key-unobtainable (15),
       decryption-key-unobtainable (16),
       key-failure (17),
       unsupported-request-for-security-service (18),
       inconsistent-request-for-security-service (19),
       ipn-non-repudiation-provided-instead-of-content-proof (20),
       token-decryption-failed (21),
       double-enveloping-message-restoring-failure (22),
       unauthorised-dl-member (23),
       reception-security-failure (24),
       unsuitable-alternate-recipient (25),
       security-services-refusal (26),
       unauthorised-recipient (27),
       unknown-certification-authority-name (28),
       unknown-dl-name (29),
       unknown-originator-name (30),
       unknown-recipient-name (31),
       security-policy-violation (32) }
```

The Security Diagnostic Code may have one of the following values:

a) *integrity-failure-on-subject-message*: validation of the content-integrity-check argument of the subject message failed, the contents of the message received could not be validated.

b) *integrity-failure-on-forwarded-message*: validation of the content-integrity-check argument of a subject message which has been forwarded has failed, the contents of the message-body part received could not be validated.

c) *moac-failure-on-subject-message*: validation of the message-origin-authentication-check argument of the subject message failed, the original contents of the subject message received could not be validated.

d) *unsupported-security-policy*: the recipient does not support the required security policy, as identified in the message-security-label argument of the subject message.

e) *unsupported-algorithm-identifier*: the recipient does not support the algorithm identifiers used the security argument of the subject message.

f) *decryption-failed*: the recipient could not decrypt the message content.

g) *token-error*: an error has been detected with the message-token argument of the subject message.

h) *unable-to-sign-notification*: the recipient is unable to sign IPNs.

i) *unable-to-sign-message-receipt*: the recipient is unable to validate the contents or sign IPNs.

j) *authentication-failure-on-subject-message*: validation of the content-integrity-check, message-origin-authentication-check, or message-token (i.e. token signature, or any other the token data) argument of the subject message failed, the contents of the message received could not be authenticated or validated.

k) *security-context-failure-message*: the message-security-label failed the security-context.

l) *message-sequence-failure*: failure of the message-sequence-number.

m) *message-security-labelling-failure*: a fault was detected in the message-security-label in the envelope or in the message-token.

n) *repudiation-failure-of-message*: a fault was detected in the repudiation security arguments in the subject message, or the subject message content could not be validated.

o) *failure-of-proof-of-message*: a fault was detected in the proof-of security arguments in the subject message.

p) *signature-key-unobtainable*: the recipient could not obtain the required signature keys for one or more of the pieces of signed information in the subject message.

q) *decryption-key-unobtainable*: the recipient could not obtain the required decryption keys for the message token encrypted data or for content confidentiality.

r) *key-failure*: the recipient could not obtain the required keys.

s) *unsupported-request-for-security-service*: the recipient could not support the requested security services in the Recipient Security Request.

t) *inconsistent-request-for-security-service*: the recipient could not support the requested security services in the Recipient Security Request because the requests where not consistent.

u) *ipn-non-repudiation-provided-instead-of-content-proof*: the recipient has supported ipn-non-repudiation but not content-proof.

v) *token-decryption-failed*: The recipient could not decrypt the message token.

w) *double-enveloping-message-restoring-failure*: The message contained an inner envelope, but failure of security services on the outer envelope prevented the UA from extracting the inner message for subsequent processing.

x) *unauthorised-dl-member*: The UA has detected that the message has been received via a DL, yet this recipient was prohibited by the security policy from being a member of that DL.

y) *reception-security-failure*: The message could not be received due to the failure of one of the message security services.

z) *unsuitable-alternate-recipient*: The message was not able to be processed as it has been delivered to an alternate recipient and this recipient is unable to process the security functions.

aa) *security-services-refusal*: The security services cannot be supported.

ab) *unauthorised-recipient*: The recipient is not allowed to get the required decryption keys for content confidentiality. The recipient is not authorised to read the message content.

ac) *unknown-certification-authority-name*: The message cannot be processed because the certification authority named in a certificate contained within one of the security arguments is not known to the UA, or is not trusted by the UA.

ad) *unknown-dl-name*: The security policy requires the UA to perform checks on messages that have been received via DLs, and in this case one of the DLs named in the DL-expansion-history was unknown to the UA.

ae) *unknown-originator-name*: The originator MTS-user OR-name identifies a user who is not known to the receiving UA, and hence the security arguments cannot be validated.

af) *unknown-recipient-name*: The recipient MTS-user OR-name identifies a user who is not known to the receiving UA, and hence the security arguments cannot be validated.

ag) *security-policy-violation*: The security policy is violated.

B.4 Additional UA Procedures

This clause defines additional procedures for user agent operation to support Recipient Security Request and IPN Security Response.

B.4.1 Originate IPM

If the UA supports the Recipient Security Request, the UA shall perform the Originate IPM abstract operation by invoking Message Submission with the arguments indicated in 18.2.2 and the following additional arguments.

The arguments of Message Submission shall be as follows:

a) *Envelope*: When the Recipient Security Request is requested, and the security policy specifies the support of one or more of Non-Repudiation of Origin or Content Integrity or Message Origin Authentication, the following shall apply.

The UA shall sign the IPM by generating one or more of:

i) *content-integrity-check*: defined in 8.2.1.1.1.28 of ITU-T Rec. X.411 | ISO/IEC 10021-4.

ii) *content-integrity-check*: defined in 8.2.1.1.1.28 of ITU-T Rec. X.411 | ISO/IEC 10021-4 included in the message-token defined in 8.2.1.1.1.26 of ITU-T Rec. X.411 | ISO/IEC 10021-4.

iii) *message-origin-authentication-check*: defined in 8.2.1.1.1.29 of ITU-T Rec. X.411 | ISO/IEC 10021-4.

b) *Content*: The Recipient Security Request shall be absent (or all bits within it shall be zero) unless the Notification-requests contains the value rn or nrn.

B.4.2 Originate IPN

If the UA supports the IPN Security Response, the UA shall perform the Originate RN abstract operation by invoking Message Submission with the arguments indicated in 18.2.3 extended to support security as defined below, and by returning to its user the results indicated in 18.2.3.

B.4.2.1 Message Submission

The arguments of Message Submission shall be as follows:

a) *Envelope*: The UA shall generate and submit security arguments defined in 8.2.1.1.1 of ITU-T Rec. X.411 | ISO/IEC 10021-4 as required by the IPN Security Response procedures defined in B.4.2.2.

When the security policy specifies the support of one or more of the following Elements of Service, Non-Repudiation of Origin, Content Integrity or Message Origin Authentication the following shall apply. The UA shall sign the IPN by generating one or more:

i) *content-integrity-check*: defined in 8.2.1.1.1.28 of ITU-T Rec. X.411 | ISO/IEC 10021-4.

ii) *content-integrity-check*: defined in 8.2.1.1.1.28 of ITU-T Rec. X.411 | ISO/IEC 10021-4 included in the message-token defined in 8.2.1.1.1.26 of ITU-T Rec. X.411 | ISO/IEC 10021-4.

iii) *message-origin-authentication-check*: defined in 8.2.1.1.1.29 of ITU-T Rec. X.411 | ISO/IEC 10021-4.

Subject to the security policy in force the UA may generate and submit other security arguments as defined in 8.2.1.1.1 of ITU-T Rec. X.411 | ISO/IEC 10021-4.

b) *Content*: The components of the IPN Security Response shall be as required by the IPN Security Response procedures defined below.

B.4.2.2 IPN Security Response Procedures

Unless the security policy of the receiving UA requires the UA to honour the Recipient Security Request, a UA can ignore the Recipient Security Request on reception. Also, a UA is able to support the Recipient Security Request on reception without supporting the IPN Security Response. Alternatively, a security policy may dictate that the UA is able to support the Recipient Security Request on reception only when the UA supports the IPN Security Response, in which case the UA shall obey all the IPN Security Request procedures defined below.

NOTE – The security context established for the UA may be used to ensure that delivery can only be to a UA that is able to support either or both the Recipient Security Request and the IPN Security Response.

B.4.2.2.1 Precedence of Requests

If more than one value is present in Recipient Security Request and the UA supports more than one of the requests, then the following precedence rules shall apply:

a) the content-non-repudiation procedures (see B.4.2.2.2) shall be the only procedures invoked when the request is present and supported, otherwise

b) the ipn-non-repudiation procedures (see B.4.2.2.4) shall be the only procedures invoked when ipn-non-repudiation together with either or both content-proof or ipn-proof are requested and supported, otherwise

c) the content-proof procedures (see B.4.2.2.3) shall be the only procedures invoked when both content-proof and ipn-proof are requested and supported.

When both ipn-non-repudiation and content-proof are requested and supported, the UA shall in addition to the ipn-non-repudiation procedures also generate a Security Diagnostic Code with the value *ipn-non-repudiation-provided-instead-of-content-proof*.

If more than one value is present in Recipient Security Request but the UA supports only one of the requests, then the procedure for the supported request shall apply.

B.4.2.2.2 Content-non-repudiation Requested

If, in the subject IPM, the Recipient Security Request is set to *content-non-repudiation* then:

a) The UA shall validate security arguments present in the subject message as required by the security policy in force. The UA shall copy only validated arguments from the subject message to the IPN Security Response as defined in Table B.1.

If the UA cannot validate any of the security arguments in the subject message, then the UA shall generate an IPN Security Response with an appropriate Security Diagnostic Code (such as: *repudiation-failure-of-message*, *unsupported-algorithm-identifier*, *authentication-failure-on-the-subject*), see B.3.

Table B.1 – Security argument mapping

FROM: **Subject message security arguments**	**TO** **IPN Security Response arguments**
message-token original-message-token content-integrity-check message-origin-authentication-check	original-message-token original-content-integrity-check original-message-origin-authentication-check

b) If the UA supports Non-Repudiation of Origin then the UA shall submit the IPN with one or more of the following (depending on the security policy in force):

i) The security element content-integrity-check, possibly in the message-token, which possesses non-repudiation properties; or

ii) The security element message-origin-authentication-check which possesses non-repudiation properties.

If the UA does not supports Non-Repudiation of Origin the UA shall generate an IPN Security Response with a Security Diagnostic Code of *unsupported-request-for-security-service*.

c) If the UA supports Non-Repudiation of Origin but the subject message has not included any of the following security arguments:

i) message-token;

ii) content-integrity-check; or

iii) message-origin-authentication-check.

Then the content of the subject message shall be copied into the original-content of the IPN Security Response or the UA shall generate an IPN Security Response with an appropriate Security Diagnostic Code (such as: *authentication-failure-on-subject-message*, *repudiation-failure-of-message*), see B.3.

NOTE – The choice of returning the content or Security Diagnostic Code is a local issue and can be a matter for the User or the implementation to decide.

Then the UA shall submit the IPN with one or more of the following (depending on the security policy in force):

i) the security element content-integrity-check, possibly in the message-token, which possesses non-repudiation properties.

ii) Or the security element message-origin-authentication-check which possesses non-repudiation properties.

B.4.2.2.3 Content-proof Requested

If, in the subject IPM, the Recipient Security Request is set to *content-proof* then:

a) The UA shall validate security arguments present in the subject message as required by the security policy in force. The UA shall copy only validated arguments from the subject message to the IPN Security Response as defined in Table B.1. If the UA cannot validate any of the security arguments in the subject message the UA shall generate an IPN Security Response with an appropriate Security Diagnostic Code, see B.3.

b) If the UA supports Content Integrity, or Message Origin Authentication then the UA shall submit the IPN with one or more of the following (depending on the security policy in force):

i) The security element content-integrity-check, possibly in the message-token; or

ii) the security element message-origin-authentication-check.

If the UA does not supports Content Integrity, or Message Origin Authentication the UA shall generate an IPN Security Response with a Security Diagnostic Code of *unsupported-request-for-security-service.*

c) If the UA supports Content Integrity, or Message Origin Authentication but the subject message has not included any of the following security arguments:

i) message-token;

ii) content-integrity-check; or

iii) message-origin-authentication-check.

Then the content of the subject message shall be copied into the original-content of the IPN Security Response or the UA shall generate an IPN Security Response with an appropriate Security Diagnostic Code.

NOTE – The choice of returning the content or Security Diagnostic Code is a local issue and can be a matter for the User or the implementation to decide.

Then the UA shall submit the IPN with one or more of the following (depending on the security policy in force):

i) the security element content-integrity-check, possibly in the message-token; or

ii) the security element message-origin-authentication-check.

B.4.2.2.4 IPN-non-repudiation Requested

If, in the Subject IPM, Recipient Security Request is set to *ipn-non-repudiation* then:

a) If the UA supports Non-Repudiation of Origin, then the UA shall submit the IPN with one or more of the following (depending on the security policy in force):

i) the security element content-integrity-check, possibly in the message-token, which possesses non-repudiation properties; or

ii) the security element message-origin-authentication-check which possesses non-repudiation properties.

If the UA does not supports Non-Repudiation of Origin the UA shall generate an IPN Security Response with an appropriate Security Diagnostic Code.

B.4.2.2.5 IPN-proof Requested

If, in the Subject IPM, Recipient Security Request is set to *ipn-proof* then:

a) If the UA supports Content Integrity, or Message Origin Authentication, then the UA shall submit the IPN with one or more of the following (depending on the security policy in force):

i) the security element content-integrity-check, possibly in the message-token; or

ii) the security element message-origin-authentication-check.

If the UA does not supports the Content Integrity, or Message Origin Authentication the UA shall generate an IPN Security Response with an appropriate Security Diagnostic Code.

B.5 Additional MS Procedures

If the Recipient Security Request is present in a subject message, then the Message Store actions are subject to the security policy in force. No additional procedures for the message store are defined in this Specification.

B.6 MTS Extensions

B.6.1 Body Part Encryption Token

The **Body Part Encryption Token** MTS extension may be present in the per-recipient-message-submission-extensions field of a message-submission envelope and in the message-delivery-extensions field of a message-delivery envelope. The Body Part Encryption Token shall be present only if the IPM contains (directly, or within a forwarded IPM) at least one encrypted body part (see 7.4.6) whose algorithm-identifier identifies a symmetric encryption algorithm. The Body Part Encryption Token contains a protected copy of the session key for each body part that has been encrypted using a symmetric encryption algorithm.

```
body-part-encryption-token EXTENSION ::= {
        BodyPartTokens,
        RECOMMENDED CRITICALITY {for-delivery},
        IDENTIFIED BY standard-extension:43 }

BodyPartTokens ::= SET OF SET {
        body-part-number             BodyPartNumber,
        body-part-choice             CHOICE {
             encryption-token                EncryptionToken,
             message-or-content-body-part    BodyPartTokens } }
```

Each member of **body-part-tokens** has the following components:

a) **Body-part-number** (M): Identifies one of the body parts in this IPM, numbered starting at '1', which is either an Encrypted body part or else a Message or Forwarded Content body part which contains (directly or recursively) an Encrypted body part.

NOTE – A body-part-number may occur twice in a body-part-token only if an encrypted body part contains a Message or Forwarded Content body part which itself contains an encrypted body part.

b) **Encryption-token** (C): Contains a protected copy of a session key used with a symmetric encryption algorithm to encrypt the identified body part, if the identified body part is an Encrypted body part.

```
EncryptionToken ::= SET {
        encryption-algorithm-identifier      AlgorithmIdentifier,
        encrypted-key                        ENCRYPTED { EncryptionKey },
        recipient-certificate-selector   [0] CertificateAssertion OPTIONAL,
        recipient-certificate            [1] Certificates OPTIONAL,
        originator-certificate-selector  [2] CertificateAssertion OPTIONAL,
        originator-certificates          [3] ExtendedCertificates OPTIONAL,
        ... }
```

An encryption token has the following components:

i) **Encryption-algorithm-identifier** (M): Identifies the asymmetric encryption algorithm used to protect the session keys, i.e. to compute the encrypted-key component.

ii) **Encrypted-key** (M): This contains a session-key which is encrypted by the originator of the IPM using the algorithm identified by the encryption-algorithm-identifier and the public asymmetric encryption key of the recipient.

iii) **Recipient-certificate-selector** (C): If the recipient has more than one public key for the key encryption algorithm identified by the encryption-algorithm-identifier (and therefore more than one certificate with the same algorithm-identifier) and the recipient-certificate component is absent, then the recipient-certificate-selector shall be present containing components of certificate-assertion (see 12.7.2 of ITU-T Rec. X.509 | ISO/IEC 9594-8) appropriate to enable the recipient to determine which one of the recipient's certificates (and thus which one of the recipient's public keys for this algorithm) was used by the originator, and thus to determine which of the recipient's private keys is applicable.

iv) **Recipient-certificate** (C): If the recipient has more than one public key for the key encryption algorithm identified by the encryption-algorithm-identifier (and therefore more than one certificate with the same algorithm-identifier), then either this component, or the recipient-certificate on the envelope, or the recipient-certificate-selector component shall be present to enable the recipient to determine which one of the recipient's certificates (and thus which one of the recipient's public keys for this algorithm) was used by the originator, and thus to determine which of the recipient's private keys is applicable.

v) **Originator-certificate-selector** (C): If the key encryption algorithm identified by the encryption-algorithm-identifier requires a key of the originator (for example, the Diffie-Hellman algorithm), and if the originator has more than one public key for this algorithm (and therefore more than one certificate with the same algorithm-identifier), then either the originator-certificate-selector shall be present containing components of certificate-assertion (see 12.7.2 of ITU-T Rec. X.509 | ISO/IEC 9594-8) appropriate to enable the recipient to determine which one of the originator's certificates (and thus which one of the originator's public keys for this algorithm) is applicable, or else precisely one certificate for this algorithm shall be present within the originator-certificates component, or within the originator-certificates component of the encrypted body part parameters.

vi) **Originator-certificates** (C): This may be used to convey a verified copy of the public-asymmetric-encryption-key of the originator who encrypted the body part (i.e. a Certificate), or the name of a Directory entry containing the originator's certificate, or multiple certificates (or Directory names) where the public-asymmetric-encryption-key is verified with different certification paths or by different certification authorities. This component shall be present only if the asymmetric encryption algorithm requires a key of both originator and recipient (for example, the Diffie-Hellman algorithm). Where the originator certificate is required but is identical for all recipients then this component shall be absent, and the originator-certificate may be included within the parameters of the encrypted body part, or in originator certificate or multiple-originator certificates which may be present on the envelope. If no certificates are present, it is assumed that another method (e.g. use of the Directory) is to be used to obtain a verified copy of the originator's public key.

c) **Message-or-content-body-part** (C): Contains body-part-tokens, if the identified body part is a Message (or Encrypted Message) or Forwarded Content (or Encrypted Forwarded Content) body part which contains (directly or recursively) an Encrypted body part.

B.6.2 Forwarded Content Token

The **Forwarded Content Token** MTS extension, which may be present in the per-recipient-message-submission-extensions field of a message-submission envelope and in the message-delivery-extensions field of a message-delivery envelope, enables an IPM's originator to convey one or more message-tokens (containing encrypted-data) to each of the IPM's recipients. Each Token enables a recipient to verify the security properties of a Forwarded Content body part contained either directly in the Body of the IPM or recursively within a Message or another Forwarded Content body part. The Forwarded Content Token shall be present only if the IPM contains (directly or recursively) at least one Forwarded Content body part (see 7.4.16) where that original message's envelope contains a message-token (see 8.2.1.1.1.26 of ITU-T Rec. X.411 | ISO/IEC 10021-4) which itself contains encrypted-data. The Forwarded Content Token contains a message-token for each such Forwarded Content body part contained (directly or recursively) in the forwarding IPM. The Forwarded Content Token is created by the originator of the forwarding IPM after decrypting the encrypted-data of the forwarded message's message-token (or of its Forwarded Content Token) to contain message-tokens with encrypted-data components encrypted appropriately for each recipient of the forwarding IPM.

```
forwarded-content-token EXTENSION ::= {
       ForwardedContentToken,
       RECOMMENDED CRITICALITY {for-delivery},
       IDENTIFIED BY standard-extension:44 }

ForwardedContentToken ::= SET OF SET {
       body-part-number          BodyPartNumber,
       body-part-choice          CHOICE {
           forwarding-token               MessageToken,
           message-or-content-body-part   ForwardedContentToken } }
```

A Forwarded Content Token has the following components:

a) **Body-part-number** (M): Identifies one of the body parts in this IPM, numbered starting at '1', which is a Message (or Encrypted Message) or Forwarded Content (or Encrypted Forwarded Content) body part.

   NOTE – A body-part-number may occur twice in a Forwarded Content Token only if a Forwarded Content body part containing encrypted content itself contains a Forwarded Content body part containing encrypted content.

b) **Forwarding-token** (C): Contains a message-token (see 8.2.1.1.1.26 of ITU-T Rec. X.411 | ISO/IEC 10021-4) which itself contains encrypted-data which is encrypted appropriately for each recipient by the originator of the forwarding IPM, if the identified body part is a Forwarded Content body part where that message's envelope contains a message-token which itself contains encrypted-data.

c) **Message-or-content-body-part** (C): Contains a Forwarded Content Token if the identified body part is a Message (or Encrypted Message) or Forwarded Content (or Encrypted Forwarded Content) body part which itself contains (directly or recursively) another Forwarded Content body part with which a Forwarding-token is to be associated.

Annex C

Reference Definition of Object Identifiers

(This annex forms an integral part of this Recommendation | International Standard)

The 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 Specification 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-itu-t mhs(6) ipms(1) modules(0) object-identifiers(0)
               version-1999(1) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS -- nothing -- ;

ID ::= OBJECT IDENTIFIER

-- Interpersonal Messaging (not definitive)

id-ipms ID ::= {joint-iso-itu-t mhs(6) ipms(1)} -- not definitive

-- 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-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
id-cat ID ::= {id-ipms 13} -- correlation attributes
id-mr  ID ::= {id-ipms 14} -- matching-rules
id-aa  ID ::= {id-ipms 15} -- auto-actions
id-aae ID ::= {id-ipms 16} -- auto-action errors
id-mst ID ::= {id-ipms 17} -- message store types
id-sec ID ::= {id-ipms 18} -- ipm security extensions
id-on  ID ::= {id-ipms 19} -- other notification type extensions
id-rex ID ::= {id-ipms 20} -- recipient extensions

-- 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
id-mod-forwarded-report-body-part-type  ID ::= {id-mod 12} -- not definitive
id-mod-auto-actions                     ID ::= {id-mod 13} -- not definitive
id-mod-ipm-security-extensions          ID ::= {id-mod 14} -- not definitive
id-mod-forwarded-content-body-part-type ID ::= {id-mod 15} -- not definitive
id-mod-pkcs7-body-part-type             ID ::= {id-mod 16} -- not definitive
```

```
-- Object types

id-ot-ipms-user ID ::= {id-ot 1}
id-ot-ipms      ID ::= {id-ot 2}

-- Port types

id-pt-origination ID ::= {id-pt 0}
id-pt-reception   ID ::= {id-pt 1}
id-pt-management  ID ::= {id-pt 2}

-- 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}
--  Value {id-et 13} is no longer defined
id-et-report              ID ::= {id-et 14}
id-et-notification        ID ::= {id-et 15}
id-et-voice               ID ::= {id-et 16}
id-et-content             ID ::= {id-et 17}   -- This value is not used directly, only as a prefix
id-et-pkcs7               ID ::= {id-et 18}

-- 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}
id-hex-body-part-signatures         ID ::= {id-hex 3}
id-hex-ipm-security-label           ID ::= {id-hex 4}
id-hex-authorization-time           ID ::= {id-hex 5}
id-hex-circulation-list-recipients  ID ::= {id-hex 6}
id-hex-distribution-codes           ID ::= {id-hex 7}
id-hex-extended-subject             ID ::= {id-hex 8}
id-hex-information-category         ID ::= {id-hex 9}
id-hex-manual-handling-instructions ID ::= {id-hex 10}
id-hex-originators-reference        ID ::= {id-hex 11}
id-hex-precedence-policy-id         ID ::= {id-hex 12}

-- Summary attributes

id-sat-ipm-entry-type        ID ::= {id-sat 0}
id-sat-ipm-synopsis          ID ::= {id-sat 1}
id-sat-body-parts-summary    ID ::= {id-sat 2}
id-sat-ipm-auto-discarded    ID ::= {id-sat 3}
```

```
-- 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}
id-hat-body-part-signatures                    ID ::= {id-hat 23}
id-hat-ipm-security-label                      ID ::= {id-hat 24}
id-hat-body-part-security-label                ID ::= {id-hat 25}
id-hat-body-part-encryption-token              ID ::= {id-hat 26}
id-hat-authorization-time                      ID ::= {id-hat 27}
id-hat-circulation-list-recipients             ID ::= {id-hat 28}
id-hat-distribution-codes                      ID ::= {id-hat 29}
id-hat-extended-subject                        ID ::= {id-hat 30}
id-hat-information-category                    ID ::= {id-hat 31}
id-hat-manual-handling-instructions            ID ::= {id-hat 32}
id-hat-originators-reference                   ID ::= {id-hat 33}
id-hat-precedence-policy-id                    ID ::= {id-hat 34}
id-hat-forwarded-content-token                 ID ::= {id-hat 35}
id-hat-forwarding-token                        ID ::= {id-hat 36}
id-hat-precedence                              ID ::= {id-hat 37}
id-hat-body-part-signature-verification-status ID ::= {id-hat 38}

-- 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-intended-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}
id-nat-notification-extensions         ID ::= {id-nat 11}
id-nat-nrn-extensions                  ID ::= {id-nat 12}
id-nat-rn-extensions                   ID ::= {id-nat 13}
id-nat-other-notification-type-fields  ID ::= {id-nat 14}

-- Correlation attributes

id-cat-correlated-delivered-ipns       ID ::= {id-cat  0}
id-cat-correlated-delivered-replies    ID ::= {id-cat  1}
id-cat-delivered-ipn-summary           ID ::= {id-cat  2}
id-cat-delivered-replies-summary       ID ::= {id-cat  3}
id-cat-forwarded-ipms                  ID ::= {id-cat  4}
id-cat-forwarding-ipms                 ID ::= {id-cat  5}
id-cat-ipm-recipients                  ID ::= {id-cat  6}
id-cat-obsoleted-ipms                  ID ::= {id-cat  7}
id-cat-obsoleting-ipms                 ID ::= {id-cat  8}
id-cat-related-ipms                    ID ::= {id-cat  9}
id-cat-relating-ipms                   ID ::= {id-cat 10}
id-cat-replied-to-ipm                  ID ::= {id-cat 11}
id-cat-replying-ipms                   ID ::= {id-cat 12}
id-cat-revised-reply-time              ID ::= {id-cat 13}
id-cat-submitted-ipn-status            ID ::= {id-cat 14}
id-cat-submitted-ipns                  ID ::= {id-cat 15}
id-cat-submitted-reply-status          ID ::= {id-cat 16}
id-cat-subject-ipm                     ID ::= {id-cat 17}
id-cat-recipient-category              ID ::= {id-cat 18}

-- Message content types (for use by MS and Directory)

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}
--  Value {id-ep 13} is no longer defined
id-ep-notification        ID ::= {id-ep 15}
id-ep-voice               ID ::= {id-ep 16}
id-ep-content             ID ::= {id-ep 17}   -- This value is not used directly, only as a prefix

-- Encoded Information Types

id-eit-file-transfer  ID ::= {id-eit 0}
id-eit-voice          ID ::= {id-eit 1}

-- Voice Encoded Information Types

id-voice-11khz-sample      ID ::= {id-eit-voice 0}
id-voice-22khz-sample      ID ::= {id-eit-voice 1}
id-voice-cd-quality        ID ::= {id-eit-voice 2}
id-voice-g711-mu-law       ID ::= {id-eit-voice 3}
id-voice-g726-32k-adpcm    ID ::= {id-eit-voice 4}
id-voice-g728-16k-ld-celp  ID ::= {id-eit-voice 5}
```

```
-- Matching-rules

id-mr-ipm-identifier                           ID ::= {id-mr 0}
id-mr-or-descriptor                            ID ::= {id-mr 1}
id-mr-or-descriptor-elements                   ID ::= {id-mr 2}
id-mr-or-descriptor-substring-elements         ID ::= {id-mr 3}
id-mr-recipient-specifier                      ID ::= {id-mr 4}
id-mr-recipient-specifier-elements             ID ::= {id-mr 5}
id-mr-recipient-specifier-substring-elements   ID ::= {id-mr 6}
id-mr-ipm-location                             ID ::= {id-mr 7}
id-mr-or-descriptor-single-element             ID ::= {id-mr 8}
id-mr-recipient-specifier-single-element       ID ::= {id-mr 9}
id-mr-circulation-member                       ID ::= {id-mr 10}
id-mr-circulation-member-elements              ID ::= {id-mr 11}
id-mr-circulation-member-substring-elements    ID ::= {id-mr 12}
id-mr-circulation-member-single-element        ID ::= {id-mr 13}
id-mr-circulation-member-checkmark             ID ::= {id-mr 14}
id-mr-distribution-code                        ID ::= {id-mr 15}
id-mr-information-category                     ID ::= {id-mr 16}

-- Auto-actions

id-aa-ipm-auto-acknowledgement                 ID ::= {id-aa 0}
id-aa-ipm-auto-correlate                       ID ::= {id-aa 1}
id-aa-ipm-auto-discard                         ID ::= {id-aa 2}
id-aa-ipm-auto-advise                          ID ::= {id-aa 3}

-- Auto-action-errors

id-aae-auto-discard-error                      ID ::= {id-aae 0}
id-aae-auto-forwarding-loop                    ID ::= {id-aae 1}
id-aae-duplicate-ipn                           ID ::= {id-aae 2}

-- Message Store types

id-mst-invalid-assembly-instructions           ID ::= {id-mst 0}
id-mst-invalid-ipn                             ID ::= {id-mst 1}
id-mst-assembly-instructions                   ID ::= {id-mst 2}
id-mst-suspend-auto-acknowledgement            ID ::= {id-mst 3}
id-mst-prevent-nrn-generation                  ID ::= {id-mst 4}
id-mst-originator-body-part-encryption-token   ID ::= {id-mst 5}
id-mst-originator-forwarded-content-token      ID ::= {id-mst 6}
id-mst-assembly-capability                     ID ::= {id-mst 7}

-- Security extensions

id-sec-ipm-security-request                    ID ::= {id-sec 0}
id-sec-security-common-fields                  ID ::= {id-sec 1}

-- Other notification types

id-on-absence-advice              ID ::= {id-on 0}
id-on-change-of-address-advice    ID ::= {id-on 1}

-- Recipient extensions

id-rex-circulation-list-indicator   ID ::= {id-rex 0}
id-rex-precedence                   ID ::= {id-rex 1}

END -- of IPMSObjectIdentifiers
```

```
IPMSObjectIdentifiers2 { iso standard mhs(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 mhs(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 D

Reference Definition of Abstract Information Objects

(This annex forms an integral part of this Recommendation | International Standard)

This annex, a supplement to section two, defines for reference purposes the abstract information objects of Interpersonal Messaging.

```
IPMSInformationObjects { joint-iso-itu-t mhs(6) ipms(1) modules(0) information-objects(2)
               version-1999(1) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS

       -- IPMS Extended Body Parts

       bilaterally-defined-body-part, encrypted-body-part, g3-facsimile-body-part,
       g4-class1-body-part, ia5-text-body-part, message-body-part, mixed-mode-body-part,
       nationally-defined-body-part, teletex-body-part, videotex-body-part
               ----
               FROM IPMSExtendedBodyPartTypes { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      extended-body-part-types(7) version-1994(0) }

       general-text-body-part
               ----
               FROM IPMSExtendedBodyPartTypes2 { iso standard mhs(10021) ipms(7) modules(0)
                      extended-body-part-types-2(1) }

       file-transfer-body-part
               ----
               FROM IPMSFileTransferBodyPartType { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      file-transfer-body-part-type(9) }

       voice-body-part
               ----
               FROM IPMSExtendedVoiceBodyPartType { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      extended-voice-body-part-type(11) }

       notification-body-part, report-body-part
               ----
               FROM IPMSForwardedReportBodyPartType { joint-iso-itu-t mhs(6) ipms(1)
       modules(0)
                      forwarded-report-body-part-type(12) }

       content-body-part { }
               ----
               FROM IPMSForwardedContentBodyPartType { joint-iso-itu-t mhs(6) ipms(1)
                      modules(0) forwarded-content-body-part-type(15) }

       pkcs7-body-part
               ----
               FROM PKCS7BodyPartType { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      pkcs7-body-part-type(16) }

       -- IPMS Heading Extensions

       authorization-time, auto-submitted, body-part-signatures, circulation-list-indicator,
       circulation-list-recipients, distribution-codes, extended-subject, incomplete-copy,
       information-category, ipm-security-label, languages, manual-handling-instructions,
       originators-reference, precedence, precedence-policy-identifier
               ----
               FROM IPMSHeadingExtensions { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      heading-extensions(6) version-1999(1) }
```

```
        -- IPMS Security Extensions

        body-part-encryption-token, BodyPartTokens, forwarded-content-token,
        ForwardedContentToken, ipn-security-response, recipient-security-request
                ----
                FROM IPMSSecurityExtensions { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                        ipm-security-extensions(14) version-1999(1) }

        -- IPMS Upper bounds

        ub-auto-forward-comment, ub-free-form-name, ub-local-ipm-identifier, ub-subject-field,
        ub-telephone-number
                ----
                FROM IPMSUpperBounds { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                        upper-bounds(10) version-1999(1) }

        -- ODIF

        Interchange-Data-Element
                ----
                FROM Interchange-Data-Elements { 2 8 1 5 5 }

        -- MTS Abstract Service

        EncodedInformationTypes, ExtendedCertificates, EXTENSION,
        G3FacsimileNonBasicParameters, MessageDeliveryTime, ORName, OtherMessageDeliveryFields,
        SupplementaryInformation, TeletexNonBasicParameters
                ----
                FROM MTSAbstractService { joint-iso-itu-t mhs(6) mts(3) modules(0)
                        mts-abstract-service(1) version-1999(1) }

        -- MS Abstract Service

        MS-EXTENSION, SequenceNumber
                ----
                FROM MSAbstractService { joint-iso-itu-t mhs(6) ms(4) modules(0)
                        abstract-service(1) version-1999(1) }

        -- Directory Authentication Framework

        AlgorithmIdentifier, ENCRYPTED {}
                ----
                FROM AuthenticationFramework { joint-iso-itu-t ds(5) module(1)
                        authenticationFramework(7) 3 }

        -- IPMS Object Identifiers

        id-mst-assembly-capability, id-mst-assembly-instructions,
        id-mst-invalid-assembly-instructions, id-mst-invalid-ipn,
        id-mst-originator-body-part-encryption-token,
        id-mst-originator-forwarded-content-token, id-mst-suspend-auto-acknowledgement,
        id-mst-prevent-nrn-generation      , id-on-absence-advice,
        id-on-change-of-address-advice
                ----
                FROM IPMSObjectIdentifiers { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                        object-identifiers(0) version-1999(1) };

Time ::= UTCTime

-- Information object

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

-- IPM

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

```
-- MTS Extensions

IPMPerRecipientEnvelopeExtensions EXTENSION  ::= {
       blind-copy-recipients |
       body-part-encryption-token |
       forwarded-content-token, ... }

-- IPMS Extensions

IPMSExtension {IPMS-EXTENSION:ChosenFrom} ::= SEQUENCE {
       type  IPMS-EXTENSION.&id({ChosenFrom}),
       value IPMS-EXTENSION.&Type({ChosenFrom} {@type}) DEFAULT NULL:NULL }

IPMS-EXTENSION ::= CLASS {
       &id OBJECT IDENTIFIER UNIQUE,
       &Type DEFAULT NULL }
       WITH SYNTAX { [VALUE &Type , ] IDENTIFIED BY &id }

PrivateIPMSExtensions_IPMS-EXTENSION ::= { ... }

-- 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),
       an-supported (3),
       suppress-an  (4)}

RecipientExtensionsField ::= SET OF IPMSExtension {{ RecipientExtensions }}
```

```
RecipientExtensions_IPMS-EXTENSION ::= {
       circulation-list-indicator |
       precedence |
       recipient-security-request |
       PrivateIPMSExtensions, ... }
```

-- *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
```

-- *Blind Copy Recipients envelope field*

```
blind-copy-recipients EXTENSION  ::= {
       BlindCopyRecipientsField,
       IDENTIFIED BY standard-extension:41 }
```

-- *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 (WITH COMPONENTS{..., formal-name PRESENT})

-- 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 {{ HeadingExtensions }}

HeadingExtensions IPMS-EXTENSION ::= {
       authorization-time |
       auto-submitted |
       body-part-signatures |
       circulation-list-recipients |
       distribution-codes |
       extended-subject |
       incomplete-copy |
       information-category |
       ipm-security-label |
       languages |
       manual-handling-instructions |
       originators-reference |
       precedence-policy-identifier |
       PrivateIPMSExtensions, ... }

-- Body

Body ::= SEQUENCE OF BodyPart

BodyPart ::= CHOICE {
       basic     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 },
       extended                   [15]  ExtendedBodyPart}{{IPMBodyPartTable}}
```

```
-- Extended body part

ExtendedBodyPart{EXTENDED-BODY-PART-TYPE:IPMBodyPartTable} ::= SEQUENCE {
       parameters [0] INSTANCE OF TYPE-IDENTIFIER OPTIONAL,
       data       INSTANCE OF TYPE-IDENTIFIER }
                                   (CONSTRAINED BY {-- must correspond to the &parameters field and
&data field
              -- of a member of -- IPMBodyPartTable})

IPMBodyPartTable EXTENDED-BODY-PART-TYPE ::= {
       StandardBodyParts |
       ApplicationSpecificBodyParts }

StandardBodyParts EXTENDED-BODY-PART-TYPE ::= {
       ia5-text-body-part |
       g3-facsimile-body-part |
       g4-class1-body-part |
       teletex-body-part |
       videotex-body-part |
       encrypted-body-part |
       message-body-part |
       mixed-mode-body-part |
       bilaterally-defined-body-part |
       nationally-defined-body-part |
       general-text-body-part |
       file-transfer-body-part |
       voice-body-part |
       report-body-part |
       notification-body-part |
       content-body-part |{{1 2 3 -- RELATIVE-OID to be provided --}}
       pkcs7-body-part,
       ... }

ApplicationSpecificBodyParts EXTENDED-BODY-PART-TYPE ::= {
       --any body part defined in other Specifications, or for proprietary or private use --
       ... }

EXTENDED-BODY-PART-TYPE ::= CLASS {
       &parameters   TYPE-IDENTIFIER OPTIONAL,
       &data         TYPE-IDENTIFIER }
       WITH SYNTAX { [PARAMETERS  &parameters,] DATA  &data }

-- 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 {
       algorithm-identifier      AlgorithmIdentifier,
       originator-certificates   ExtendedCertificates OPTIONAL,
       ... }

EncryptedData ::= ENCRYPTED { BodyPart }

-- 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

NATIONAL-BODY-PARTS ::= CLASS {&Type}

NationallyDefinedBodyPart ::= NATIONAL-BODY-PARTS.&Type
       -- Provided for Historic reasons. Use is strongly deprecated.
-- 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 COMPONENTS {
              ... ,
              choice (WITH COMPONENTS { receipt-fields PRESENT }) })

NRN ::= IPN (WITH COMPONENTS {
              ... ,
              choice (WITH COMPONENTS { non-receipt-fields PRESENT }) })

ON ::= IPN (WITH COMPONENTS {
              ... ,
              choice (WITH COMPONENTS { other-notification-type-fields PRESENT }) })

CommonFields ::= SET {
       subject-ipm                 SubjectIPMField,
       ipn-originator          [1] IPNOriginatorField OPTIONAL,
       ipm-intended-recipient  [2] IPMIntendedRecipientField 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

IPMIntendedRecipientField ::= ORDescriptor

ConversionEITsField ::= EncodedInformationTypes

NotificationExtensionsField ::= SET OF IPMSExtension {{ NotificationExtensions }}

NotificationExtensions IPMS-EXTENSION ::= {
       ipn-security-response |
       PrivateIPMSExtensions, ... }

-- Non-receipt fields

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


[ ITU-T version:
DiscardReasonField ::= ENUMERATED {
       ipm-expired                  (0),
       ipm-obsoleted                (1),
       user-subscription-terminated (2),
       not-used                     (3) }
```

| *ISO/IEC version:*

```
DiscardReasonField ::= ENUMERATED {
        ipm-expired                   (0),
        ipm-obsoleted                 (1),
        user-subscription-terminated  (2),
        -- The following value may not be supported by implementations of earlier versions of this Specification
        ipm-deleted                   (3),
        ... }
```

]

```
AutoForwardCommentField ::= AutoForwardComment

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

ReturnedIPMField ::= IPM

NRNExtensionsField ::= SET OF IPMSExtension {{_NRNExtensions_}}

NRNExtensions_IPMS-EXTENSION ::= {
        PrivateIPMSExtensions, ... }

-- Receipt fields

ReceiptTimeField ::= Time

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

SupplReceiptInfoField ::= SupplementaryInformation

RNExtensionsField ::= SET OF IPMSExtension {{_RNExtensions_}}

RNExtensions_IPMS-EXTENSION ::= {
        PrivateIPMSExtensions, ... }

-- Other Notification Type fields

OtherNotificationTypeFields ::= SET OF IPMSExtension {{ OtherNotifications }}

OtherNotifications IPMS-EXTENSION ::= {
        AdviceNotifications |
        PrivateIPMSExtensions, ... }

AdviceNotifications IPMS-EXTENSION ::= {
        absence-advice |
        change-of-address-advice,
        ... }

-- Advice Notification fields

absence-advice IPMS-EXTENSION ::= {
        VALUE AbsenceAdvice, IDENTIFIED BY id-on-absence-advice}

AbsenceAdvice ::= SEQUENCE {
        advice            BodyPart OPTIONAL,
        next-available    Time OPTIONAL
        -- at least one component shall be present -- }

change-of-address-advice IPMS-EXTENSION ::= {
        VALUE ChangeOfAddressAdvice, IDENTIFIED BY id-on-change-of-address-advice}

ChangeOfAddressAdvice ::= SEQUENCE {
        new-address      [0] ORDescriptor (WITH COMPONENTS {
                             ... ,
                             formal-name PRESENT }),
        effective-from   [1] Time OPTIONAL }
```

```
-- Message Store Realization

prevent-nrn-generation MS-EXTENSION ::= {
        NULL  IDENTIFIED BY  id-mst-prevent-nrn-generation }

suspend-auto-acknowledgement MS-EXTENSION ::= {
        NULL  IDENTIFIED BY  id-mst-suspend-auto-acknowledgement }

assembly-capability MS-EXTENSION ::= {
        NULL IDENTIFIED BY id-mst-assembly-capability }

IPMSubmissionOptions MS-EXTENSION ::= {
        ipm-assembly-instructions |
        originator-body-part-encryption-token |
        originator-forwarded-content-token,
        . . .  -- For future extension additions --  }

ipm-assembly-instructions MS-EXTENSION ::= {
        IPMAssemblyInstructions  IDENTIFIED BY  id-mst-assembly-instructions }

IPMAssemblyInstructions ::= SET {
        assembly-instructions   [0]  BodyPartReferences }

BodyPartReferences ::= SEQUENCE OF BodyPartReference

BodyPartReference ::= CHOICE {
        stored-entry            [0]  SequenceNumber,
        stored-content          [1]  SequenceNumber,
        submitted-body-part     [2]  INTEGER (1..MAX),
        stored-body-part        [3]  SEQUENCE {
               message-entry           SequenceNumber,
               body-part-number        INTEGER (1..MAX) } }

originator-body-part-encryption-token MS-EXTENSION ::= {
        BodyPartTokens IDENTIFIED BY id-mst-originator-body-part-encryption-token }

originator-forwarded-content-token MS-EXTENSION ::= {
        ForwardedContentToken IDENTIFIED BY id-mst-originator-forwarded-content-token }

IPMSubmissionErrors MS-EXTENSION ::= {
        invalid-assembly-instructions |
        invalid-ipn,
        . . .  -- For future extension additions --  }

invalid-assembly-instructions MS-EXTENSION ::= {
        BodyPartReferences  IDENTIFIED BY  id-mst-invalid-assembly-instructions }

invalid-ipn MS-EXTENSION ::= {
        NULL  IDENTIFIED BY  id-mst-invalid-ipn }

END -- of IPMSInformationObjects
```

Annex E

Reference Definition of Extended Body Part Types

(This annex forms an integral part of this Recommendation | International Standard)

This annex, a supplement to 7.4, defines for reference purposes certain extended body part types.

E.1 Equivalents of Basic Body Part Types

```
IPMSExtendedBodyPartTypes { joint-iso-itu-t mhs(6) ipms(1) modules(0)
              extended-body-part-types(7)version-1994(0) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
-- Prologue
-- Exports everything.

IMPORTS

       -- IPMS Information Objects

       BilaterallyDefinedBodyPart, EncryptedData, EncryptedParameters,
       EXTENDED-BODY-PART-TYPE, G3FacsimileData, G3FacsimileParameters, G4Class1BodyPart,
       IA5TextData, IA5TextParameters, MessageData, MessageParameters, MixedModeBodyPart,
       NationallyDefinedBodyPart, TeletexData, TeletexParameters, VideotexData,
       VideotexParameters
                ----
                FROM IPMSInformationObjects { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      information-objects(2) version-1999(1) }

       -- IPMS Object Identifiers

       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-itu-t mhs(6) ipms(1) modules(0)
                      object-identifiers(0) version-1999(1) };

-- Extended IA5 Text body part

ia5-text-body-part EXTENDED-BODY-PART-TYPE ::= {
       PARAMETERS {IA5TextParameters IDENTIFIED BY id-ep-ia5-text},
       DATA       {IA5TextData IDENTIFIED BY 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 IDENTIFIED BY id-et-g3-facsimile} }

-- Extended G4 Class 1 body part

g4-class1-body-part EXTENDED-BODY-PART-TYPE ::= {
       DATA {G4Class1BodyPart IDENTIFIED BY id-et-g4-class1} }

-- Extended Teletex body part

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

```
-- Extended Videotex body part

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

-- Extended Encrypted body part

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

-- Extended Message body part

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

-- Extended Mixed-mode body part

mixed-mode-body-part EXTENDED-BODY-PART-TYPE ::= {
       DATA {MixedModeBodyPart IDENTIFIED BY id-et-mixed-mode} }

-- Extended Bilaterally Defined body part

bilaterally-defined-body-part EXTENDED-BODY-PART-TYPE ::= {
       DATA {BilaterallyDefinedBodyPart IDENTIFIED BY id-et-bilaterally-defined} }

-- Extended Nationally Defined body part

nationally-defined-body-part EXTENDED-BODY-PART-TYPE ::= {
       DATA {NationallyDefinedBodyPart IDENTIFIED BY id-et-nationally-defined} }

END -- of IPMSExtendedBodyPartTypes
```

E.2 General Text

```
                                                  ----------

IPMSExtendedBodyPartTypes2 {iso standard mhs(10021) ipms(7) modules(0)
              extended-body-part-types-2(1)}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS

        -- IPMS Information Objects

       EXTENDED-BODY-PART-TYPE
               ----
               FROM IPMSInformationObjects { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      information-objects(2) version-1999(1) }

       -- IPMS Object Identifiers

       id-ep-general-text, id-et-general-text
               ----
               FROM IPMSObjectIdentifiers { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      object-identifiers(0) version-1999(1) };

-- General Text body part

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

```
GeneralTextParameters ::= SET OF CharacterSetRegistration

GeneralTextData ::= GeneralString

CharacterSetRegistration ::= INTEGER (1..32767)

END -- of IPMSExtendedBodyPartTypes2
```

E.3 File Transfer

```
IPMSFileTransferBodyPartType { joint-iso-itu-t mhs(6) ipms(1) modules(0)
               file-transfer-body-part-type(9) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS

       -- FTAM Attribute Types

       Attribute-Extensions, Concurrency-Access, Date-and-Time-Attribute,
       Legal-Qualification-Attribute, Object-Availability-Attribute, Object-Size-Attribute,
       Pathname, Permitted-Actions-Attribute, Private-Use-Attribute
               ----
               FROM ISO8571-FTAM

       -- ACSE definitions of AP-title and AE-qualifier

       AE-qualifier, AP-title
               ----
               FROM ACSE-1 { joint-iso-itu-t association-control(2) modules(0) apdus(0)
                      version1(1) }

       -- IPMS Information Objects

       EXTENDED-BODY-PART-TYPE, ExtensionsField
               ----
               FROM IPMSInformationObjects { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      information-objects(2) version-1999(1) }

       -- IPMS Object Identifiers

       id-ep-file-transfer, id-et-file-transfer
               ----
               FROM IPMSObjectIdentifiers { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      object-identifiers(0) version-1999(1) }

       -- MTS Abstract Service

       ORName
               ----
               FROM MTSAbstractService { joint-iso-itu-t mhs(6) mts(3) modules(0)
                      mts-abstract-service(1) version-1999(1) };

-- File Transfer body part

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

FileTransferParameters ::= SEQUENCE {
       related-stored-file [0] RelatedStoredFile OPTIONAL,
       contents-type       [1] ContentsTypeParameter DEFAULT document-type:{document-type-name
                               {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 OPTIONAL }
```

```
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 explicit-relationship: unspecified }

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

PathnameandVersion ::= SEQUENCE {
       pathname      [0] Pathname-Attribute,
       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 OPTIONAL,
       -- Defined in 8.5.5 of ITU-T Rec. X.411 | ISO/IEC 10021-4
       user-relative-identifier  [1] PrintableString }

Relationship ::= CHOICE {
       explicit-relationship     [0] ExplicitRelationship,
       descriptive-relationship  [1] GraphicString }

ExplicitRelationship ::= INTEGER {
       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] DOCUMENT-PARAMETER.&Type 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 } }

Document-Type-Name ::= OBJECT IDENTIFIER

DOCUMENT-PARAMETER ::= CLASS {&Type}

Constraint-Set-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]
COMPRESSION-ALGORITHM.&id ({CompressionAlgorithmTable}),
       compression-algorithm-param  [1]

COMPRESSION-ALGORITHM.&Type ({CompressionAlgorithmTable} {@compression-algorithm-id}) }

COMPRESSION-ALGORITHM ::= TYPE-IDENTIFIER
```

```
CompressionAlgorithmTable COMPRESSION-ALGORITHM ::= { ... }

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,
       date-and-time-of-last-attribute-modification[7] 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,
       identity-of-last-attribute-modifier           [11] User-Identity-Attribute OPTIONAL,
       object-availability                           [12] Object-Availability-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 }

Pathname-Attribute ::=  CHOICE {
       incomplete-pathname [0]  Pathname,
       complete-pathname   [23] Pathname }

Account-Attribute ::=  CHOICE {
       no-value-available [0] NULL,
       -- Indicates 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 }

User-Identity ::= GraphicString

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 }

Password ::= CHOICE { graphic-string GraphicString, octet-string OCTET STRING }
```

```
Pass-Passwords ::= SEQUENCE OF Password

Application-Entity-Title ::= SEQUENCE {
       ap-title     AP-title,
       ae-qualifier AE-qualifier }

END -- of IPMSFileTransferBodyPartType
```

E.4 Voice

```
IPMSExtendedVoiceBodyPartType { joint-iso-itu-t mhs(6) ipms(1) modules(0)
              extended-voice-body-part-type(11) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
-- Prologue

-- Exports everything.

IMPORTS

       -- IPMS Information Objects

       EXTENDED-BODY-PART-TYPE
               ----
               FROM IPMSInformationObjects { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      information-objects(2) version-1999(1) }

       -- IPMS Object Identifiers

       id-ep-voice, id-et-voice
               ----
               FROM IPMSObjectIdentifiers { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      object-identifiers(0) version-1999(1) };

-- Extended Voice body part

voice-body-part EXTENDED-BODY-PART-TYPE ::= {
       PARAMETERS {VoiceParameters IDENTIFIED BY id-ep-voice},
       DATA       {VoiceData IDENTIFIED BY id-et-voice} }

VoiceParameters ::= SEQUENCE {
       voice-message-duration [0] INTEGER OPTIONAL,   -- In seconds
       voice-encoding-type [1] OBJECT IDENTIFIER,
       supplementary-information [2] IA5String OPTIONAL }

VoiceData ::= OCTET STRING

END -- of IPMSExtendedVoiceBodyPartType
```

E.5 Report and Notification

```
IPMSForwardedReportBodyPartType {joint-iso-itu-t mhs(6) ipms(1) modules(0)
              forwarded-report-body-part-type(12)}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS

       -- MTS Abstract Service

       ReportDeliveryArgument
               ----
               FROM MTSAbstractService { joint-iso-itu-t mhs(6) mts(3) modules(0)
                      mts-abstract-service(1) version-1999(1) }
```

```
        -- IPMS Information Objects

        EXTENDED-BODY-PART-TYPE, IPN, MessageParameters
                ----
                FROM IPMSInformationObjects { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                        information-objects(2) version-1999(1) }

        -- IPMS Object Identifiers

        id-ep-notification, id-et-report, id-et-notification
                ----
                FROM IPMSObjectIdentifiers { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                        object-identifiers(0) version-1999(1) };

-- Report body part

report-body-part EXTENDED-BODY-PART-TYPE ::= {
        DATA {ReportDeliveryArgument IDENTIFIED BY id-et-report} }

-- Notification body part

notification-body-part EXTENDED-BODY-PART-TYPE ::= {
        PARAMETERS {MessageParameters IDENTIFIED BY id-ep-notification},
        DATA       {IPN IDENTIFIED BY id-et-notification} }

END -- of IPMSForwardedReportBodyPartType
```

E.6 Forwarded Content

```
IPMSForwardedContentBodyPartType {joint-iso-itu-t mhs(6) ipms(1) modules(0)
              forwarded-content-body-part-type(15)}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS

        -- MTS Abstract Service

        Content, ExtendedContentType, MessageDeliveryIdentifier, MessageDeliveryTime,
        MessageSubmissionEnvelope, OriginatingMTACertificate, OtherMessageDeliveryFields,
        ProofOfSubmission
                ----
                FROM MTSAbstractService { joint-iso-itu-t mhs(6) mts(3) modules(0)
                        mts-abstract-service(1) version-1999(1) }

        -- IPMS Information Objects

        EXTENDED-BODY-PART-TYPE
                ----
                FROM IPMSInformationObjects { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                        information-objects(2) version-1999(1) }

        -- IPMS Object Identifiers

        id-ep-content, id-et-content
                ----
                FROM IPMSObjectIdentifiers { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                        object-identifiers(0) version-1999(1) };

-- Forwarded Content body part

content-body-part {ExtendedContentType:content-type} EXTENDED-BODY-PART-TYPE ::= {
        PARAMETERS {ForwardedContentParameters IDENTIFIED BY
                        {id-ep-content -- concatenated with content-type -- content-type}},
        DATA       {Content IDENTIFIED BY {id-et-content -- concatenated with content-type --
content-type}} }
```

```
ForwardedContentParameters ::= SET {
       delivery-time      [0] MessageDeliveryTime OPTIONAL,
       delivery-envelope  [1] OtherMessageDeliveryFields OPTIONAL,
       mts-identifier     [2] MessageDeliveryIdentifier OPTIONAL,
       submission-proof   [3] SubmissionProof OPTIONAL}

SubmissionProof ::= SET {
       proof-of-submission           [0] ProofOfSubmission,
       originating-MTA-certificate   [1] OriginatingMTACertificate,
       message-submission-envelope       MessageSubmissionEnvelope}

END -- of IPMSForwardedContentBodyPartType
```

E.7 PKCS7

```
PKCS7BodyPartType {joint-iso-itu-t mhs(6) ipms(1) modules(0) pkcs7-body-part-type(16) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS

        -- PKCS#7

       ContentInfo
               ----
               FROM PKCS7 { iso(1) member-body(2) usa(840) rsadsi(113549) pkcs(1) 7 module(0)
                      -- module not formally defined in the PKCS#7document, therefore defined in Annex O --  }

       -- IPMS Information Objects

       EXTENDED-BODY-PART-TYPE
               ----
               FROM IPMSInformationObjects { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      information-objects(2) version-1999(1) }

       -- IPMS Object Identifiers

       id-et-pkcs7
               ----
               FROM IPMSObjectIdentifiers { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      object-identifiers(0) version-1999(1) };

-- PKCS7 body part

pkcs7-body-part EXTENDED-BODY-PART-TYPE ::= {
       DATA { ContentInfo IDENTIFIED BY id-et-pkcs7 } }

END -- of PKCS7BodyPartType
```

Annex F

Reference Definition of Functional Objects

(This annex forms an integral part of this Recommendation | International Standard)

This annex, a supplement to clauses 10, 11, and 16, defines for reference purposes the functional objects of Interpersonal Messaging. It uses the MHS-OBJECT information object class of ITU-T Rec. X.411 | ISO/IEC 10021-4 and the CONTRACT information object class of ITU-T Rec. X.880 | ISO/IEC 13712-1.

```
IPMSFunctionalObjects { joint-iso-itu-t mhs(6) ipms(1) modules(0) functional-objects(1)
              version-1994(0) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS

       -- IPMS Abstract Service

       management, origination, reception
               ----
               FROM IPMSAbstractService { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      abstract-service(3) version-1994(0) }

       -- IPMS Object Identifiers

       id-ot-ipms, id-ot-ipms-user
               ----
               FROM IPMSObjectIdentifiers { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      object-identifiers(0) version-1999(1) }

       -- MTS Abstract Service

       MHS-OBJECT
               ----
               FROM MTSAbstractService { joint-iso-itu-t mhs(6) mts(3) modules(0)
                      mts-abstract-service(1) version-1999(1) }

     -- Remote Operations

       CONTRACT
               ----
               FROM Remote-Operations-Information-Objects { joint-iso-itu-t
                      remote-operations(4) informationObjects(5) version1(0) };

-- Primary object types

ipms-user MHS-OBJECT ::= {
       INITIATES    {ipms-access-contract}
       ID           id-ot-ipms-user }

ipms-access-contract CONTRACT ::= {
       INITIATOR CONSUMER OF {origination | reception | management} }

ipms MHS-OBJECT ::= {
       RESPONDS     {ipms-access-contract}
       ID           id-ot-ipms }

END -- of IPMSFunctionalObjects
```

Annex G

Reference Definition of Abstract Service

(This annex forms an integral part of this Recommendation | International Standard)

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 information object class of ITU-T Rec. X.411 | ISO/IEC 10021-4.

```
IPMSAbstractService { joint-iso-itu-t mhs(6) ipms(1) modules(0) abstract-service(3)
              version-1994(0) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS

       -- IPMS Information Objects

       AutoForwardComment, Heading, InformationObject, IPM, NRN, ON, RN
               ----
               FROM IPMSInformationObjects { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      information-objects(2) version-1999(1) }

       -- IPMS Object Identifiers

       id-pt-management, id-pt-origination, id-pt-reception
               ----
               FROM IPMSObjectIdentifiers { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      object-identifiers(0) version-1999(1) }

       -- MTS Abstract Service

       ABSTRACT-ERROR, ABSTRACT-OPERATION, MessageDeliveryEnvelope, MessageSubmissionEnvelope,
       MessageSubmissionIdentifier, MessageSubmissionTime, ORName, PORT,
       ProbeSubmissionEnvelope, ProbeSubmissionIdentifier, ProbeSubmissionTime,
       recipient-improperly-specified, ReportDeliveryEnvelope, SupplementaryInformation
               ----
               FROM MTSAbstractService { joint-iso-itu-t mhs(6) mts(3) modules(0)
                      mts-abstract-service(1) version-1999(1) };

-- Ports

origination PORT ::= {
       CONSUMER INVOKES { originate-probe | originate-ipm | originate-rn | originate-on }
       ID               id-pt-origination }

reception PORT ::= {
       SUPPLIER INVOKES { receive-report | receive-ipm | receive-rn | receive-nrn |
                          receive-on }
       ID               id-pt-reception }

management PORT ::= {
       CONSUMER INVOKES { change-auto-discard | change-auto-acknowledgment |
                          change-auto-forwarding }
       ID               id-pt-management }
```

```
-- Origination abstract operations

originate-probe ABSTRACT-OPERATION ::= {
        ARGUMENT SET {
               envelope [0] ProbeSubmissionEnvelope,
               content  [1] IPM}
        RESULT SET {
               submission-identifier [0] ProbeSubmissionIdentifier,
               submission-time       [1] ProbeSubmissionTime}
        ERRORS {
               subscription-error |
               recipient-improperly-specified} }

originate-ipm ABSTRACT-OPERATION ::= {
        ARGUMENT SET {
               envelope [0] MessageSubmissionEnvelope,
               content  [1] IPM}
        RESULT SET {
               submission-identifier [0] MessageSubmissionIdentifier,
               submission-time       [1] MessageSubmissionTime}
        ERRORS {
               subscription-error |
               recipient-improperly-specified} }

originate-rn ABSTRACT-OPERATION ::= {
        ARGUMENT SET {
               envelope [0] MessageSubmissionEnvelope,
               content  [1] RN}
        RESULT SET {
               submission-identifier [0] MessageSubmissionIdentifier,
               submission-time       [1] MessageSubmissionTime}
        ERRORS {
               subscription-error |
               recipient-improperly-specified} }

originate-on ABSTRACT-OPERATION ::= {
        ARGUMENT SET {
               envelope [0] MessageSubmissionEnvelope,
               content  [1] ON}
        RESULT SET {
               submission-identifier [0] MessageSubmissionIdentifier,
               submission-time       [1] MessageSubmissionTime}
        ERRORS {
               subscription-error |
               recipient-improperly-specified} }

-- Reception abstract operations

receive-report ABSTRACT-OPERATION ::= {
        ARGUMENT SET {
               envelope          [0] ReportDeliveryEnvelope,
               undelivered-object [1] InformationObject OPTIONAL} }

receive-ipm ABSTRACT-OPERATION ::= {
        ARGUMENT SET {
               envelope [0] MessageDeliveryEnvelope,
               content  [1] IPM} }

receive-rn ABSTRACT-OPERATION ::= {
        ARGUMENT SET {
               envelope [0] MessageDeliveryEnvelope,
               content  [1] RN} }

receive-nrn ABSTRACT-OPERATION ::= {
        ARGUMENT SET {
               envelope [0] MessageDeliveryEnvelope,
               content  [1] NRN} }

receive-on ABSTRACT-OPERATION ::= {
        ARGUMENT SET {
               envelope [0] MessageDeliveryEnvelope,
               content  [1] ON} }
```

```
-- Management abstract operations

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

change-auto-acknowledgment ABSTRACT-OPERATION ::= {
        ARGUMENT SET {
               auto-acknowledge-IPMs [0] BOOLEAN,
               auto-acknowledge-suppl-receipt-info [1]
                     SupplementaryInformation OPTIONAL}
        ERRORS {
               subscription-error} }

change-auto-forwarding 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}
        ERRORS {
               subscription-error |
               recipient-improperly-specified} }

-- Abstract errors

subscription-error ABSTRACT-ERROR ::= {
        PARAMETER SET {
               problem [0] SubscriptionProblem} }

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

END -- of IPMSAbstractService
```

Annex H

Reference Definition of IPM Extensions

(This annex forms an integral part of this Recommendation | International Standard)

This annex, a supplement to annex A, defines for reference purposes the general extensions defined for Interpersonal Messaging. It uses the IPMS-EXTENSION information object class of 7.2.17.

```
IPMSHeadingExtensions { joint-iso-itu-t mhs(6) ipms(1) modules(0) heading-extensions(6)
              version-1999(1) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS

       -- IPMS Information Objects

       IPMS-EXTENSION, ORDescriptor, RecipientSpecifier, ThisIPMField, BodyPart
               ----
               FROM IPMSInformationObjects { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      information-objects(2) version-1999(1) }

       -- MTS Abstract Service

       ExtendedCertificates, SecurityLabel, UniversalOrBMPString
               ----
               FROM MTSAbstractService { joint-iso-itu-t mhs(6) mts(3) modules(0)
                      mts-abstract-service(1) version-1999(1) }

       -- Directory Authentication Framework

       AlgorithmIdentifier, SIGNATURE { }, SIGNED { }
               ----
               FROM AuthenticationFramework { joint-iso-itu-t ds(5) module(1)
                      authenticationFramework(7) 3 }

       -- Directory Certificate Extensions

       CertificateAssertion
               ----
               FROM CertificateExtensions { joint-iso-itu-t ds(5) module(1)
                      certificateExtensions(26) 0 }

       -- IPMS upper bounds

       ub-alpha-code-length, ub-circulation-list-members, ub-distribution-codes,
       ub-extended-subject-length, ub-information-categories, ub-information-category-length,
       ub-manual-handling-instruction-length, ub-manual-handling-instructions,
       ub-originators-reference-length, ub-precedence
               ----
               FROM IPMSUpperBounds { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      upper-bounds(10) version-1999(1) }

       -- IPMS Object Identifiers

       id-hex-authorization-time, id-hex-auto-submitted, id-hex-body-part-signatures,
       id-hex-circulation-list-recipients, id-hex-distribution-codes, id-hex-extended-subject,
       id-hex-incomplete-copy, id-hex-information-category, id-hex-ipm-security-label,
       id-hex-languages, id-hex-manual-handling-instructions, id-hex-originators-reference,
       id-hex-precedence-policy-id, id-rex-circulation-list-indicator, id-rex-precedence
               ----
               FROM IPMSObjectIdentifiers { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      object-identifiers(0) version-1999(1) };
```

```
-- Incomplete Copy

incomplete-copy IPMS-EXTENSION ::= {VALUE IncompleteCopy, IDENTIFIED BY
id-hex-incomplete-copy}

IncompleteCopy ::= NULL

-- Languages

languages IPMS-EXTENSION ::= {VALUE SET OF Language, IDENTIFIED BY id-hex-languages}

Language ::= PrintableString (SIZE (2|5))

-- Auto-submitted

auto-submitted IPMS-EXTENSION ::= {VALUE AutoSubmitted, IDENTIFIED BY id-hex-auto-submitted}

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

body-part-signatures IPMS-EXTENSION ::= {VALUE BodyPartSignatures,
                                        IDENTIFIED BY id-hex-body-part-signatures }

BodyPartSignatures ::= SET OF SET {
        body-part-number                BodyPartNumber,
        body-part-signature             BodyPartSignature,
        originator-certificate-selector CertificateAssertion OPTIONAL,
        originator-certificates         [0] ExtendedCertificates OPTIONAL,
        ... }

BodyPartNumber ::= INTEGER (1..MAX)

BodyPartSignature ::= SIGNATURE { SEQUENCE {
        signature-algorithm-identifier  AlgorithmIdentifier,
        body-part                       BodyPart,
        body-part-security-label        SecurityLabel OPTIONAL } }

ipm-security-label IPMS-EXTENSION ::= {
        VALUE                   IPMSecurityLabel,
        IDENTIFIED BY           id-hex-ipm-security-label }

IPMSecurityLabel ::= SEQUENCE{
        content-security-label      [0] SecurityLabel,
        heading-security-label      [1] SecurityLabel OPTIONAL,
        body-part-security-labels   [2] SEQUENCE OF BodyPartSecurityLabel OPTIONAL }

BodyPartSecurityLabel ::= CHOICE {
        body-part-unlabelled        [0] NULL,
        body-part-security-label    [1] SecurityLabel }

-- Authorization Time

authorization-time IPMS-EXTENSION ::= {
        VALUE                   AuthorizationTime,
        IDENTIFIED BY           id-hex-authorization-time }

AuthorizationTime ::= GeneralizedTime

-- Circulation List

circulation-list-recipients IPMS-EXTENSION::= {
        VALUE                   CirculationList,
        IDENTIFIED BY           id-hex-circulation-list-recipients }

CirculationList ::= SEQUENCE (SIZE(2..ub-circulation-list-members))
                              OF CirculationMember
```

```
CirculationMember ::= SET {
        circulation-recipient RecipientSpecifier (WITH COMPONENTS {
                                   ...,
                                   recipient (WITH COMPONENTS {
                                        ...,
                                        formal-name PRESENT } ) } ),
        checked                     Checkmark OPTIONAL }

Checkmark ::= CHOICE {
        simple                NULL,
        timestamped           CirculationTime,
        signed                CirculationSignature }

CirculationTime ::= GeneralizedTime

CirculationSignature ::= SIGNED { SEQUENCE {
        algorithm-identifier CirculationSignatureAlgorithmIdentifier,
        this-IPM             ThisIPMField,
        timestamp            CirculationTime } }

CirculationSignatureAlgorithmIdentifier ::= AlgorithmIdentifier

-- Circulation List Indicator

circulation-list-indicator  IPMS-EXTENSION ::= {
        VALUE                NULL,
        IDENTIFIED BY        id-rex-circulation-list-indicator }

-- Distribution Codes

distribution-codes IPMS-EXTENSION ::= {
        VALUE               DistributionCodes,
        IDENTIFIED BY       id-hex-distribution-codes }

DistributionCodes ::= SEQUENCE (SIZE (1..ub-distribution-codes)) OF DistributionCode

DistributionCode ::= SEQUENCE {
        oid-code             OBJECT IDENTIFIER OPTIONAL,
        alphanumeric-code    AlphaCode OPTIONAL,
        or-descriptor        ORDescriptor OPTIONAL }

AlphaCode ::= UniversalOrBMPString {ub-alpha-code-length}

-- Extended Subject

extended-subject IPMS-EXTENSION ::= {
        VALUE                ExtendedSubject,
        IDENTIFIED BY        id-hex-extended-subject }

ExtendedSubject ::= UniversalOrBMPString {ub-extended-subject-length}

-- Information category

information-category IPMS-EXTENSION ::= {
        VALUE                InformationCategories,
        IDENTIFIED BY        id-hex-information-category }

InformationCategories ::= SEQUENCE (SIZE (1..ub-information-categories))
        OF InformationCategory

InformationCategory ::= SEQUENCE {
        reference        [0] OBJECT IDENTIFIER OPTIONAL,
        description      [1] DescriptionString OPTIONAL }

DescriptionString ::= UniversalOrBMPString {ub-information-category-length}

-- Manual handling Instructions

manual-handling-instructions IPMS-EXTENSION ::= {
        VALUE                ManualHandlingInstructions,
        IDENTIFIED BY        id-hex-manual-handling-instructions }
```

```
ManualHandlingInstructions ::= SEQUENCE (SIZE (1..ub-manual-handling-instructions))
                                        OF ManualHandlingInstruction

ManualHandlingInstruction ::=
        UniversalOrBMPString {ub-manual-handling-instruction-length}

-- Originator's Reference

originators-reference IPMS-EXTENSION ::= {
        VALUE                   OriginatorsReference,
        IDENTIFIED BY           id-hex-originators-reference }

OriginatorsReference ::= UniversalOrBMPString {ub-originators-reference-length}

-- Precedence Policy Identifier

precedence-policy-identifier IPMS-EXTENSION ::= {
        VALUE                   PrecedencePolicyIdentifier,
        IDENTIFIED BY           id-hex-precedence-policy-id }

PrecedencePolicyIdentifier ::= OBJECT IDENTIFIER

-- Precedence

precedence IPMS-EXTENSION ::= {
        VALUE                   Precedence,
        IDENTIFIED BY           id-rex-precedence }

Precedence ::= INTEGER (0..ub-precedence)

END -- of IPMSHeadingExtensions
```

Annex I

Reference Definition of Message Store Attributes

(This annex forms an integral part of this Recommendation | International Standard)

This annex, a supplement to 19.6, defines for reference purposes the MS attributes and matching rules specific to Interpersonal Messaging. It uses the ATTRIBUTE information object class of ITU-T Rec. X.413 | ISO/IEC 10021-5.

```
IPMSMessageStoreAttributes { joint-iso-itu-t mhs(6) ipms(1) modules(0)
              message-store-attributes(8) version-1999(1) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS

MSString{}, mSStringMatch, mSSubstringsMatch

FROM MSMatchingRules {joint-iso-itu-t mhs(6) ms(4) modules(0) general-matching-rules(5)
         version-1999(1)}

       -- IPMS Heading Extensions

       AuthorizationTime, AutoSubmitted, BodyPartNumber, BodyPartSecurityLabel,
       BodyPartSignatures, CirculationMember, DistributionCode, ExtendedSubject,
       IncompleteCopy, InformationCategory, IPMSecurityLabel, Language,
       ManualHandlingInstruction, OriginatorsReference, Precedence, PrecedencePolicyIdentifier
               ----
               FROM IPMSHeadingExtensions { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      heading-extensions(6) version-1999(1) }

       -- IPMS Security Extensions

       BodyPartTokens, ForwardedContentToken
               ----
               FROM IPMSSecurityExtensions { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      ipm-security-extensions(14) version-1999(1) }

       -- IPMS Information Objects

       AcknowledgmentModeField, AuthorizingUsersSubfield, AutoForwardCommentField,
       AutoForwardedField, BilaterallyDefinedBodyPart, BlindCopyRecipientsSubfield, Body,
       ConversionEITsField, CopyRecipientsSubfield, DiscardReasonField, EncryptedBodyPart,
       EncryptedData, EncryptedParameters, ExpiryTimeField, EXTENDED-BODY-PART-TYPE,
       G3FacsimileBodyPart, G3FacsimileData, G3FacsimileParameters, G4Class1BodyPart, Heading,
       IA5TextBodyPart, IA5TextData, IA5TextParameters, ImportanceField, IPMIdentifier,
       IPMIntendedRecipientField, IPMSExtension { }, IPNOriginatorField, MessageBodyPart,
       MessageData, MessageParameters, MixedModeBodyPart, NationallyDefinedBodyPart,
       NonReceiptReasonField, NotificationExtensions,_NRNExtensions, ObsoletedIPMsSubfield,
       ORDescriptor, OriginatorField,  OtherNotifications, PrimaryRecipientsSubfield,
       ReceiptTimeField, RecipientSpecifier, RelatedIPMsSubfield, RepliedToIPMField,
       ReplyRecipientsSubfield, ReplyTimeField, ReturnedIPMField, RNExtensions,
       SensitivityField, SubjectField, SubjectIPMField, SupplReceiptInfoField,
       TeletexBodyPart, TeletexData, TeletexParameters, ThisIPMField, VideotexBodyPart,
       VideotexData, VideotexParameters
               ----
               FROM IPMSInformationObjects { joint-iso-itu-t mhs(6) ipms(1) modules(0)
                      information-objects(2) version-1999(1) }
```

```
-- IPMS Object Identifiers

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-cat-correlated-delivered-ipns, id-cat-correlated-delivered-replies,
id-cat-delivered-ipn-summary, id-cat-delivered-replies-summary, id-cat-forwarded-ipms,
id-cat-forwarding-ipms, id-cat-ipm-recipients, id-cat-obsoleted-ipms,
id-cat-obsoleting-ipms, id-cat-related-ipms, id-cat-relating-ipms,
id-cat-replied-to-ipm, id-cat-recipient-category, id-cat-replying-ipms,
id-cat-revised-reply-time, id-cat-subject-ipm, id-cat-submitted-ipn-status,
id-cat-submitted-ipns, id-cat-submitted-reply-status, id-hat-authorization-time,
id-hat-authorizing-users, id-hat-auto-forwarded, id-hat-auto-submitted,
id-hat-blind-copy-recipients, id-hat-body-part-encryption-token,
id-hat-body-part-security-label, id-hat-body-part-signature-verification-status,
id-hat-body-part-signatures, id-hat-circulation-list-recipients,
id-hat-copy-recipients, id-hat-distribution-codes, id-hat-expiry-time,
id-hat-extended-subject, id-hat-forwarded-content-token, id-hat-forwarding-token,
id-hat-heading, id-hat-importance, id-hat-incomplete-copy, id-hat-information-category,
id-hat-ipm-security-label, id-hat-languages, id-hat-manual-handling-instructions,
id-hat-nrn-requestors, id-hat-obsoleted-IPMs, id-hat-originator,
id-hat-originators-reference, id-hat-precedence, id-hat-precedence-policy-id,
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-mr-ipm-identifier, id-mr-ipm-location, id-mr-or-descriptor,
id-mr-or-descriptor-elements, id-mr-or-descriptor-single-element,
id-mr-or-descriptor-substring-elements, id-mr-circulation-member,
id-mr-circulation-member-checkmark, id-mr-circulation-member-elements,
id-mr-circulation-member-single-element, id-mr-circulation-member-substring-elements,
id-mr-distribution-code, id-mr-information-category, id-mr-recipient-specifier,
id-mr-recipient-specifier-elements, id-mr-recipient-specifier-single-element,
id-mr-recipient-specifier-substring-elements, id-nat-acknowledgment-mode,
id-nat-auto-forward-comment, id-nat-conversion-eits, id-nat-discard-reason,
id-nat-ipm-intended-recipient, id-nat-ipn-originator, id-nat-non-receipt-reason,
id-nat-notification-extensions, id-nat-nrn-extensions,
id-nat-other-notification-type-fields, id-nat-receipt-time, id-nat-returned-ipm,
id-nat-rn-extensions, id-nat-subject-ipm, id-nat-suppl-receipt-info,
id-sat-body-parts-summary, id-sat-ipm-auto-discarded, id-sat-ipm-entry-type,
id-sat-ipm-synopsis
        ----
        FROM IPMSObjectIdentifiers { joint-iso-itu-t mhs(6) ipms(1) modules(0)
               object-identifiers(0) version-1999(1) }

-- MS Abstract Service

ATTRIBUTE, MS-EIT, SequenceNumber
        ----
        FROM MSAbstractService { joint-iso-itu-t mhs(6) ms(4) modules(0)
               abstract-service(1) version-1999(1) }

-- MS General Attribute Types

SignatureStatus
        ----
        FROM MSGeneralAttributeTypes { joint-iso-itu-t mhs(6) ms(4) modules(0)
               general-attribute-types(2) version-1999(1) }

-- MS matching-rules

mSStringMatch, mSSubstringsMatch
        ----
        FROM MSMatchingRules { joint-iso-itu-t mhs(6) ms(4) modules(0)
               general-matching-rules(5) version-1999(1) }
```

```
       -- MTS Abstract Service

       EncodedInformationTypes, MessageToken
               ----
               FROM MTSAbstractService { joint-iso-itu-t mhs(6) mts(3) modules(0)
                      mts-abstract-service(1) version-1999(1) }

       -- Directory Information Framework

       objectIdentifierMatch, MATCHING-RULE
               ----
               FROM InformationFramework { joint-iso-itu-t ds(5) module(1)
                      informationFramework(1) 3 }

       -- Directory Abstract Service

       booleanMatch, generalizedTimeMatch, generalizedTimeOrderingMatch, integerMatch,
       integerOrderingMatch, uTCTimeMatch, uTCTimeOrderingMatch
               ----
               FROM SelectedAttributeTypes { joint-iso-itu-t ds(5) module(1)
                      selectedAttributeTypes(5) 3 };

-- IPMS attribute table information object set

IPMSAttributeTable ATTRIBUTE ::= {
       acknowledgment-mode | authorizing-users | auto-forward-comment | auto-forwarded |
       auto-submitted | bilaterally-defined-body-parts | blind-copy-recipients | body |
       conversion-eits | copy-recipients | discard-reason | encrypted-body-parts |
       encrypted-data | encrypted-parameters | expiry-time | extended-body-part-types |
       g3-facsimile-body-parts | g3-facsimile-data | g3-facsimile-parameters |
       g4-class1-body-parts | heading | ia5-text-body-parts | ia5-text-data |
       ia5-text-parameters | importance | incomplete-copy | ipm-entry-type |
       ipm-intended-recipient | ipm-synopsis | ipn-originator | languages |
       message-body-parts | message-data | message-parameters | mixed-mode-body-parts |
       nationally-defined-body-parts | non-receipt-reason | nrn-requestors | obsoleted-IPMs |
       originator | primary-recipients | receipt-time | related-IPMs | replied-to-IPM |
       reply-recipients | reply-requestors | reply-time | returned-ipm | rn-requestors |
       sensitivity | subject | subject-ipm | suppl-receipt-info | teletex-body-parts |
       teletex-data | teletex-parameters | this-ipm | videotex-body-parts | videotex-data |
       videotex-parameters,
       ...   -- 1994 extension additions --  ,
       ac-correlated-delivered-ipns | ac-correlated-delivered-replies |
       ac-delivered-ipn-summary | ac-delivered-replies-summary | ac-forwarded-ipms |
       ac-forwarding-ipms | ac-ipm-recipients | ac-obsoleted-ipms | ac-obsoleting-ipms |
       ac-related-ipms | ac-relating-ipms | ac-replied-to-ipm | ac-replying-ipms |
       ac-subject-ipm | ac-submitted-ipn-status | ac-submitted-ipns |
       ac-submitted-reply-status | authorization-time | body-part-encryption-token |
       body-part-security-label | body-part-signature-verification-status |
       body-part-signatures | body-parts-summary | circulation-list-recipients |
       distribution-codes | extended-subject | forwarded-content-token | forwarding-token |
       information-category | ipm-auto-discarded | ipm-security-label |
       manual-handling-instructions | notification-extensions | nrn-extensions |
       originators-reference | other-notification-type-fields | precedence |
       precedence-policy-identifier | recipient-category | revised-reply-time |
       rn-extensions }

-- SUMMARY ATTRIBUTES

-- IPM entry type

ipm-entry-type ATTRIBUTE ::= {
       WITH ATTRIBUTE-SYNTAX      IPMEntryType,
       EQUALITY MATCHING-RULE     integerMatch,
       NUMERATION                 single-valued,
       ID                         id-sat-ipm-entry-type }

IPMEntryType ::= ENUMERATED {
       ipm          (0),
       rn           (1),
       nrn          (2),
       on           (3) }
```

-- IPM synopsis

```
ipm-synopsis ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX     IPMSynopsis,
        NUMERATION                single-valued,
        ID                        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]  INSTANCE OF TYPE-IDENTIFIER OPTIONAL,
        size          [2]  INTEGER,
        processed     [3]  BOOLEAN DEFAULT FALSE}
```

-- Body part summary

```
body-parts-summary ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX     BodyPartDescriptor,
        NUMERATION                multi-valued,
        ID                        id-sat-body-parts-summary }

BodyPartDescriptor ::= SEQUENCE {
        data              [0]  OBJECT IDENTIFIER,
        parameters        [1]  OBJECT IDENTIFIER OPTIONAL,
        this-child-entry  [2]  SequenceNumber OPTIONAL,
        position          [3]  INTEGER,
        size              [4]  INTEGER,
        processed         [5]  BOOLEAN DEFAULT FALSE }
```

-- IPM auto discarded

```
ipm-auto-discarded ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX     BOOLEAN,
        EQUALITY MATCHING-RULE    booleanMatch,
        NUMERATION                single-valued,
        ID                        id-sat-ipm-auto-discarded }
```

-- Body part signature verification status

```
body-part-signature-verification-status  ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX     BodyPartSignatureVerification,
        NUMERATION                single-valued,
        ID                        id-hat-body-part-signature-verification-status }

BodyPartSignatureVerification ::= SET OF SET {
        body-part-sequence-number BodyPartNumber,
        body-part-signature       SignatureStatus }
```

-- HEADING ATTRIBUTES

-- Heading

```
heading ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX     Heading,
        NUMERATION                single-valued,
        ID                        id-hat-heading }
```

-- Heading analyses

```
rn-requestors ATTRIBUTE ::= {
        WITH ATTRIBUTE-SYNTAX     ORDescriptor,
        EQUALITY MATCHING-RULE    oRDescriptorMatch,
        NUMERATION                multi-valued,
        ID                        id-hat-rn-requestors }
```