

INTERNATIONAL STANDARD

ISO/IEC
10021-7

First edition
1990-12-01

Information technology — Text Communication — Message-Oriented Text Interchange Systems (MOTIS) —

Part 7: Interpersonal Messaging System

*Technologies de l'information — Communication de texte — Systèmes d'échange
de texte en mode message —*

Partie 7. Système de messagerie de personne à personne



Reference number
ISO/IEC 10021-7 : 1990 (E)

Contents

Foreword	ix
Introduction	x
Section one - Introduction	1
1 Scope	1
2 Normative references	1
2.1 Open Systems Interconnection	1
2.2 Message Handling Systems	1
2.3 Directory Systems	2
2.4 Language Code	2
2.5 Character Sets	2
2.6 Telematic Services	2
3 Definitions	2
4 Abbreviations	3
5 Conventions	3
5.1 ASN.1	3
5.2 Grade	4
5.3 Terms	4
Section two - Abstract Information Objects	5
6 Overview	5
7 Interpersonal Messages	5
7.1 Heading Field Component Types	6
7.1.1 IPM Identifier	6
7.1.2 Recipient Specifier	7
7.1.3 O/R Descriptor	7
7.2 Heading Fields	8
7.2.1 This IPM	8
7.2.2 Originator	9
7.2.3 Authorizing Users	9
7.2.4 Primary Recipients	9
7.2.5 Copy Recipients	9
7.2.6 Blind Copy Recipients	10
7.2.7 Replied-to IPM	10
7.2.8 Obsolete IPMs	10
7.2.9 Related IPMs	10
7.2.10 Subject	11
7.2.11 Expiry Time	11

© ISO/IEC 1990

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

International Organization for Standardization
Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

7.2.12	Reply Time	11
7.2.13	Reply Recipients	11
7.2.14	Importance	11
7.2.15	Sensitivity	12
7.2.16	Auto-forwarded	12
7.2.17	Extensions	12
7.3	Body Part Types	13
7.3.1	IA5 Text	13
7.3.2	Voice	14
7.3.3	G3 Facsimile	14
7.3.4	G4 Class 1	15
7.3.5	Teletex	15
7.3.6	Videotex	15
7.3.7	Encrypted	16
7.3.8	Message	16
7.3.9	Mixed-mode	17
7.3.10	Bilaterally Defined	17
7.3.11	Nationally Defined	17
7.3.12	Externally Defined	18
8	Interpersonal Notifications	19
8.1	Common Fields	20
8.1.1	Subject IPM	20
8.1.2	IPN Originator	20
8.1.3	IPM Preferred Recipient	21
8.1.4	Conversion EITs	21
8.2	Non-receipt Fields	21
8.2.1	Non-receipt Reason	21
8.2.2	Discard Reason	21
8.2.3	Auto-forward Comment	22
8.2.4	Returned IPM	22
8.3	Receipt Fields	22
8.3.1	Receipt Time	22
8.3.2	Acknowledgment Mode	23
8.3.3	Suppl Receipt Info	23
Section three - Abstract service Definition		24
9	Overview	24
10	Primary Object Types	24
10.1	Interpersonal Messaging System User	25
10.2	Interpersonal Messaging System	25
11	Primary Port Types	26
11.1	Origination	26
11.2	Reception	26
11.3	Management	26
12	Abstract Operations	26
12.1	Origination Abstract Operations	27
12.1.1	Originate Probe	27
12.1.2	Originate IPM	27
12.1.3	Originate RN	28
12.2	Reception Abstract Operations	29
12.2.1	Receive Report	29
12.2.2	Receive IPM	30
12.2.3	Receive RN	30
12.2.4	Receive NRN	30
12.3	Management Abstract Operations	31
12.3.1	Change Auto-discard	31
12.3.2	Change Auto-acknowledgment	31
12.3.3	Change Auto-forwarding	32

13	Abstract Errors	32
13.1	Subscription Error.....	33
13.2	Recipient Improperly Specified.....	33
14	Other Capabilities.....	33
Section four - Abstract Service Provision		34
15	Overview	34
16	Secondary Object Types	34
16.1	Interpersonal Messaging System User Agent.....	36
16.2	Interpersonal Messaging System Message Store.....	36
16.3	Telematic Agent.....	36
16.4	Telex Access Unit.....	37
16.5	Physical Delivery Access Unit.....	37
16.6	Message Transfer System	37
17	Secondary Port Types.....	37
17.1	Submission.....	38
17.2	Delivery.....	38
17.3	Retrieval.....	38
17.4	Administration	38
17.5	Import.....	38
17.6	Export.....	38
18	User Agent Operation.....	38
18.1	State Variables.....	39
18.2	Performance of Origination Operations	39
18.2.1	Originate Probe	39
18.2.2	Originate IPM.....	40
18.2.3	Originate RN.....	40
18.3	Performance of Management Operations.....	41
18.3.1	Change Auto-discard.....	41
18.3.2	Change Auto-acknowledgment	42
18.3.3	Change Auto-forwarding	42
18.4	Invocation of Reception Operations.....	42
18.4.1	Receive Report.....	42
18.4.2	Receive IPM.....	43
18.4.3	Receive RN.....	43
18.4.4	Receive NRN	43
18.5	Internal Procedures.....	43
18.5.1	Auto-discard.....	44
18.5.1.1	Discard of IPM	44
18.5.1.2	Construction of NRN	44
18.5.1.3	Submission of NRN	44
18.5.2	Auto-acknowledgment.....	45
18.5.2.1	Construction of RN.....	45
18.5.2.2	Submission of RN.....	45
18.5.3	Auto-forwarding.....	45
18.5.3.1	Prevention of Loops	46
18.5.3.2	Construction of IPM.....	46
18.5.3.3	Submission of IPM.....	46
18.5.3.4	Construction of NRN	46
18.5.3.5	Submission of NRN.....	47
19	Message Store Operation.....	47
19.1	Creation of Information Objects	47
19.2	Maintenance of Attributes	47
19.3	Notification of Non-receipt.....	48
19.4	Auto-forwarding.....	48
19.5	Manual Forwarding	48

20	Message Contents	49
20.1	Content	49
20.2	Content Type	49
20.3	Content Length	50
20.4	Encoded Information Types	50
21	Port Realization	51
22	Conformance	51
22.1	Origination Versus Reception	51
22.2	Statement Requirements	51
22.3	Static Requirements	51
22.4	Dynamic Requirements	52
Annexes		53
A	Heading Extensions	53
A.1	Incomplete Copy	53
A.2	Languages	53
B	Extended Body Part Types	54
B.1	Equivalents of Basic Body Part Types	54
B.2	General Text	55
C	Message Store Attributes	56
C.1	Summary Attributes	58
C.1.1	IPM Entry Type	58
C.1.2	IPM Synopsis	59
C.2	Heading Attributes	60
C.2.1	Heading	60
C.2.2	Heading Analyses	60
C.2.3	Heading Fields	61
C.2.4	Heading Sub-fields	62
C.2.5	Heading Extensions	62
C.3	Body Attributes	63
C.3.1	Body	63
C.3.2	Basic Body Parts	63
C.3.3	Basic Body Part Parameters Components	64
C.3.4	Basic Body Part Data Components	65
C.3.5	Extended Body Part Types	65
C.3.6	Extended Body Parts	65
C.4	Notification Attributes	66
C.4.1	Common Fields	66
C.4.2	Non-receipt Fields	66
C.4.3	Receipt Fields	67
D	Reference Definition of Object Identifiers	68
E	Reference Definition of Abstract Information Objects	72
F	Reference Definition of Functional Objects	79
G	Reference Definition of Abstract Service	82
H	Reference Definition of Heading Extensions	85
I	Reference Definition of Extended Body Part Types	86
I.1	Equivalents of Basic Body Part Types	86
I.2	General Text	87
J	Reference Definition of Message Store Attributes	89

K Reference Definition of Upper Bounds..... 97

L Support of the Interpersonal Messaging Service 98

 L.1 Support of Recipient Specifier Components..... 98

 L.2 Support of Heading Fields 99

 L.3 Support of Body Aspects..... 99

M Differences Between ISO/IEC 10021-7 and CCITT Recommendation X.420 100

N Summary of Changes to 1984 CCITT Specification 101

O Index 102

IECNORM.COM : Click to view the full PDF of ISO/IEC 10021-7:1990

List of Figures

1 An Interpersonal Message..... 6

2 An Interpersonal Notification 20

3 The Interpersonal Messaging Environment 25

4 The Interpersonal Messaging System 35

IECNORM.COM : Click to view the full PDF of ISO/IEC 10021-7:1990

List of Tables

1 Uses of the ASN.1 Notation.....3

2 Interpersonal Messaging Basic EITs and NBPs 50

C.1 Summary of MS Attributes 57-58

L.1 Support of Recipient Specifier Components..... 98

L.2 Support of Heading Fields 99

L.3 Support of Body Aspects..... 99

IECNORM.COM : Click to view the full PDF of ISO/IEC 10021-7:1990

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. 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.

International Standard ISO/IEC 10021-7 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

ISO/IEC 10021-7 consists of the following parts, under the general title: *Information technology — Text Communication — Message-Oriented Text Interchange Systems (MOTIS)* —

- *Part 1: System and Service Overview*
- *Part 2: Overall Architecture*
- *Part 3: Abstract Service Definition Conventions*
- *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*

Annexes A, B, C, D, E, F, G, H, I, J and L form an integral part of this part of ISO/IEC 10021. Annexes K, M, N and O are for information only.

Introduction

This part of ISO/IEC 10021 is one of a number of parts of ISO/IEC 10021 (the International Standards for Message-Oriented Text Interchange Systems (MOTIS)). ISO/IEC 10021 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 part of ISO/IEC 10021 defines the Message Handling application called *Interpersonal Messaging*, specifying in the process the message content type and associated procedures known as P2.

The text of this part of ISO/IEC 10021 is the subject of joint CCITT-ISO agreement. The corresponding CCITT specification is Recommendation X.420.

IECNORM.COM : Click to view the full PDF of ISO/IEC 10021-7:1990

Information technology - Text Communication - Message-Oriented Text Interchange Systems (MOTIS) - Part 7 : Interpersonal Messaging System

Section one - Introduction

1 Scope

This part of ISO/IEC 10021 defines **Interpersonal Messaging**, a form of Message Handling tailored for ordinary interpersonal business or private correspondence.

This part of ISO/IEC 10021 is one of a series on Message Handling. 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 International Standards. ISO/IEC 10021-2 identifies those documents as well.

This part of ISO/IEC 10021 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 part of ISO/IEC 10021 are given in clause 22.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 10021. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 10021 are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of ISO and IEC maintain registers of currently valid International Standards.

2.1 Open Systems Interconnection

This part of ISO/IEC 10021 cites the following OSI specification:

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

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

2.2 Message Handling Systems

This part of ISO/IEC 10021 cites the following Message Handling System specifications:

ISO/IEC 10021:1990, *Information technology - Text communication - Message-Oriented Text Interchange Systems (MOTIS) -*

Part 1: Service and system overview.

Part 2: Overall architecture.

Part 3: Abstract service definition conventions.

Part 4: Message transfer system : Abstract service definition and procedures.

Part 5: Message store : Abstract service definition.

Part 6: Protocol specifications.

CCITT X.408:1988 *Message handling systems: Encoded information type conversion rules.*

CCITT X.420:1984 *Message handling systems: Interpersonal messaging user agent layer.*

2.3 **Directory Systems**

This part of ISO/IEC 10021 cites the following Directory System specification:

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

2.4 **Language Code**

This part of ISO/IEC 10021 cites the following Language Code specification:

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

2.5 **Character Sets**

This part of ISO/IEC 10021 cites the following Character Set specifications:

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

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

2.6 **Telematic Services**

This part of ISO/IEC 10021 cites the following Telematic Service specifications:

CCITT T.4:1988, *Standardization of group 3 facsimile apparatus for document transmission.*

CCITT T.30:1988, *Procedures for document facsimile transmission in the general switched telephone network.*

CCITT T.100:1988, *International information exchange for interactive videotex.*

CCITT T.101:1988, *International interworking for videotex services.*

CCITT T.330:1988, *Telematic access to IPMS.*

3 **Definitions**

For the purposes of this part of ISO/IEC 10021, the definitions given in ISO/IEC 10021-2 apply.

4 Abbreviations

For the purposes of this part of ISO/IEC 10021, the abbreviations given in ISO/IEC 10021-2 apply.

5 Conventions

This part of ISO/IEC 10021 uses the descriptive conventions identified below.

5.1 ASN.1

This part of ISO/IEC 10021 uses for the indicated purposes the following ASN.1-based descriptive conventions:

- a) To define the information objects of Interpersonal Messaging, and other data types and values of all kinds, ASN.1 itself.
- b) To define the functional objects of Interpersonal Messaging, the OBJECT and REFINE macros of ISO/IEC 10021-3.
- c) To define the abstract service of Interpersonal Messaging, the PORT and ABSTRACT-OPERATION and -ERROR macros of ISO/IEC 10021-3.
- d) To define the *heading extensions*, the HEADING-EXTENSION macro of clause 7.2.17.
- e) To define *extended body part types*, the EXTENDED-BODY-PART-TYPE macro of clause 7.3.12.
- f) To define MS attributes, the ATTRIBUTE macro of ISO/IEC 9594-2.

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 part of ISO/IEC 10021 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 D
Abstract information objects	section two	annex E
Functional objects	clauses 10, 11, 16	annex F
Abstract service	clauses 12-13	annex G
Heading extensions	annex A	annex H
Extended body part types	annex B	annex I
Message store attributes	annex C	annex J
Upper bounds	-	annex K

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

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

NOTES

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.

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

5.2 Grade

This part of ISO/IEC 10021 uses the concept of grade as developed in ISO/IEC 10021-2.

5.3 Terms

Throughout this part of ISO/IEC 10021, 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.

IECNORM.COM : Click to view the full PDF of ISO/IEC 10021-7:1990

Section two - 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.

NOTES

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

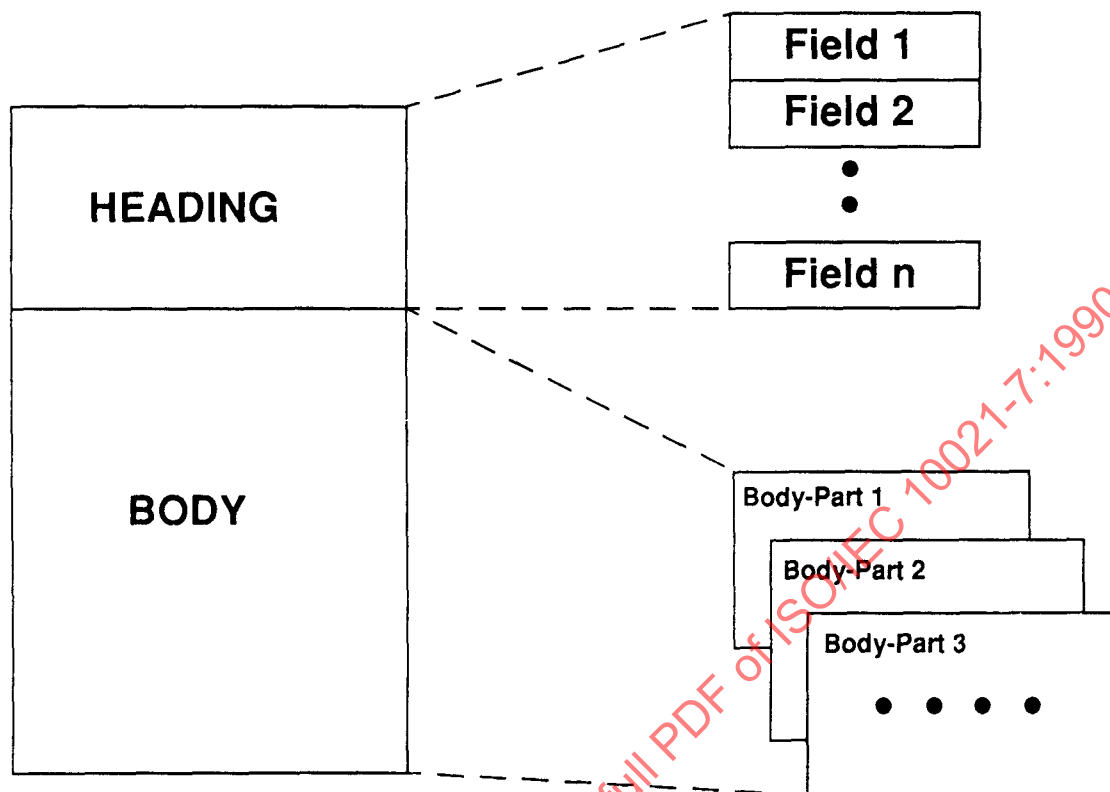


Figure 1
An Interpersonal Message

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*, and *O/R descriptor*--are defined and described below.

7.1.1 IPM Identifier

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

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

An IPM identifier has the following components:

- a) **User (O):** Identifies the user who originates the IPM. One of the user's O/R addresses. 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. 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 part of ISO/IEC 10021 assigns.

7.1.2 Recipient Specifier

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

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

A recipient specifier has the following components:

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

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

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

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

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

- i) *rn*: A *receipt notification* is requested in the circumstances prescribed in clause 8.
 - ii) *nrn*: A *non-receipt notification* is requested in the circumstances prescribed in clause 8.
 - iii) *ipm-return*: It is requested that the IPM be returned in any *non-receipt notification*.
- c) **Reply-requested (D false)**: Indicates whether a reply is requested of the preferred recipient denoted by the Recipient component. A Boolean.

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

7.1.3 O/R Descriptor

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

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

An O/R descriptor has the following components:

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

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

- i) The *Free-form-name* component is absent.
- ii) The O/R descriptor appears in the *Reply Recipients* heading field.
- iii) The O/R descriptor is the Recipient component of a recipient specifier and the conditions stated in item a of clause 7.1.2 are satisfied.

- b) **Free-form-name (O)**: Identifies the user or DL in question. A Teletex String, 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, chosen from the graphical subset of the Printable String character set. A length of zero is discouraged.

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

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

7.2 Heading Fields

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

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

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

7.2.1 This IPM

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

ThisIPMField ::= IPMIdentifier

7.2.2 Originator

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

OriginatorField ::= ORDescriptor

7.2.3 Authorizing Users

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

AuthorizingUsersField ::= SEQUENCE OF AuthorizingUsersSubfield

AuthorizingUsersSubfield ::= ORDescriptor

An **authorizing user** is a user who, either individually or in concert with others, authorizes the origination of an IPM. The word "authorizes" above is not precisely defined by this part of ISO/IEC 10021; it is given meaning by users.

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

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

7.2.4 Primary Recipients

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

PrimaryRecipientsField ::= SEQUENCE OF PrimaryRecipientsSubfield

PrimaryRecipientsSubfield ::= RecipientSpecifier

The phrase "primary recipients" above is not precisely defined by this part of ISO/IEC 10021; 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 part of ISO/IEC 10021; it is given meaning by users.

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

7.2.6 Blind Copy Recipients

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

BlindCopyRecipientsField ::= SEQUENCE OF BlindCopyRecipientsSubfield

BlindCopyRecipientsSubfield ::= RecipientSpecifier

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

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

7.2.7 Replied-to IPM

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

RepliedToIPMField ::= IPMIdentifier

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

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

7.2.8 Obsolete IPMs

The **Obsolete 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.

ObsoleteIPMsField ::= SEQUENCE OF ObsoleteIPMsSubfield

ObsoleteIPMsSubfield ::= 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 part of ISO/IEC 10021; it is given meaning by users.

NOTES

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

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, chosen from the graphic subset of the Teletex String character set. A length of zero is discouraged.

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

7.2.11 Expiry Time

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

ExpiryTimeField ::= Time

7.2.12 Reply Time

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

ReplyTimeField ::= Time

7.2.13 Reply Recipients

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

ReplyRecipientsField ::= SEQUENCE OF ReplyRecipientsSubfield

ReplyRecipientsSubfield ::= ORDescriptor

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

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

7.2.14 Importance

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

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

The values above are not defined by this part of ISO/IEC 10021; they are given meaning by users.

7.2.15 Sensitivity

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

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

This field may assume any one of the following values:

- a) *personal*: The IPM is conveyed to its preferred recipients as individuals, rather than in their professional capacities.
- b) *private*: The IPM should be conveyed to no one other than its preferred recipients.
- c) *company-confidential*: The IPM contains information that should be handled according to company-specific procedures.

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

7.2.16 Auto-forwarded

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

```
AutoForwardedField ::= BOOLEAN
```

7.2.17 Extensions

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

```
ExtensionsField ::= SET OF HeadingExtension

HeadingExtension ::= SEQUENCE (
    type OBJECT IDENTIFIER,
    value ANY DEFINED BY type DEFAULT NULL NULL)
```

Each extension has the following components:

- a) **Type** (M): Identifies the semantics and restricts the abstract syntax of the *Value* component. An Object Identifier.
- b) **Value** (D null): An information item whose abstract syntax is restricted only by the Type component. An Any.

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

All extensions are defined in annex A. Thus each extension's Type component shall have one of the values given in that annex. An extension whose Type component has another value shall be ignored.

Every extension is defined by means of the following macro.

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

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

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

NOTE - Future addenda to this part of ISO/IEC 10021 may define additional extensions. Furthermore, future addenda are likely to add information to the Heading only by means of this field.

7.3 Body Part Types

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

```
BodyPart ::= CHOICE {
    ia5-text          [0] IA5TextBodyPart,
    voice             [2] VoiceBodyPart,
    g3-facsimile      [3] G3FacsimileBodyPart,
    g4-class1         [4] G4Class1BodyPart,
    teletex           [5] TeletexBodyPart,
    videotex          [6] VideotexBodyPart,
    encrypted         [8] EncryptedBodyPart,
    message           [9] MessageBodyPart,
    mixed-mode        [11] MixedModeBodyPart,
    bilaterally-defined [14] BilaterallyDefinedBodyPart,
    nationally-defined  [7] NationallyDefinedBodyPart,
    externally-defined [15] ExternallyDefinedBodyPart}
```

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

NOTES

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. These tags, therefore, are avoided in BodyPart.

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

7.3.1 IA5 Text

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

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

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

IA5TextData ::= IA5String
```

The Parameters component comprises the following parameters:

- 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).

7.3.2 Voice

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

```
VoiceBodyPart ::= SEQUENCE (
    parameters VoiceParameters,
    data       VoiceData)

VoiceParameters ::= SET -- for future standardization

VoiceData ::= BIT STRING -- for future standardization
```

The parameters of such a body part, and the digitized speech encoding technique that those parameters might identify and parameterize, may be the subject of future standardization.

The Data component is the speech, a Bit String.

7.3.3 G3 Facsimile

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

```
G3FacsimileBodyPart ::= SEQUENCE (
    parameters G3FacsimileParameters,
    data       G3FacsimileData)

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

G3FacsimileData ::= SEQUENCE OF BIT STRING
```

The Parameters component comprises the following parameters:

- a) **Number-of-pages** (O): Identifies the number of pages of Group 3 facsimile data present in the Data component. A non-negative Integer.
- b) **Non-basic-parameters** (C): Identifies the non-basic parameters (NBPs) for Group 3 facsimile that characterize the Data component. A G3 NBPs descriptor.

This conditional parameter shall be present if (but not only if) the Body contains two or more G3 Facsimile body parts.

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 CCITT Recommendations T.4 and T.30.

NOTES

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

7.3.4 G4 Class 1

A **G4 Class 1** body part represents a final-form document of the sort that is processable by Group 4 Class 1 facsimile terminals. It comprises a Sequence of protocol elements which describe the document's layout structure.

```
G4Class1BodyPart ::= SEQUENCE OF ProtocolElement
```

7.3.5 Teletex

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

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

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

```
TeletexData ::= SEQUENCE OF TeletexString
```

The Parameters component comprises the following parameters:

- a) **Number-of-pages** (O): Identifies the number of pages of Teletex text present in the Data component. A non-negative Integer.
- b) **Telex-compatible** (D *false*): Indicates whether the document in the Data component is telex-compatible. A Boolean.

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

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

This conditional parameter shall be present if (but not only if) the Body contains two or more Teletex body parts.

The Data component is the document, a Sequence of Teletex Strings, each of which encodes one of its pages.

NOTES

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

7.3.6 Videotex

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

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

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

```
VideotexData ::= VideotexString
```

The Parameters component comprises the following parameters:

- 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-syntax2*: Data Syntax 2.
- iv) *data-syntax3*: Data Syntax 3.

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

7.3.7 Encrypted

An **Encrypted** body part represents the result of encrypting a body part of a type defined by this part of ISO/IEC 10021. It has Parameters and Data components.

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

```
EncryptedParameters ::= SET -- for future standardization
```

```
EncryptedData ::= BIT STRING -- for future standardization
```

The parameters of such a body part, and the encryption technique that those parameters might identify and parameterize, may be the subject of future standardization.

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 ISO 8825.

7.3.8 Message

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

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

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

```
MessageData ::= IPM
```

The Parameters component comprises the following parameters:

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

The Data component is the IPM.

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

NOTES

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).

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

7.3.9 Mixed-mode

A **Mixed-mode** body part represents a final-form document of the sort that is processable by mixed-mode Teletex terminals and Group 4 Classes 2 and 3 facsimile terminals. It comprises a Sequence of protocol elements which describe the document's layout structure.

MixedModeBodyPart ::= SEQUENCE OF ProtocolElement

7.3.10 Bilaterally Defined

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

BilaterallyDefinedBodyPart ::= OCTET STRING

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

7.3.11 Nationally Defined

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

NationallyDefinedBodyPart ::= ANY

NOTES

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.

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

7.3.12 Externally Defined

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

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

```
ExternallyDefinedParameters ::= EXTERNAL
```

```
ExternallyDefinedData ::= EXTERNAL
```

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

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

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

All basic body part types are defined in this part of ISO/IEC 10021.

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

Some (but not necessarily all) extended body part types are defined in annex B of this part of ISO/IEC 10021.

Every extended body part type this part of ISO/IEC 10021 defines is defined by means of the following macro. Every extended body part type defined elsewhere shall be so defined as well.

```
EXTENDED-BODY-PART-TYPE MACRO ::=
BEGIN
    TYPE NOTATION ::= Parameters Data
    VALUE NOTATION ::= value (VALUE OBJECT IDENTIFIER)

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

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

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

NOTES

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

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

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

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

8 Interpersonal Notifications

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

```
IPN ::= SET (
    -- common-fields -- COMPONENTS OF CommonFields,
    choice [0] CHOICE (
        non-receipt-fields [0] NonReceiptFields,
        receipt-fields      [1] ReceiptFields))
```

An IPN may take either of the following forms:

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

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

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

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

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

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

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

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

- a) **common field**: A notification field applicable to both NRNs and RNs.
- b) **non-receipt field**: A notification field applicable to NRNs alone.
- c) **receipt field**: A notification field applicable to RNs alone.

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-preferred-recipient     [2] IPMPreferredRecipientField OPTIONAL,
    conversion-eits             ConversionEITsField OPTIONAL}
```

8.1.1 Subject IPM

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

```
SubjectIPMField ::= IPMIdentifier
```

8.1.2 IPN Originator

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

```
IPNOriginatorField ::= ORDescriptor
```

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

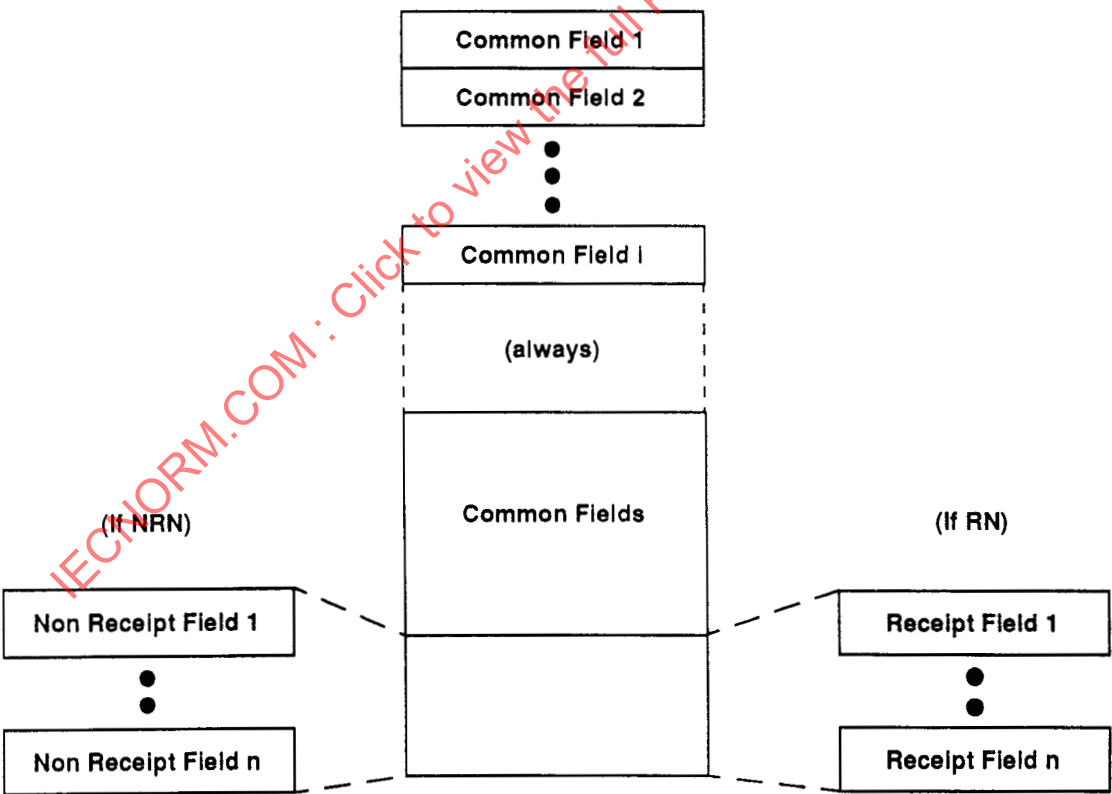


Figure 2
An Interpersonal Notification

8.1.3 IPM Preferred Recipient

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

IPMPreferredRecipientField ::= ORDescriptor

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

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

8.1.4 Conversion EITs

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

ConversionEITsField ::= EncodedInformationTypes

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

8.2 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)
```

8.2.1 Non-receipt Reason

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

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

This field may assume any one of the following values:

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

8.2.2 Discard Reason

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

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

This field may assume any one of the following values:

- a) *ipm-expired: Auto-discard* was in effect, expired IPMs were being discarded, and the time identified by the subject IPM's Expiry Time heading field had arrived.
- b) *ipm-obsolete: Auto-discard* was in effect, obsolete IPMs were being discarded, and the Obsolete IPMs heading field of another IPM, delivered to the NRN's originator, identified the subject IPM.
- c) *user-subscription-terminated*: The Interpersonal Messaging subscription of the NRN's originator was terminated.

This conditional field shall be present only if the Non-receipt Reason field has the value *ipm-discarded*.

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. 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.

8.3 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 DEFAULT ""}
```

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.

```
SupplReceiptInfoField ::= SupplementaryInformation
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 10021-7:1990

Section three - 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 ISO/IEC 10021-3.

This section covers the following topics:

- a) Primary object types;
- b) Primary port types;
- c) Abstract operations;
- d) Abstract errors;
- e) Other capabilities.

10 Primary Object Types

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

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

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

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

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

The structure of the IPME is depicted in Figure 3.

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

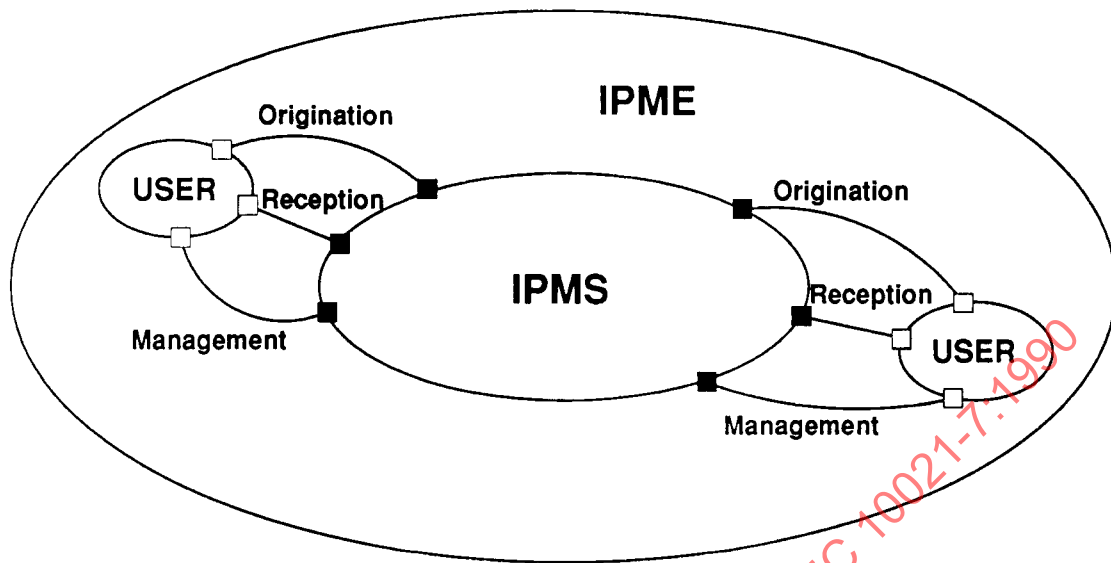


Figure 3
The Interpersonal Messaging Environment

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

The IPME comprises any number of IPMS users.

NOTES

1. As its name suggests, Interpersonal Messaging is typically an activity of people. Often, therefore, this part of ISO/IEC 10021 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.
2. For brevity, the term "user" is used throughout the rest of this part of ISO/IEC 10021 with the meaning of "IPMS user".

10.2 Interpersonal Messaging System

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

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

The IPME comprises exactly one IPMS.

11 Primary Port Types

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

NOTE - In clause 16 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 clause 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 clause 16.5).

12 Abstract Operations

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

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

NOTES

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

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 {
    OriginateProbe,
    OriginateIPM,
    OriginateRN}
  ::= id-pt-origination
```

12.1.1 Originate Probe

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

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

This abstract operation has the following arguments:

- a) **Envelope (M)**: A probe submission envelope, whose make-up the MTS Abstract Service defines. The UA supplies all but the following envelope components, which the user provides:
 - i) The desired per-message options (i.e., per-message indicators and extensions).
 - ii) The O/R names of the preferred recipients and the per-recipient options (i.e., originator report request, explicit conversion, and extensions) desired for each.
- b) **Content (M)**: An instance of the class of IPM whose deliverability is to be probed.

This abstract operation has the following results:

- 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.

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

This abstract operation has the following arguments:

- a) **Envelope (M):** A message submission envelope, whose make-up the MTS Abstract Service defines. The UA supplies all but the following envelope components, which the user provides:
 - i) The desired per-message options (i.e., priority, per-message indicators, deferred delivery time, and extensions).
 - ii) The O/R names of the preferred recipients and the per-recipient options (i.e., originator report request, explicit conversion, and extensions) desired for each.
- b) **Content (M):** The IPM being originated. Its Auto-forwarded heading field shall be absent or have the value *false*.

This abstract operation has the following results:

- 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.

```

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

```

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

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

This abstract operation has the following arguments:

- a) **Envelope (M):** A message submission envelope, whose make-up the MTS Abstract Service defines. The UA supplies all but the following envelope components, which the user provides:
 - i) The desired per-message options (i.e., priority, per-message indicators, and extensions). Implicit conversion shall be prohibited, priority that of the subject IPM.
 - ii) The O/R names of the preferred recipients and the per-recipient options (i.e., explicit conversion and extensions) desired for each. Reports shall not be requested.
- 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 the message was directly submitted.

12.2 Reception Abstract Operations

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

```
reception PORT
  SUPPLIER INVOKES (
    ReceiveReport,
    ReceiveIPM,
    ReceiveRN,
    ReceiveNRN)
  ::= id-pt-reception
```

NOTES

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.

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

12.2.1 Receive Report

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

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

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

- a) A probe concerning a message whose content was an IPM that was originated with the Originate Probe abstract operation.
- b) A message whose content was an NRN that was originated as a result of *auto-discard* of *auto-forward*.
- c) A message whose content was an RN that was originated with the Originate RN abstract operation or by *auto-acknowledgment*.
- d) A message whose content was an IPM that was originated with the Originate IPM abstract operation or by *auto-forwarding*.

This abstract operation has the following arguments:

- 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.

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

This abstract operation has the following arguments:

- a) **Envelope (M)**: The message's delivery envelope.
- b) **Content (M)**: The IPM that is the message's content.

This abstract operation has no results.

12.2.3 Receive RN

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

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

This abstract operation has the following arguments:

- a) **Envelope (M)**: The message's delivery envelope.
- b) **Content (M)**: The RN that is the message's content.

This abstract operation has no results.

12.2.4 Receive NRN

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

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

This abstract operation has the following arguments:

- a) **Envelope (M)**: The message's delivery envelope.
- b) **Content (M)**: The NRN that is the message's content.

This abstract operation has no results.

12.3 Management Abstract Operations

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

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

12.3.1 Change Auto-discard

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

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

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

This abstract operation has the following arguments:

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

This abstract operation has no results.

12.3.2 Change Auto-acknowledgment

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

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

This abstract operation has the following arguments:

- Auto-acknowledge-IPMs (M):** Whether or not IPMs are to be auto-acknowledged. A Boolean.
- Auto-acknowledge-suppl-receipt-info (C):** The Suppl Receipt Info receipt field of each RN provoked by auto-acknowledgment.

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

This abstract operation has no results.

12.3.3 Change Auto-forwarding

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

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

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

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

This abstract operation has the following arguments:

- a) **Auto-forward-IPMs** (M): Whether or not IPMs are to be auto-forwarded. A Boolean.
- b) **Auto-forward-recipients** (C): The users or DLs to which IPMs are to be auto-forwarded. A Sequence of O/R names.

This conditional argument shall be present if, and only if, the Auto-forward-IPMs argument has the value *true*.
- c) **Auto-forward-heading** (C): The Heading that is to be used for each forwarding IPM. Its Auto-forwarded heading field shall have the value *true*.

This conditional argument shall be present if, and only if, the Auto-forward-IPMs argument has the value *true*.
- d) **Auto-forward-comment** (C): The value that is to be supplied as the Auto-forward Comment non-receipt field of each NRN conveyed to the originator of an auto-forwarded IPM.

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

This abstract operation has no results.

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

13 Abstract Errors

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

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

13.1 Subscription Error

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

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

This abstract error has the following parameters:

- 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 O/R names supplied as arguments of the abstract operation whose performance is aborted, or as components of its arguments, are invalid.

This abstract error is defined by the MTS Abstract Service.

14 Other Capabilities

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

The following additional capabilities shall be provided:

- a) *Submission*: Capabilities of the MS' or MTS' submission port not embodied in the IPMS Abstract Service, e.g., the ability to cancel delivery of a previously originated message whose content is an IPM (but not an RN), if deferred delivery was selected.
- b) *Delivery*: Capabilities of the MTS' delivery port not embodied in the IPMS Abstract Service, e.g., the ability to temporarily control the kinds of information objects the MTS conveys to the user's UA.
- c) *Administration*: The capabilities of the MS's or MTS's administration port.
- d) *Retrieval*: The capabilities of the MS' retrieval port.

In addition to the above and as a local matter, the IPMS may provide to users additional capabilities neither defined nor limited by this part of ISO/IEC 10021. 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 four - Abstract Service Provision

15 Overview

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

This section covers the following topics:

- a) Secondary object types;
- b) Secondary port types;
- c) User agent operation;
- d) Message store operation;
- e) Message contents;
- f) Port realization;
- g) Conformance.

16 Secondary Object Types

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

```

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

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

The structure of the IPMS is depicted in Figure 4. As shown by the figure, *IPMS UAs*, *TLMA*s, *TLXAU*s, and *PDAU*s are the instruments by means of which the IPMS provides the IPMS Abstract Service to users.

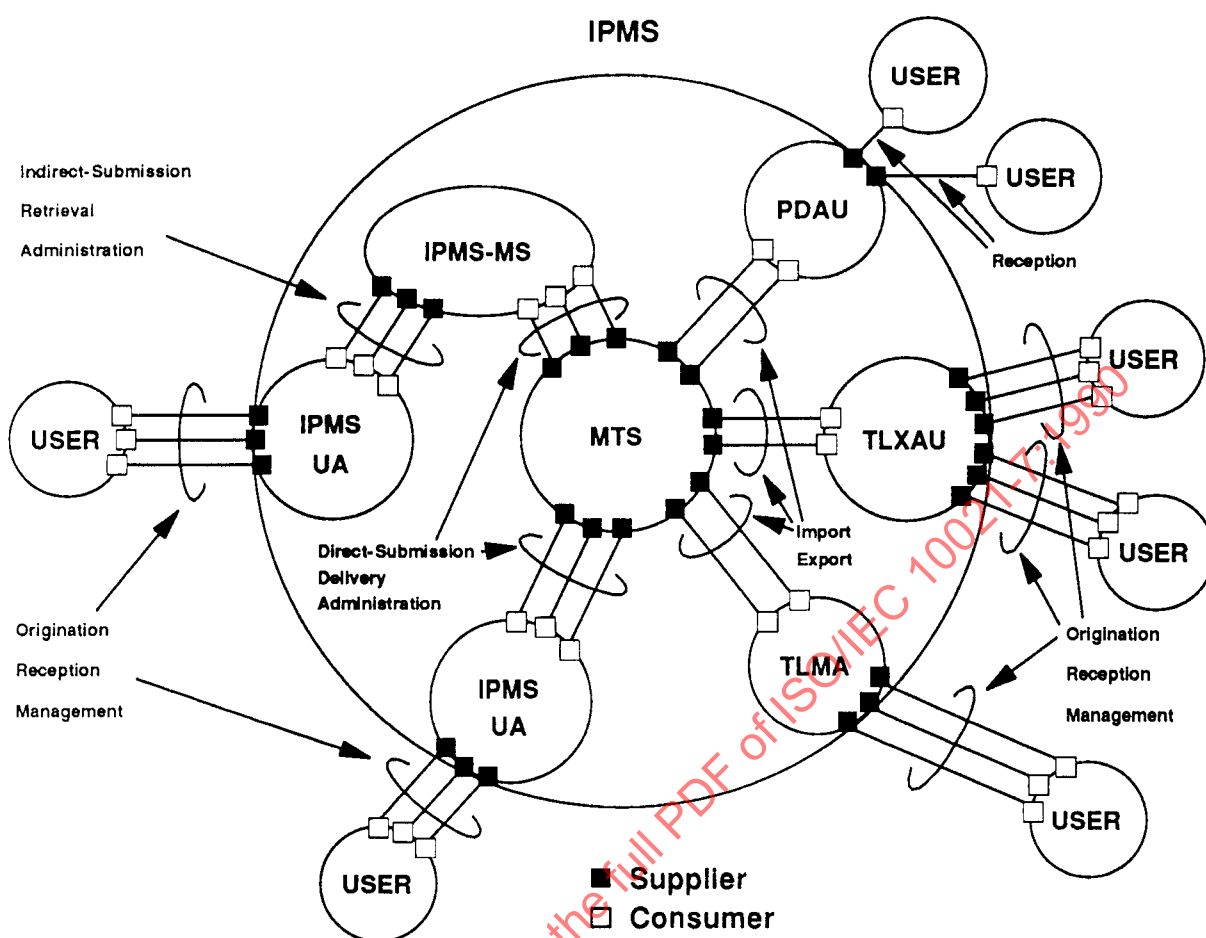


Figure 4
The Interpersonal Messaging System

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

NOTES

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 ISO/IEC 10021-2.
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 clause 16.3.
3. The MTS supplies import and export ports. However, since those ports are not formally defined (in 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.

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

The IPMS comprises any number of IPMS UAs.

NOTE - For brevity, the term "UA" is used throughout the rest of this part of ISO/IEC 10021 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.

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

The IPMS comprises any number of IPMS MSs.

NOTE - For brevity, the term "MS" is used throughout the rest of this part of ISO/IEC 10021 with the meaning of "IPMS MS".

16.3 Telematic Agent

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

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

The IPMS comprises any number of TLMA's.

NOTES

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

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 part of ISO/IEC 10021. 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.

```

tlxau OBJECT
  PORTS (
    origination [S],
    reception   [S],
    management  [S])
  ::= id-ot-tlxau

```

The IPMS comprises any number of TLXAUs.

NOTE - A TLXAU consumes import and export ports. However, since those ports are not formally defined (in 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.

```

pdau OBJECT
  PORTS (
    reception [S])
  ::= id-ot-pdau

```

The IPMS comprises any number of PDAUs.

NOTE - A PDAU consumes import and export ports. However, since those ports are not formally defined (in 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 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 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 part of ISO/IEC 10021.

NOTES

1. It is for historical reasons that the part of ISO/IEC 10021 that defines the IPMS Abstract Service also specifies how a UA (and MS), but not a TLMA or AU, provides it.
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 part of ISO/IEC 10021.

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 O/R name of the UA's user.
 - ii) *Content-type*, *Content-length*, and *Original-encoded-information-types*: Determined from Originate Probe's Content argument as specified in clauses 20.2-20.4.
 - iii) *Content-identifier*: 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.

NOTES

1. The UA shall ignore all properties of Originate Probe's Content argument other than those mentioned above.
2. How the UA employs Probe Submission's Content-identifier result is a local matter.

18.2.2 Originate IPM

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

The arguments of Message Submission shall be as follows:

- a) *Envelope*: The components of this argument that constitute per-message fields shall be as follows; those not explicitly mentioned below shall be as specified by Originate IPM's Envelope argument:
 - i) *Originator-name*: The O/R name of the UA's user.
 - ii) *Content-type* and *Original-encoded-information-types*: Determined from Originate IPM's Content argument as specified in clauses 20.2 and 20.4, respectively.
 - iii) *Content-identifier*: Specified or omitted as a local matter.

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

- b) *Content*: Determined from Originate IPM's Content argument (identified as an IPM) as specified in clause 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 clause 7.2.6.

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.

NOTES

1. How the UA employs Message Submission's Content-identifier result is a local matter.
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 O/R 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*: Specified or omitted as a local matter.
 - iv) *Deferred-delivery-time*: Omitted.

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 clause 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.

NOTES

1. How the UA employs Message Submission's Content-identifier result is a local matter.
2. The inclusion of Message Submission's Extensions result among Originate RN's results is proper and may be the subject of future standardization.

18.3 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 part of ISO/IEC 10021.

18.3.1 Change Auto-discard

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

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

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

18.3.2 Change Auto-acknowledgment

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

- a) **Auto-acknowledge-IPMs:** A Boolean that indicates whether or not auto-acknowledgment is in effect.
- b) **Auto-acknowledge-suppl-receipt-info:** The Suppl Receipt Info field of each RN provoked by auto-acknowledgment.

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

18.3.3 Change Auto-forwarding

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

- a) **Auto-forward-IPMs:** A Boolean that indicates whether or not auto-forwarding is in effect.
- b) **Auto-forward-recipients:** A Sequence of O/R names that identify the users and DLs to which IPMs are being auto-forwarded.
- c) **Auto-forward-heading:** The Heading of each forwarding IPM provoked by auto-forwarding. Its Auto-forwarded field has the value *true*.
- d) **Auto-forward-comment:** The Auto-forward Comment non-receipt field of each NRN conveyed to the originator of an auto-forwarded IPM.

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

18.4 Invocation of Reception Operations

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

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

- a) Report Delivery
- b) Message Delivery

NOTE - The abstract operations of a reception port report no errors.

18.4.1 Receive Report

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

- a) *Envelope:* Report Delivery's Envelope argument.
- b) *Undelivered-object:* Determined from Report Delivery's Returned-content argument as specified in clause 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 clause 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 clause 18.5):

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

18.5 Internal Procedures

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

The procedures involve the following abstract operations of the MTS Abstract Service (which, for the remainder of this 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 Obsolete 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 clause 18.5.2.1).

The NRN shall have the following receipt fields:

- a) *Non-receipt Reason*: The value *ipm-discarded*.
- b) *Discard Reason*: The value *ipm-expired* or *ipm-obsolete*, whichever applies. If both apply, either value may be specified.
- c) *Auto-forward Comment*: Omitted.
- d) *Returned IPM*: If the IPM's return is requested by means of the Notification-requests component of its subject recipient specifier, and the Converted-encoded-information-types component of Message Delivery's Envelope argument is absent, the IPM. Omitted otherwise.

18.5.1.3 Submission of NRN

The UA shall submit the NRN (if any) above by invoking Message Submission. Its Envelope argument shall be as prescribed for auto-acknowledgment (see clause 18.5.2.2), its Content argument determined from the NRN as specified in clause 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 clause 8.1.2).
- c) *IPM Preferred Recipient*: The Recipient component of the IPM's subject recipient specifier, unless its Formal-name component is the O/R name of the UA's user, in which case the field shall be omitted.
- d) *Conversion EITs*: The Converted-encoded-information-types component of Message Delivery's Envelope argument.

The RN shall have the following receipt fields:

- 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.
 - iii) *Per-recipient-fields*: A single field whose Recipient-name component shall be the Originator-name component of Message Delivery's Envelope argument. Reports shall not be requested.
- b) *Content*: Determined from the RN as specified in clause 20.1.

18.5.3 Auto-forwarding

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

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

18.5.3.1 Prevention of Loops

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

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

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

18.5.3.2 Construction of IPM

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

The Message body part shall have the following components:

- a) *Parameters*: The Envelope argument of Message Delivery.
- b) *Data*: The IPM to be forwarded.

18.5.3.3 Submission of IPM

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

- a) *Envelope*: The components of this argument shall be as follows:
 - i) *Originator-name*: The O/R name of the UA's user.
 - ii) *Content-type* and *Original-encoded-information-types*: Determined from the IPM as specified in clauses 20.2 and 20.4.
 - iii) *Content-identifier*: Specified or omitted as a local matter.
 - iv) *Priority*: As specified by Message Delivery's Envelope argument.
 - v) *Per-message-indicators* and *Extensions*: A local matter.
 - vi) *Deferred-delivery-time*: Omitted.
 - vii) *Per-recipient-fields*: Their Recipient-name components shall be the O/R names that make up the Auto-forward-recipients state variable. Their other components are a local matter.
- b) *Content*: Determined from the IPM as specified in clause 20.1.

18.5.3.4 Construction of NRN

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

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

The NRN shall have the following receipt fields:

- a) *Non-receipt Reason*: The value *ipm-auto-forwarded*.
- b) *Discard Reason*: Omitted.
- c) *Auto-forward Comment*: The Auto-forward-comment state variable.
- d) *Returned IPM*: If the IPM's return is requested by means of the Notification-requests component of its subject recipient specifier, and the Converted-encoded-information-types component of Message Delivery's Envelope argument is absent, the IPM. Omitted otherwise.

18.5.3.5 Submission of NRN

The UA shall submit the NRN (if any) above by invoking Message Submission. Message Submission's Envelope argument shall be as prescribed for auto-acknowledgment, its Content argument determined from the NRN as specified in clause 20.1.

19 Message Store Operation

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

19.1 Creation of Information Objects

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

- a) The MS shall maintain a separate information object for each (message containing an) IPM or IPN that is delivered to it.
- b) The MS shall maintain as a separate information object not only each (message containing a) forwarding IPM (pursuant to Item a) but also each (message containing a) forwarded IPM (recursively).
- c) The MS shall assign sequence numbers depth-first to the messages in the hierarchy formed by a forwarding IPM and its forwarded IPMs.

EXAMPLE - If IPM *A* contains IPMs *B* and *C* among its body parts, and if IPM *B* contains IPMs *D* and *E* among its body parts, sequence numbers will be assigned to the IPMs in the following order: *A*, *B*, *D*, *E*, and *C*.

19.2 Maintenance of Attributes

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

- a) For each IPM or IPN it holds, the MS shall support the attributes of annex C as specified therein.
- b) For each IPM it holds, the MS shall give the following meanings to the defined values of the MS-status attribute:
 - i) *new*: No attribute values have been conveyed to the UA.
 - ii) *listed*: At least one attribute value has been conveyed to the UA, and at least one body part has not been conveyed.

- iii) *processed*: All body parts have been conveyed to the UA.
- c) For each IPN it holds, the MS shall give the following meanings to the defined values of the MS-status attribute:
 - i) *new*: No attribute values have been conveyed to the UA.
 - ii) *listed*: At least one attribute value has been conveyed to the UA, and at least one attribute other than Returned IPM has not been conveyed.
 - iii) *processed*: All attributes, with the possible exception of Returned IPM, have been conveyed to the UA.
- d) The MS-status attribute shall reflect the state of affairs prior to an abstract operation invocation that alters its value.
- e) The Content-type attribute of each (message containing an) IPM or IPN that is delivered to the MS shall have the value id-mct-p2-1984 or id-mct-p2-1988 (see annex D), as appropriate, depending upon the content type of the delivered message (see clause 20.2).

19.3 Notification of Non-receipt

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

19.4 Auto-forwarding

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

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

In addition, the MS shall satisfy the following requirements:

- a) Submit an NRN even if it retains a copy of the forwarded IPM.
- b) Draw the NRN's Auto-forward Comment field, if any, from the Other-parameters component.
- c) Draw the cover-note, if any, to be included with the forwarded IPM, from the Other-parameters component.
- d) Prefix the User-relative-identifier component of the This IPM field of the forwarding IPM's heading with, if present, the This-ipm-prefix.

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

19.5 Manual Forwarding

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

The submitted message body and the forwarded message are then combined by inserting the forwarded message as a Message Body Part into the submitted message body.

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 ISO 8825.

20.2 Content Type

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

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

- i) The Heading (of an IPM) lacks the Extensions field.
- ii) The Body (of an IPM) lacks Externally Defined body parts.
- iii) The Parameters element of any Videotex body part (of an IPM) lacks the Syntax member.
- iv) Every component of the IPM or IPN that is a value of a data type defined as part of the MTS Abstract Service meets the constraints of CCITT Recommendation X.411 (1984).

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

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

Otherwise, the Integer 22 shall be specified.

NOTES

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 *X.400-series Implementor's Guide*), except that the Simple Formattable Document body part type, defined in the latter, is omitted from the former.

2. The Integer 2 is favoured, above, over the Integer 22 to foster interworking between systems conforming to this part of ISO/IEC 10021 and systems conforming (only) to CCITT Recommendation X.420 (1984).

3. The MTS does not convert between message content protocols. Thus it does not convert between P2 as defined by this part of ISO/IEC 10021 alone (and denoted by the Integer 22) and P2 as defined by both this part of ISO/IEC 10021 and CCITT Recommendation X.420 (1984) (and denoted by the Integer 2).

20.3 Content Length

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

20.4 Encoded Information Types

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

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

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

- a) *Multiple body parts*: The basic EITs (if any) and NBPs (if any) of the message shall comprise the logical union of the basic EITs and NBPs of the IPM's individual body parts, respectively.
- b) *(Forwarded) Message body part*: The basic EITs (if any) and NBPs (if any) of a Message body part shall be those of the forwarded message.
- c) *Externally Defined body part*: An Externally Defined body part whose extended type corresponds to a basic type (see annex B) shall be treated in the manner prescribed for the basic type.

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

- d) *Basic body part*: The basic EITs (if any) and NBPs (if any) of an individual body part of type other than Message and Externally Defined shall depend upon that body part type as specified in Table 2. A body part type for which the table specifies no basic EITs shall result in the setting of no bits in the basic EITs Bit String.
- e) *Encrypted body part*: The effect of an Encrypted body part upon the basic EITs and NBPs to be specified may be the subject of future standardization.

Table 2
Interpersonal Messaging Basic EITs and NBPs

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

21 Port Realization

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

How a UA, TLMA, or AU concretely realizes the primary ports it supplies is beyond the scope of this part of ISO/IEC 10021.

NOTES

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.
2. A TLMA's realization of its primary ports is specified in part by CCITT Recommendation T.330.
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 part of ISO/IEC 10021 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 part of ISO/IEC 10021, 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 part of ISO/IEC 10021, that particular heading field or extension, or body parts of that particular basic or extended type, whenever the MTS or a user's MS (the latter only in the case of a UA) calls upon it to convey to the user an IPM containing them.

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

22.2 Statement Requirements

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

- a) The heading fields and heading extensions for which he claims conformance.
- b) The basic and extended body part types for which he claims conformance.
- c) In the case of an IPMS UA or IPMS MS, the Interpersonal Messaging-specific MS attributes for which it claims conformance.

22.3 Static Requirements

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

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

- b) An IPMS UA or IPMS MS shall support the Interpersonal Messaging-specific MS attributes for which conformance is claimed, but including as a minimum those designated mandatory in annex C.
- c) An IPMS UA, IPMS MS, TLMA, or AU shall concretely realize its abstract ports as specified in clause 21.
- d) An IPMS UA or IPMS MS shall be able to both submit and accept delivery of messages of both of the content types of clause 20.2. A TLMA or AU shall be able to both import and export such messages.

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 clause 20.2.

Annex A

(normative)

Heading Extensions

This annex defines all (presently defined) heading extensions.

A.1 Incomplete Copy

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

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

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

A.2 Languages

The **Languages** heading extension identifies the languages used in the composition of the IPM's Subject heading field and Body. The extension comprises a Set of zero or more Printable Strings, each one of the two-character language codes identified by ISO 639.

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

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

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

Annex B

(normative)

Extended Body Part Types

This annex specifies a number of extended body part types. Some are equivalent to basic body part types. Others are supplemental.

B.1 Equivalents of Basic Body Part Types

For each basic body part type, this part of ISO/IEC 10021 defines as follows an equivalent extended body part type.

```

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

voice-body-part EXTENDED-BODY-PART-TYPE
  PARAMETERS VoiceParameters IDENTIFIED BY id-ep-voice
  DATA      VoiceData
  ::= id-et-voice

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

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

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

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

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

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

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

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

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

```

B.2 General Text

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

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

```
GeneralTextParameters ::= SET (
  g0-designator [0] CharacterSetDesignator OPTIONAL,
  g1-designator [1] CharacterSetDesignator OPTIONAL,
  g2-designator [2] CharacterSetDesignator OPTIONAL,
  g3-designator [3] CharacterSetDesignator OPTIONAL,
  c0-designator [4] CharacterSetDesignator OPTIONAL,
  c1-designator [5] CharacterSetDesignator OPTIONAL)
```

```
GeneralTextData ::= GeneralString
```

The Parameters component comprises the designators of the G0, G1, G2, G3, C0 and C1 sets that may be present in the Data component. Each character set designator is represented by the escape sequence defined in the registration of that character set, registered in accordance with ISO 2375.

The absence of the G0 and C0 designators imply the use of the characters sets registered with registration numbers 2 and 1 respectively. The absence of the other designators imply that the corresponding set is not designated.

```
CharacterSetDesignator ::= GeneralString (SIZE(3..5))
```

The Data component comprises a single General String. G and C sets other than those defined in the Parameters component shall not be used.

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

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

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

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

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

NOTES

1. The defaults for the G0 and C0 character set designators are those assumed as designated and invoked by the ASN.1 Basic Encoding Rules (ISO 8825) for a General String.

2. These Basic Encoding Rules require that the escape sequences for the G1, G2, G3 and C1 character set designators are repeated within the encoding of the General String. The G sets then have to be invoked using locking shift or single shift control functions.

Annex C

(normative)

Message Store Attributes

As described in ISO/IEC 10021-5, an MS maintains and provides access to certain attributes (e.g., the importance) of each information object it holds. An attribute comprises a type and, depending upon the type, one or more values. Attributes that may assume several values simultaneously (all pertaining to one object) are termed multi-valued, those that may assume just one value, single-valued. Some attributes pertain to information objects of all kinds, others only to those of certain kinds (e.g., those of section two).

This annex defines the MS attributes specific to Interpersonal Messaging.

All of the attributes defined in this annex, except those corresponding to extended body part types (which cannot be enumerated; see clause C.3.6), are listed alphabetically, for reference, in the first column of Table C.1. This table records their presence in a delivered message entry. None of them appears in either a delivered report entry or a returned content entry. See ISO/IEC 10021-5 for an elaboration of the table's legend.

IECNORM.COM : Click to view the full PDF of ISO/IEC 10021-7:1990

Table C.1 (Part 1 of 2)
Summary of MS Attributes

Attribute	V	L	IPM	P NRN RN	L	S
A						
Acknowledgment Mode	S	O	-	- M	Y	Y
Authorizing Users	M	O	C	-	Y	N
Auto-forward Comment	S	O	-	C	Y	N
Auto-forwarded	S	O	C	-	Y	Y
B						
Bilaterally Defined Body Parts	M	O	C	-	N	N
Blind Copy Recipients	M	O	C	-	Y	N
Body	S	M	M	-	N	N
C						
Conversion EITs	M	O	-	C C	Y	N
Copy Recipients	M	O	C	-	Y	N
D						
Discard Reason	S	O	-	C	Y	Y
E						
Encrypted Body Parts	M	O	C	-	N	N
Encrypted Data	M	O	C	-	N	N
Encrypted Parameters	M	O	C	-	N	N
Expiry Time	S	O	C	-	Y	N
Extended Body Part Types	M	O	C	-	Y	Y
G						
G3 Facsimile Body Parts	M	O	C	-	N	N
G3 Facsimile Data	M	O	C	-	N	N
G3 Facsimile Parameters	M	O	C	-	N	N
G4 Class 1 Body Parts	M	O	C	-	N	N
H						
Heading	S	M	M	-	N	N
I						
IA5 Text Body Parts	M	O	C	-	N	N
IA5 Text Data	M	O	C	-	N	N
IA5 Text Parameters	M	O	C	-	N	N
Importance	S	O	C	-	Y	Y
Incomplete Copy	S	O	C	-	Y	N
IPM Entry Type	S	M	M	M	Y	Y
IPM Preferred Recipient	S	O	-	C C	Y	N
IPM Synopsis	S	O	M	-	N	N
IPN Originator	S	O	-	C C	Y	N
L						
Languages	M	O	C	-	Y	N
M						
Message Body Parts	M	O	C	-	N	N
Message Data	M	O	C	-	N	N
Message Parameters	M	O	C	-	N	N
Mixed-mode Body Parts	M	O	C	-	N	N

Table C.1 (Part 2 of 2)
Summary of MS Attributes

Attribute	V	L	IPM	P	NRN	RN	L	S
N								
Nationally Defined Body Parts	M	O	C	-	-		N	N
Non-receipt Reason	S	O	-	M	-		Y	Y
NRN Requestors	M	O	C	-	-		Y	N
O								
Obsoleted IPMs	M	O	C	-	-		Y	N
Originator	S	O	C	-	-		Y	N
P								
Primary Recipients	M	O	C	-	-		Y	N
R								
Receipt Time	S	O	-	-	M		Y	N
Related IPMs	M	O	C	-	-		Y	N
Replied-to IPM	S	O	C	-	-		Y	N
Reply Recipients	M	O	C	-	-		Y	N
Reply Requestors	M	O	C	-	-		Y	N
Reply Time	S	O	C	-	-		Y	N
Returned IPM	S	O	-	C	-		Y	N
RN Requestors	M	O	C	-	-		Y	N
S								
Sensitivity	S	O	C	-	-		Y	Y
Subject	S	O	C	-	-		Y	N
Subject IPM	S	M	-	M	M		Y	N
Suppl Receipt Info	S	O	-	-	C		Y	N
T								
Teletex Body Parts	M	O	C	-	-		N	N
Teletex Data	M	O	C	-	-		N	N
Teletex Parameters	M	O	C	-	-		N	N
This IPM	S	M	M	-	-		Y	N
V								
Videotex Body Parts	M	O	C	-	-		N	N
Videotex Data	M	O	C	-	-		N	N
Videotex Parameters	M	O	C	-	-		N	N
Voice Body Parts	M	O	C	-	-		N	N
Voice Data	M	O	C	-	-		N	N
Voice Parameters	M	O	C	-	-		N	N

Legend
V Single/multi valued
L Support level by MS and access UA
P Presence in delivered message entry
L Available for List, Alert
S Available for Summarize

C.1 Summary Attributes

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

C.1.1 IPM Entry Type

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

```
ipm-entry-type ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX IPMEntryType
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-sat-ipm-entry-type
```

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

This attribute may assume any one of the following values:

- a) *ipm*: The information object is an IPM.
- b) *rn*: The information object is an RN.
- c) *nrn*: The information object is an NRN.

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM or IPN.

C.1.2 IPM Synopsis

The **IPM Synopsis** attribute gives the structure, characteristics, size, and processing status of an IPM at the granularity of individual body parts.

```
ipm-synopsis ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX IPMSynopsis
  SINGLE VALUE
  ::= id-sat-ipm-synopsis
```

The synopsis of an IPM comprises a synopsis of each of its body parts. The synopses appear in the order in which the body parts appear.

```
IPMSynopsis ::= SEQUENCE OF BodyPartSynopsis
```

The synopsis of a body part takes either of two forms depending upon whether the body part is of type Message. This enables the synopsis of a forwarding IPM to encompass the body parts of each forwarded IPM (recursively), as well as those of the forwarding IPM itself.

```
BodyPartSynopsis ::= CHOICE {
  message      [0] MessageBodyPartSynopsis,
  non-message  [1] NonMessageBodyPartSynopsis}

MessageBodyPartSynopsis ::= SEQUENCE {
  number       [0] SequenceNumber,
  synopsis     [1] IPMSynopsis}

NonMessageBodyPartSynopsis ::= SEQUENCE {
  type         [0] OBJECT IDENTIFIER,
  parameters   [1] ExternallyDefinedParameters,
  size         [2] INTEGER,
  processed    [3] BOOLEAN DEFAULT FALSE}
```

The synopsis of a Message body part has the following components:

- a) **Number (M)**: The sequence number that the MS assigns to the entry that the Message body part represents.
- b) **Synopsis (M)**: The synopsis of the IPM that forms the content of the message that the body part represents.

The synopsis of a body part of type other than Message has the following components. For purposes of this synopsis, the body part is considered to be of type Externally Defined, whether or not (see annex B) it was so conveyed to the MS:

- a) **Type (M)**: The body part's extended type, i.e., the Direct-reference component of the body part's Data component. An Object Identifier.
- b) **Parameters (M)**: The body part's format and control parameters, i.e., the body part's Parameters component. An Any.
- c) **Size (M)**: The size in octets of the encoding of the Encoding component of the body part's Data component when the Basic Encoding Rules of ISO 8825 are followed. If those rules permit

several (e.g., both primitive and constructed) encodings of the component, the size may reflect any one of them. An Integer.

- d) **Processed** (D *false*): An indication of whether or not the body part has been conveyed to the UA by means of the MS' List or Fetch abstract operation. A Boolean.

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM.

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

C.2 Heading Attributes

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

C.2.1 Heading

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

```
heading ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX Heading
  SINGLE VALUE
  ::= id-hat-heading
```

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM.

C.2.2 Heading Analyses

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

```
rn-requestors ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ORDescriptor
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-hat-rn-requestors
```

```
nrn-requestors ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ORDescriptor
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-hat-nrn-requestors
```

```
reply-requestors ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ORDescriptor
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-hat-reply-requestors
```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose Heading requests, of at least one user or DL, an RN, NRN, or reply, respectively. It shall maintain one attribute value for every recipient specifier in the IPM's Primary, Copy, or Blind Copy Recipients field whose Notification-requests component includes the value rn (in the case of the first attribute) or nrn (in the case of the second), or whose Reply-requested component signifies, by either its presence or its absence, that a reply is requested (in the case of the third). The value shall be the recipient specifier's Recipient component.

C.2.3 Heading Fields

Some attributes bear the names of heading fields and have those fields as their values. The ordering for the Expiry Time and Reply Time attributes is increasing chronological order.

```

this-ipm ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ThisIPMField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-this-ipm

originator ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX OriginatorField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-originator

replied-to-IPM ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX RepliedToIPMField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-replied-to-IPM

subject ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX SubjectField
  MATCHES FOR EQUALITY SUBSTRINGS
  SINGLE VALUE
  ::= id-hat-subject

expiry-time ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ExpiryTimeField
  MATCHES FOR EQUALITY ORDERING
  SINGLE VALUE
  ::= id-hat-expiry-time

reply-time ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ReplyTimeField
  MATCHES FOR EQUALITY ORDERING
  SINGLE VALUE
  ::= id-hat-reply-time

importance ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ImportanceField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-importance

sensitivity ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX SensitivityField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-sensitivity

auto-forwarded ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX AutoForwardedField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-auto-forwarded

```

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

C.2.4 Heading Sub-fields

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

```

authorizing-users ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX AuthorizingUsersSubfield
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-hat-authorizing-users

primary-recipients ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX PrimaryRecipientsSubfield
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-hat-primary-recipients

copy-recipients ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX CopyRecipientsSubfield
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-hat-copy-recipients

blind-copy-recipients ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX BlindCopyRecipientsSubfield
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-hat-blind-copy-recipients

obsoleted-IPMs ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ObsoletedIPMsSubfield
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-hat-obsoleted-IPMs

related-IPMs ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX RelatedIPMsSubfield
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-hat-related-IPMs

reply-recipients ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ReplyRecipientsSubfield
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-hat-reply-recipients

```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose Heading contains the field whose name the attribute bears. It shall maintain one attribute value for each sub-field.

C.2.5 Heading Extensions

Some attributes bear the names of heading extensions and have as their values the values of those extensions or a part thereof.

```

incomplete-copy ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX IncompleteCopyExtensionValue
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-incomplete-copy

languages ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX Language
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-hat-languages

```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose Heading contains the extension

whose name the attribute bears. In the case of the Languages attribute, the MS shall maintain one attribute value for each language the extension identifies.

C.3 Body Attributes

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

C.3.1 Body

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

```
body ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX Body
  SINGLE VALUE
  ::= id-bat-body
```

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM.

C.3.2 Basic Body Parts

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

An MS holds each forwarded IPM (i.e., each Message body part) as an information object in its own right, separate from the forwarding IPM. That information object, of course, is a message whose content is an IPM. The Message Body Parts attribute below, therefore, has as its values the sequence numbers the MS assigns to those messages.

```
ia5-text-body-parts ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX IA5TextBodyPart
  MULTI VALUE
  ::= id-bat-ia5-text-body-parts

voice-body-parts ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX VoiceBodyPart
  MULTI VALUE
  ::= id-bat-voice-body-parts

g3-facsimile-body-parts ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX G3FacsimileBodyPart
  MULTI VALUE
  ::= id-bat-g3-facsimile-body-parts

g4-class1-body-parts ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX G4Class1BodyPart
  MULTI VALUE
  ::= id-bat-g4-class1-body-parts

teletex-body-parts ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX TeletexBodyPart
  MULTI VALUE
  ::= id-bat-teletex-body-parts

videotex-body-parts ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX VideotexBodyPart
  MULTI VALUE
  ::= id-bat-videotex-body-parts

encrypted-body-parts ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EncryptedBodyPart
  MULTI VALUE
  ::= id-bat-encrypted-body-parts
```

```

message-body-parts ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX SequenceNumber
  MULTI VALUE
  ::= id-bat-message-body-parts

mixed-mode-body-parts ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX MixedModeBodyPart
  MULTI VALUE
  ::= id-bat-mixed-mode-body-parts

bilaterally-defined-body-parts ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX BilaterallyDefinedBodyPart
  MULTI VALUE
  ::= id-bat-bilaterally-defined-body-parts

nationally-defined-body-parts ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX NationallyDefinedBodyPart
  MULTI VALUE
  ::= id-bat-nationally-defined-body-parts

```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose Body contains one or more body parts of the type whose name the attribute bears. It shall maintain one attribute value for each such body part.

C.3.3 Basic Body Part Parameters Components

Some attributes bear the names of basic body part types and have the Parameters components of such body parts as their values.

```

ia5-text-parameters ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX IA5TextParameters
  MULTI VALUE
  ::= id-bat-ia5-text-parameters

voice-parameters ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX VoiceParameters
  MULTI VALUE
  ::= id-bat-voice-parameters

g3-facsimile-parameters ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX G3FacsimileParameters
  MULTI VALUE
  ::= id-bat-g3-facsimile-parameters

teletex-parameters ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX TeletexParameters
  MULTI VALUE
  ::= id-bat-teletex-parameters

videotex-parameters ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX VideotexParameters
  MULTI VALUE
  ::= id-bat-videotex-parameters

encrypted-parameters ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EncryptedParameters
  MULTI VALUE
  ::= id-bat-encrypted-parameters

message-parameters ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX MessageParameters
  MULTI VALUE
  ::= id-bat-message-parameters

```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose Body contains one or more body parts of the type whose name the attribute bears. It shall maintain one attribute value for each such body part.

C.3.4 Basic Body Part Data Components

Some attributes bear the names of basic body part types and have the Data components of such body parts as their values.

```

ia5-text-data ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX IA5TextData
  MULTI VALUE
  ::= id-bat-ia5-text-data

voice-data ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX VoiceData
  MULTI VALUE
  ::= id-bat-voice-data

g3-facsimile-data ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX G3FacsimileData
  MULTI VALUE
  ::= id-bat-g3-facsimile-data

teletex-data ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX TeletexData
  MULTI VALUE
  ::= id-bat-teletex-data

videotex-data ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX VideotexData
  MULTI VALUE
  ::= id-bat-videotex-data

encrypted-data ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EncryptedData
  MULTI VALUE
  ::= id-bat-encrypted-data

message-data ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX MessageData
  MULTI VALUE
  ::= id-bat-message-data

```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose Body contains one or more body parts of the type whose name the attribute bears. It shall maintain one attribute value for each such body part.

C.3.5 Extended Body Part Types

The **Extended Body Part Types** attribute identifies the extended body part types represented in an IPM.

```

extended-body-part-types ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-bat-extended-body-part-types

```

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose Body contains one or more Externally Defined body parts. It shall maintain one attribute value for every such type present. The value shall denote the type as specified in clause 7.3.12.

NOTE - Each value of this attribute corresponds to one of the attributes described in clause C.3.6 below.

C.3.6 Extended Body Parts

Some attributes, unnamed, have as their values the Encoding components (see clause 7.3.12) of the ASN.1 Externals that constitute the Data components of Externally Defined body parts.

To each extended body part type there correspond two attributes. The first attribute is denoted by the Object Identifier that is the Direct-reference component (again, see clause 7.3.12) of the External that constitutes the Data component of a body part of that type. The content of this first attribute is that Data component. The second attribute is denoted by the Object Identifier that is the Direct-reference component of the External that constitutes the Parameters component of a body part of that type. The content of this second attribute is that Parameters component.

An MS that supports one of these body parts shall maintain both attributes 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 each attribute for each such body part.

NOTES

1. The extended body part attributes cannot be enumerated in practice because the extended body part types cannot be so enumerated.
2. The Extended Body Part Types attribute (see clause C.3.5) determines the extended body part attributes for a particular IPM.

C.4 Notification Attributes

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

C.4.1 Common Fields

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

```

subject-ipm ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX SubjectIPMField
  MATCHES FOR EQUALITY SUBSTRINGS
  SINGLE VALUE
  ::= id-nat-subject-ipm

ipn-originator ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX IPNOriginatorField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= d-nat-ipn-originator

ipm-preferred-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX IPMPreferredRecipientField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-ipm-preferred-recipient

conversion-eits ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX MS-EITS
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-nat-conversion-eits

```

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

C.4.2 Non-receipt Fields

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

```

non-receipt-reason ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX NonReceiptReasonField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-non-receipt-reason

```

```

discard-reason ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX DiscardReasonField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-discard-reason

auto-forward-comment ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX AutoForwardCommentField
  MATCHES FOR EQUALITY SUBSTRINGS
  SINGLE VALUE
  ::= id-nat-auto-forward-comment

returned-ipm ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ReturnedIPMField
  SINGLE VALUE
  ::= id-nat-returned-ipm

```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an NRN that contains the field whose name the attribute bears.

C.4.3 Receipt Fields

Some attributes bear the names of receipt fields and have those fields as their values. The ordering for the Receipt Time attribute is increasing chronological order.

```

receipt-time ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ReceiptTimeField
  MATCHES FOR EQUALITY ORDERING
  SINGLE VALUE
  ::= id-nat-receipt-time

acknowledgment-mode ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX AcknowledgmentModeField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-acknowledgment-mode

suppl-receipt-info ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX SupplReceiptInfoField
  MATCHES FOR EQUALITY SUBSTRINGS
  SINGLE VALUE
  ::= id-nat-suppl-receipt-info

```

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

Annex D

(normative)

Reference Definition of Object Identifiers

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 part of ISO/IEC 10021 assigns are assigned in this annex. The annex is definitive for all but those for ASN.1 modules and the IPMS application itself. The definitive assignments for the former occur in the modules themselves; other references to them appear in IMPORT clauses. The latter is fixed.

```
IPMSObjectIdentifiers (joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) object-identifiers(0))
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
```

-- Prologue

-- Exports everything.

IMPORTS -- nothing -- ;

ID ::= OBJECT IDENTIFIER

-- Interpersonal Messaging (not definitive)

id-ipms ID ::= (joint-iso-ccitt mhs-motis(6) ipms(1)) -- not definitive

-- Categories

```
id-mod ID ::= {id-ipms 0} -- modules; not definitive
id-ot ID ::= {id-ipms 1} -- object types
id-pt ID ::= {id-ipms 2} -- port types
id-ref ID ::= {id-ipms 3} -- refinements
id-et ID ::= {id-ipms 4} -- extended body part types
id-hex ID ::= {id-ipms 5} -- heading extensions
id-sat ID ::= {id-ipms 6} -- summary attributes
id-hat ID ::= {id-ipms 7} -- heading attributes
id-bat ID ::= {id-ipms 8} -- body attributes
id-nat ID ::= {id-ipms 9} -- notification attributes
id-mct ID ::= {id-ipms 10} -- message content types
id-ep ID ::= {id-ipms 11} -- extended body part parameters
```

-- 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-upper-bounds ID ::= {id-mod 10} -- not definitive
```

-- Object types

```

id-ot-ipme      ID ::= {id-ot 0}
id-ot-ipms-user ID ::= {id-ot 1}
id-ot-ipms      ID ::= {id-ot 2}
id-ot-ipms-ua   ID ::= {id-ot 3}
id-ot-ipms-ms   ID ::= {id-ot 4}
id-ot-tlma      ID ::= {id-ot 5}
id-ot-tlxau     ID ::= {id-ot 6}
id-ot-pdau      ID ::= {id-ot 7}

```

-- Port types

```

id-pt-origination ID ::= {id-pt 0}
id-pt-reception   ID ::= {id-pt 1}
id-pt-management  ID ::= {id-pt 2}

```

-- Refinements

```

id-ref-primary   ID ::= {id-ref 0}
id-ref-secondary ID ::= {id-ref 1}

```

-- Extended body part types

```

id-et-ia5-text      ID ::= {id-et 0}
id-et-voice         ID ::= {id-et 1}
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}

```

-- Heading extensions

```

id-hex-incomplete-copy ID ::= {id-hex 0}
id-hex-languages       ID ::= {id-hex 1}

```

-- Summary attributes

```

id-sat-ipm-entry-type ID ::= {id-sat 0}
id-sat-ipm-synopsis   ID ::= {id-sat 1}

```

-- Heading attributes

```

id-hat-heading      ID ::= {id-hat 0}
id-hat-this-ipm     ID ::= {id-hat 1}
id-hat-originator   ID ::= {id-hat 2}
id-hat-replied-to-IPM ID ::= {id-hat 3}
id-hat-subject      ID ::= {id-hat 4}
id-hat-expiry-time  ID ::= {id-hat 5}
id-hat-reply-time   ID ::= {id-hat 6}
id-hat-importance   ID ::= {id-hat 7}
id-hat-sensitivity  ID ::= {id-hat 8}
id-hat-auto-forwarded ID ::= {id-hat 9}
id-hat-authorizing-users ID ::= {id-hat 10}
id-hat-primary-recipients ID ::= {id-hat 11}
id-hat-copy-recipients ID ::= {id-hat 12}
id-hat-blind-copy-recipients ID ::= {id-hat 13}
id-hat-obsoleted-IPMs ID ::= {id-hat 14}
id-hat-related-IPMs ID ::= {id-hat 15}
id-hat-reply-recipients ID ::= {id-hat 16}
id-hat-incomplete-copy ID ::= {id-hat 17}
id-hat-languages    ID ::= {id-hat 18}
id-hat-rn-requestors ID ::= {id-hat 19}
id-hat-nrn-requestors ID ::= {id-hat 20}
id-hat-reply-requestors ID ::= {id-hat 21}

```

-- Body attributes

id-bat-body	ID ::= {id-bat 0}
id-bat-ia5-text-body-parts	ID ::= {id-bat 1}
id-bat-voice-body-parts	ID ::= {id-bat 2}
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-voice-parameters	ID ::= {id-bat 14}
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-voice-data	ID ::= {id-bat 21}
id-bat-g3-facsimile-data	ID ::= {id-bat 22}
id-bat-teletex-data	ID ::= {id-bat 23}
id-bat-videotex-data	ID ::= {id-bat 24}
id-bat-encrypted-data	ID ::= {id-bat 25}
id-bat-message-data	ID ::= {id-bat 26}

-- Notification attributes

id-nat-subject-ipm	ID ::= {id-nat 0}
id-nat-ipn-originator	ID ::= {id-nat 1}
id-nat-ipm-preferred-recipient	ID ::= {id-nat 2}
id-nat-conversion-eits	ID ::= {id-nat 3}
id-nat-non-receipt-reason	ID ::= {id-nat 4}
id-nat-discard-reason	ID ::= {id-nat 5}
id-nat-auto-forward-comment	ID ::= {id-nat 6}
id-nat-returned-ipm	ID ::= {id-nat 7}
id-nat-receipt-time	ID ::= {id-nat 8}
id-nat-acknowledgment-mode	ID ::= {id-nat 9}
id-nat-suppl-receipt-info	ID ::= {id-nat 10}

-- Message content types (for use by MS only)

id-mct-p2-1984	ID ::= {id-mct 0} -- P2 1984
id-mct-p2-1988	ID ::= {id-mct 1} -- P2 1988

-- Extended body part parameters

id-ep-ia5-text	ID ::= {id-ep 0}
id-ep-voice	ID ::= {id-ep 1}
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}

END -- of IPMSObjectIdentifiers

```
IPMSObjectIdentifiers2 {iso standard motis(10021) ipms(7) modules(0) object-identifiers(0)}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
```

-- Prologue

-- *Exports everything.*

IMPORTS -- *nothing* -- ;

ID ::= OBJECT IDENTIFIER

-- *Interpersonal Messaging (ISO/IEC extensions)*

id-iso-ipms ID ::= {iso standard motis(10021) ipms(7)}

-- *Categories*

id-iso-mod ID ::= {id-iso-ipms 0} -- *modules; not definitive*

id-iso-cs ID ::= {id-iso-ipms 1} -- *character sets*

-- *Modules*

id-mod-object-identifiers-2 ID ::= {id-iso-mod 0} -- *not definitive*

id-mod-extended-body-part-types-2 ID ::= {id-iso-mod 1} -- *not definitive*

-- *Registration Authority for General Text Character Set EITs*

id-cs-eit-authority ID ::= {id-iso-cs 0}

END -- *of IPMSObjectIdentifiers2*

IECNORM.COM : Click to view the full PDF of ISO/IEC 10021-7:1990

Annex E

(normative)

Reference Definition of Abstract Information Objects

This annex, a supplement to section two, defines for reference purposes the abstract information objects of Interpersonal Messaging.

```
IPMSInformationObjects (joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) information-objects(2))
```

```
DEFINITIONS IMPLICIT TAGS ::=
```

```
BEGIN
```

```
-- Prologue
```

```
-- Exports everything.
```

```
IMPORTS
```

```
-- IPMS Upper bounds
```

```
ub-auto-forward-comment, ub-free-form-name, ub-ipm-identifier-suffix, ub-local-ipm-identifier,  
ub-subject-field, ub-telephone-number
```

```
FROM IPMSUpperBounds (joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) upper-bounds(10))
```

```
-- DTAM
```

```
ProtocolElement
```

```
FROM dtam
```

```
-- MTS Abstract Service
```

```
EncodedInformationTypes, G3FacsimileNonBasicParameters, MessageDeliveryTime, ORAddress, ORName,  
OtherMessageDeliveryFields, SupplementaryInformation, TeletexNonBasicParameters
```

```
FROM MTSAbstractService (joint-iso-ccitt mhs-motis(6) mts(3) modules(0) mts-abstract-service(1));
```

```
Time ::= UTCTime
```

```
-- Information object
```

```
InformationObject ::= CHOICE {
```

```
ipm [0] IPM,  
ipn [1] IPN}
```

```
-- IPM
```

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

-- 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                    ORAddress 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
)

```

```

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

```

```

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))

```

-- This IPM heading field

```

ThisIPMField ::= IPMIdentifier

```

-- Originator heading field

```

OriginatorField ::= ORDescriptor

```

-- Authorizing Users heading field

```

AuthorizingUsersField ::= SEQUENCE OF AuthorizingUsersSubfield

```

```

AuthorizingUsersSubfield ::= ORDescriptor

```

-- Primary Recipients heading field

```

PrimaryRecipientsField ::= SEQUENCE OF PrimaryRecipientsSubfield

```

```

PrimaryRecipientsSubfield ::= RecipientSpecifier

```

-- *Copy Recipients heading field*

CopyRecipientsField ::= SEQUENCE OF CopyRecipientsSubfield

CopyRecipientsSubfield ::= RecipientSpecifier

-- *Blind Copy Recipients heading field*

BlindCopyRecipientsField ::= SEQUENCE OF BlindCopyRecipientsSubfield

BlindCopyRecipientsSubfield ::= RecipientSpecifier

-- *Replied-to IPM heading field*

RepliedToIPMField ::= IPMIdentifier

-- *Obsoleted IPMs heading field*

ObsoletedIPMsField ::= SEQUENCE OF ObsoletedIPMsSubfield

ObsoletedIPMsSubfield ::= IPMIdentifier

-- *Related IPMs heading field*

RelatedIPMsField ::= SEQUENCE OF RelatedIPMsSubfield

RelatedIPMsSubfield ::= IPMIdentifier

-- *Subject heading field*

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

-- *Expiry Time heading field*

ExpiryTimeField ::= Time

-- *Reply Time heading field*

ReplyTimeField ::= Time

-- *Reply Recipients heading field*

ReplyRecipientsField ::= SEQUENCE OF ReplyRecipientsSubfield

ReplyRecipientsSubfield ::= ORDescriptor

-- *Importance heading field*

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

-- *Sensitivity heading field*

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

-- *Auto-forwarded heading field*

AutoForwardedField ::= BOOLEAN

-- *Extensions heading field*

ExtensionsField ::= SET OF HeadingExtension

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

HEADING-EXTENSION MACRO ::=

BEGIN

 TYPE NOTATION ::= "VALUE" type | empty

 VALUE NOTATION ::= value (VALUE OBJECT IDENTIFIER)

END

-- *Body*

Body ::= SEQUENCE OF BodyPart

BodyPart ::= CHOICE {
 ia5-text [0] IA5TextBodyPart,
 voice [2] VoiceBodyPart,
 g3-facsimile [3] G3FacsimileBodyPart,
 g4-class1 [4] G4Class1BodyPart,
 teletex [5] TeletexBodyPart,
 videotex [6] VideotexBodyPart,
 encrypted [8] EncryptedBodyPart,
 message [9] MessageBodyPart,
 mixed-mode [11] MixedModeBodyPart,
 bilaterally-defined [14] BilaterallyDefinedBodyPart,
 nationally-defined [7] NationallyDefinedBodyPart,
 externally-defined [15] ExternallyDefinedBodyPart}

-- *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)}

-- *Voice body part*

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

VoiceParameters ::= SET -- *for future standardization*

VoiceData ::= BIT STRING -- *for future standardization*

-- *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 ProtocolElement

MixedModeBodyPart ::= SEQUENCE OF ProtocolElement

-- *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 -- for future standardization

EncryptedData ::= BIT STRING -- for future standardization

-- *Message body part*

MessageBodyPart ::= SEQUENCE (
 parameters MessageParameters,
 data MessageData)

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

MessageData ::= IPM

-- *Bilaterally Defined body part*

BilaterallyDefinedBodyPart ::= OCTET STRING

-- *Nationally Defined body part*

NationallyDefinedBodyPart ::= ANY

-- Externally Defined body part

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

```
ExternallyDefinedParameters ::= EXTERNAL
```

```
ExternallyDefinedData ::= EXTERNAL
```

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

```
BEGIN
```

```
    TYPE NOTATION ::= Parameters Data
```

```
    VALUE NOTATION ::= value (VALUE OBJECT IDENTIFIER)
```

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

```
    Data            ::= "DATA" type
```

```
END
```

-- IPN

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

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

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

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

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

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

-- Common fields

```
SubjectIPMField ::= IPMIdentifier
```

```
IPNOriginatorField ::= ORDescriptor
```

```
IPMPreferredRecipientField ::= ORDescriptor
```

```
ConversionEITsField ::= EncodedInformationTypes
```

-- Non-receipt fields

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

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

```
AutoForwardCommentField ::= AutoForwardComment

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

ReturnedIPMField ::= IPM

-- Receipt fields

ReceiptTimeField ::= Time

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

SupplReceiptInfoField ::= SupplementaryInformation

-- Message Store Realization

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

END -- of IPMSInformationObjects
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 10021-7:1990

Annex F

(normative)

Reference Definition of Functional Objects

This annex, a supplement to clauses 10, 11, and 16, defines for reference purposes the functional objects of Interpersonal Messaging. It uses the OBJECT and REFINE macros of ISO/IEC 10021-3.

```

IPMSFunctionalObjects {joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) functional-objects(1)}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

IMPORTS

-- IPMS Abstract Service

management, origination, reception
....
FROM IPMSAbstractService {joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) abstract-service(3)}

-- IPMS Object Identifiers

id-ot-ipme, id-ot-ipms, id-ot-ipms-ms, id-ot-ipms-ua, id-ot-ipms-user, id-ot-pdau, id-ot-tlma, id-ot-tlxau,
id-ref-primary, id-ref-secondary
....
FROM IPMSObjectIdentifiers {joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) object-identifiers(0)}

-- TLMA Abstract Service

miscellanea
....
FROM TLMAAbsService {ccitt recommendation(0) t(20) 330 tlmaabsservice(0)}

-- MS Abstract Service

retrieval
....
FROM MSAbstractService {joint-iso-ccitt mhs-motis(6) ms(4) modules(0) abstract-service(1)}

-- MTS Abstract Service

administration, delivery, mTS, submission
....
FROM MTSAbstractService {joint-iso-ccitt mhs-motis(6) mts(3) modules(0) mts-abstract-service(1)}

-- Abstract service definition conventions

OBJECT, REFINE
....
FROM AbstractServiceNotation {joint-iso-ccitt mhs-motis(6) asdc(2) modules(0) notation(1)};

-- "Root" object type

ipme OBJECT
::= id-ot-ipme

```

-- Primary refinement

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

-- Primary object types

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

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

-- Secondary refinement

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

-- Secondary objects

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

```

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

```

```

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

```

```

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

```

```

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

```

END -- of IPMSFunctionalObjects

IECNORM.COM : Click to view the full PDF of ISO/IEC 10021-7:1990

Annex G

(normative)

Reference Definition of Abstract Service

This annex, a supplement to clauses 12 and 13, defines for reference purposes the IPMS Abstract Service. It uses the PORT and ABSTRACT-OPERATION and ABSTRACT-ERROR macros of ISO/IEC 10021-3.

```
IPMSAbstractService {joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) abstract-service(3)}
```

```
DEFINITIONS IMPLICIT TAGS ::=
```

```
BEGIN
```

```
-- Prologue
```

```
-- Exports everything.
```

```
IMPORTS
```

```
-- IPMS Information Objects
```

```
AutoForwardComment, Heading, IPM, NRN, RN
```

```
----
```

```
FROM IPMSInformationObjects {joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) information-objects(2)}
```

```
-- IPMS Object Identifiers
```

```
id-pt-management, id-pt-origination, id-pt-reception
```

```
----
```

```
FROM IPMSObjectIdentifiers {joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) object-identifiers(0)}
```

```
-- MTS Abstract Service
```

```
MessageDeliveryEnvelope, MessageSubmissionEnvelope, MessageSubmissionIdentifier, MessageSubmissionTime,  
ProbeSubmissionEnvelope, ProbeSubmissionIdentifier, ProbeSubmissionTime, RecipientImproperlySpecified,  
ReportDeliveryEnvelope
```

```
----
```

```
FROM MTSAbstractService {joint-iso-ccitt mhs-motis(6) mts(3) modules(0) mts-abstract-service(1)}
```

```
-- Abstract service definition conventions
```

```
ABSTRACT-ERROR, ABSTRACT-OPERATION, PORT
```

```
----
```

```
FROM AbstractServiceNotation {joint-iso-ccitt mhs-motis(6) asdc(2) modules(0) notation(1)};
```

```
Time ::= UTCTime
```

```
-- Ports
```

```
origination PORT
```

```
CONSUMER INVOKES {  
  OriginateProbe,  
  OriginateIPM,  
  OriginateNRN}  
::= id-pt-origination
```

```
reception PORT
```

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

```

management PORT
  CONSUMER INVOKES (
    ChangeAutoDiscard,
    ChangeAutoAcknowledgment,
    ChangeAutoForwarding)
  ::= id-pt-management

```

-- *Origination abstract operations*

```

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

```

```

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

```

```

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

```

-- *Reception abstract operations*

```

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

```

```

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

```

```

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

```

```

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

```

-- *Management abstract operations*

ChangeAutoDiscard ::= ABSTRACT-OPERATION

```

ARGUMENT SET {
    auto-discard-expired-IPMs [0] BOOLEAN,
    auto-discard-obsolete-IPMs [1] BOOLEAN}

RESULT
ERRORS {}
    
```

ChangeAutoAcknowledgment ::= ABSTRACT-OPERATION

```

ARGUMENT SET {
    auto-acknowledge-IPMs [0] BOOLEAN,
    auto-acknowledge-suppl-receipt-info [1]
        SupplementaryInformation}

RESULT
ERRORS {
    SubscriptionError}
    
```

ChangeAutoForwarding ::= ABSTRACT-OPERATION

```

ARGUMENT SET {
    auto-forward-IPMs [0] BOOLEAN,
    auto-forward-recipients [1] SEQUENCE OF ORName OPTIONAL,
    auto-forward-heading [2] Heading OPTIONAL,
    auto-forward-comment [3] AutoForwardComment OPTIONAL}

RESULT
ERRORS {
    SubscriptionError,
    RecipientImproperlySpecified}
    
```

-- *Abstract errors*

SubscriptionError ::= ABSTRACT-ERROR

```

PARAMETER SET {
    problem [0] SubscriptionProblem}
    
```

SubscriptionProblem ::= ENUMERATED {

```

    ipms-eos-not-subscribed(0),
    mts-eos-not-subscribed(1)}
    
```

END -- of *IPMSAbstractService*

IECNORM.COM : Click to view the full PDF of ISO/IEC 10021-7:1990