INTERNATIONAL STANDARD

ISO/IEC 10021-7

First edition
1990-12-01
AMENDMENT 1
1994-08-01

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

Part 7:
Interpersonal Messaging System

AMENDMENT 1: Minor enhancements: File transfer body part and auto-submission indication

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

Partie 7: Système de messagerie de personne à personne

AMENDEMENT 1: Améliorations mineures: Partie du corps de transfert de dossier et indication d'autosoumission

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.

Amendment 1 to International Standard ISO/IEC 10021-7:1990 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee 18, *Document processing and related communication*.

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

Part 7:

Interpersonal Messaging System

AMENDMENT 1: Minor enhancements: File transfer body part and auto-submission indication

Page 2

Clause 2

Add the following new subclause to the end of clause 2:

2.7   File Transfer

This part of ISO/IEC 10021 cites the following File Transfer specifications:

ISO 8571-1 : 1988,   *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 1: General Introduction.*

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

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

ISO 8571-4 : 1988,   *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 4: File Protocol Specification.*

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

Page 53

Annex A

Add the following new subclause to the end of Annex A:

A.3   Auto-submitted

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

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

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

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

The auto-forwarded heading field may also indicate that the message has been submitted without human intervention. The absence of both this heading extension and the auto-forwarded heading field indicates that no information is available as to whether the message-submission involved human control.

Page 55

Annex B

Add the following new subclauses to the end of Annex B:

B.3   File Transfer

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

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

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

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

B.3.1   Related Stored File Parameter

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

- unspecified;

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

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

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

The syntax for this parameter is:

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

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

PathnameandVersion ::= SEQUENCE {
        pathname     [0] Pathname-Attribute,
        file-version [1] GraphicString OPTIONAL}

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

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

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

ExplicitRelationship ::= ENUMERATED {
        unspecified (0),
        new-file    (1),
        replacement (2),
        extension   (3) }
```

The pathname option is intended for use in a manner consistent with ISO 8571-2, as amended by Amendment 1. It is a sequence of elements, each of which represents a name component. When more than one element is encoded, the first element shall be the file name and the remaining elements shall be concatenated to represent the file name prefix.

NOTE 1 - ISO 8571-2 Amendment 1 renamed the "filename" attribute in ISO 8571-2 to the "pathname" attribute.

A message reference has the following components:

a)   **User (C):** Identifies the user who originated the referenced message. One of the user's O/R names. This conditional component shall be present unless the reference is an IPM identifier which does not contain a User component.

b) **User-relative-identifier** (M): Unambiguously identifies a message, distinguishing it from all other messages that the user who is identified by the User component originates. A Printable String of from zero to a prescribed number of characters. A length of zero is discouraged.

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

A body part reference uniquely identifies a body part within a message. It is for use when referencing a message with a content type which includes body part references.

B.3.2   Contents Type Parameter

The contents type parameter indicates the abstract data types of the contents of the file and the structuring information which is necessary if the complete file structure and semantics are to be maintained during the transfer of the file.

```
ContentsTypeParameter ::= Contents-Type-Attribute

Contents-Type-Attribute ::= CHOICE {
        document-type                    [0] SEQUENCE {
              document-type-name      Document-Type-Name,
              parameter             [0] ANY OPTIONAL },
              -- The actual types to be used for values of the parameter field
              -- are defined in the named document type.
        constraint-set-and-abstract-syntax  [1] SEQUENCE {
              constraint-set-name     Constraint-Set-Name,
              abstract-syntax-name    Abstract-Syntax-Name } }

Document-Type-Name ::= OBJECT IDENTIFIER

Constraint-Set-Name ::= OBJECT IDENTIFIER

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

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

The concepts of document-type and constraint set are described fully in ISO 8571-1 and ISO 8571-2. Examples of document types which may be used in this body part are:

a)   unstructured text file (FTAM-1)

b)   unstructured binary file (FTAM-3)

c)   sequential binary file (FTAM-4)

B.3.3   Environment Parameter

The environment parameter describes the environment (e.g., machine, operating system, and application) from which the file originated. It has the following syntax:

```
EnvironmentParameter ::= SEQUENCE {
        application-reference   [0] GeneralIdentifier OPTIONAL,
        machine                 [1] GeneralIdentifier OPTIONAL,
        operating-system        [2] OBJECT IDENTIFIER OPTIONAL,
        user-visible-string     [3] SEQUENCE OF GraphicString OPTIONAL }

GeneralIdentifier ::= CHOICE {
        registered-identifier   [0] OBJECT IDENTIFIER,
        descriptive-identifier  [1] SEQUENCE  OF GraphicString }
```

4

The application-reference field is intended to be used for identifying application programs and versions. The machine field is intended to be used for executable code modules to indicate hardware platforms. The operating-system field is intended to be used to identify the operating system of the processor from which the file originated.

B.3.4   Compression Parameter

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

```
CompressionParameter ::= SEQUENCE {
        compression-algorithm-id      [0] OBJECT IDENTIFIER,
        compression-algorithm-param   [1] ANY DEFINED BY compression-algorithm-id }
```

B.3.5   File Attributes Parameter

The file attributes parameter conveys values of any of a set of optional file attributes. When the recipient is to create a new file, these values are to be used in establishing the initial file attributes.

NOTE - Transfer of an attribute value to a recipient should be interpreted as a request only; no particular recipient behaviour is guaranteed as a result.

The file attributes are technically aligned with ISO 8571-2. The semantic descriptions of these attributes in ISO 8571-2 take precedence over the abbreviated descriptions given below. The file attributes which can be conveyed in this parameter are:

pathname
permitted actions
storage account
date and time of creation
date and time of last modification
date and time of last read access
date and time of last attribute modification
identity of creator
identity of last modifier
identity of last reader
identity of last attribute modifier
availability
object size
future object size
access control
legal qualifications
private use
attribute-extensions

The syntax for the file attributes parameter is as follows:

```
FileAttributes ::= SEQUENCE {
        pathname                                    Pathname-Attribute OPTIONAL,
        permitted-actions                      [1]  Permitted-Actions-Attribute OPTIONAL,
        storage-account                        [3]  Account-Attribute OPTIONAL,
        date-and-time-of-creation              [4]  Date-and-Time-Attribute OPTIONAL,
        date-and-time-of-last-modification     [5]  Date-and-Time-Attribute OPTIONAL,
        date-and-time-of-last-read-access      [6]  Date-and-Time-Attribute OPTIONAL,
        date-and-time-of-last-attribute-modification [7] Date-and-Time-Attribute OPTIONAL,
        identity-of-creator                    [8]  User-Identity-Attribute OPTIONAL,
        identity-of-last-modifier              [9]  User-Identity-Attribute OPTIONAL,
        identity-of-last-reader                [10] User-Identity-Attribute OPTIONAL,
        identity-of-last-attribute-modifier    [11] User-Identity-Attribute OPTIONAL,
        object-availability                    [12] Object-Availability-Attribute OPTIONAL,
        object-size                            [13] Object-Size-Attribute OPTIONAL,
        future-object-size                     [14] Object-Size-Attribute OPTIONAL,
        access-control                         [15] Access-Control-Attribute OPTIONAL,
        legal-qualifications                   [16] Legal-Qualification-Attribute OPTIONAL,
        private-use                            [17] Private-Use-Attribute OPTIONAL,
        attribute-extensions                   [22] Attribute-Extensions OPTIONAL }
```

5

The types of all the above components are defined below or imported from ISO 8571-4.

B.3.5.1 Pathname Attribute

The pathname attribute provides a file name.

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

B.3.5.2 Permitted Actions Attribute

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

B.3.5.3 Storage Account Attribute

The storage account attribute identifies the accountable authority responsible for accumulated file storage charges.

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

Account ::= GraphicString
```

B.3.5.4 Date and Time Attributes

The date and time of creation attribute indicates when the file was created.

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

The date and time of last read access attribute indicates when the contents of the file were last read.

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

B.3.5.5 Identity Attributes

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

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

User-Identity ::= GraphicString
```

B.3.5.6 Availability Attribute

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

B.3.5.7   Object Size Attributes

The object size attribute is set to the nominal size in octets of the complete file.

The future object size attribute indicates the nominal size in octets to which the file may grow as a result of modification and extension.

B.3.5.8   Access Control Attribute

The access control attribute defines conditions under which access to the file is valid.

```
Access-Control-Attribute ::= CHOICE {
        no-value-available [0] NULL,
        -- Indicates partial support of this attribute.
        actual-values      [1] SET OF Access-Control-Element }
        -- The semantics of this attribute are described in ISO 8571-2

Access-Control-Element ::= SEQUENCE {
        action-list        [0] Access-Request,
        concurrency-access [1] Concurrency-Access OPTIONAL,
        identity           [2] User-Identity OPTIONAL,
        passwords          [3] Access-Passwords OPTIONAL,
        location           [4] Application-Entity-Title OPTIONAL }

Access-Request ::= BIT STRING {
        read              (0),
        insert            (1),
        replace           (2),
        extend            (3),
        erase             (4),
        read-attribute    (5),
        change-attribute  (6),
        delete-object     (7) }

Access-Passwords ::= SEQUENCE {
        read-password                [0] Password,
        insert-password              [1] Password,
        replace-password             [2] Password,
        extend-password              [3] Password,
        erase-password               [4] Password,
        read-attribute-password      [5] Password,
        change-attribute-password    [6] Password,
        delete-password              [7] Password,
        pass-passwords               [8] Pass-Passwords,
        link-password                [9] Password }

Password ::= CHOICE { GraphicString, OCTET STRING }

Pass-Passwords ::= SEQUENCE OF Password

Application-Entity-Title ::= SEQUENCE {
        ap-title     ANY,
        ae-qualifier ANY }
```

B.3.5.9   Legal Qualifications Attribute

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

B.3.5.10   Private Use Attribute

The meaning of the private use attribute is not defined.

B.3.5.11 Attribute Extensions

The attribute extensions attribute allows for the inclusion of additional attributes in a manner consistent with Amendment 1 of ISO 8571-2 and ISO 8571-4.

B.3.6 Extensions Parameter

The extensions parameter conveys information accommodated by no other parameter of the file transfer body part. The syntax and usage of this field are the same as those of the extensions heading field specified in 7.2.17.

B.3.7 File Transfer Body Part Data

The data component of the body part contains the file contents being transferred.

The syntax for representation of these contents is implied by the contents-type parameter. When this parameter specifies a document type, the corresponding document type definition describes how to construct a sequence of presentation data values to convey the contents, and identifies the necessary abstract syntax(es). When the contents-type parameter specifies a constraint set and abstract syntax, the contents comprise a sequence of one or more data values from the identified abstract syntax.

The encoding shall be based on the transfer syntax specified as part of the document type definition, if any, or on the ASN.1 basic encoding rules otherwise.

B.3.8 Encoded Information Type

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

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

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

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

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

NOTES

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

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

Page 57

Annex C

Add a new entry in table C.1 after "Auto-forwarded" *for* "Auto-submitted":

```
+---------------------------------------+---+---+----------------+---+---+
|                                       |   |   |         P      |   |   |
| Attribute                             | V | L | IPM  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 |
| Auto-submitted                        | S | O |  C    -    -   | Y | N |
+- B ----------------------------------+---+---+----------------+---+---+
  .                                       .   .         .         .   .
  .                                       .   .         .         .   .
  .                                       .   .         .         .   .
+---------------------------------------+---+---+----------------+---+---+
```

Page 62

Subclause C.2.5

Add after the ASN.1 production for "languages":

```
auto-submitted ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX   AutoSubmitted
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-hat-auto-submitted
```

Page 68

Annex D

Under "-- Categories" *after the definition of* "id-ep" *insert the ASN.1 definition:*

id-eit ID ::= {id-ipms 12} -- *encoded information types*

Under "-- Modules" *in correct numeric sequence add:*

id-mod-file-transfer-body-part-type ID ::= {id-mod 9} -- *not definitive*

Page 69

Annex D

At the end of "-- Extended body part types" *add:*

id-et-file-transfer   ID ::= {id-et 12}

Page 69

Annex D

At the end of "-- Heading extensions" add:

```
id-hex-auto-submitted  ID ::= {id-hex 2}
```

Page 69

Annex D

At the end of "-- Heading attributes" add:

```
id-hat-auto-submitted          ID ::= {id-hat 22}
```

Page 70

Annex D

At the end of "-- Extended body part parameters" add:

```
id-ep-file-transfer       ID ::= {id-ep 12}
```

-- Encoded Information Types

```
id-eit-file-transfer  ID ::= {id-eit 0}
```

Page 85

Annex H

In the IMPORTS *of "-- IPMS Object Identifiers" add (in alphabetic sequence):*

```
      id-hex-auto-submitted,
```

Page 85

Annex H

Add at the end of Annex H before END *add:*

-- Auto-submitted

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

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

Page 88

Annex I

Add the following new subclause to the end of Annex I:

I.3   File Transfer

```
IPMSFileTransferBodyPartType {joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)
                                         file-transfer-body-part-type(9)}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
```

-- *Prologue*

-- *Exports everything.*

```
IMPORTS
```

-- *FTAM Attribute Types*

```
        Attribute-Extensions, Concurrency-Access, Date-and-Time-Attribute, Legal-Qualification-Attribute,
        Object-Availability-Attribute, Object-Size-Attribute, Pathname, Permitted-Actions-Attribute,
        Private-Use-Attribute
                ----
                FROM ISO8571-FTAM
```

-- *IPMS Information Objects*

```
        EXTENDED-BODY-PART-TYPE, ExtensionsField
                ----
                FROM IPMSInformationObjects {joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)
                        information-objects(2)}
```

-- *IPMS Object Identifiers*

```
        id-ep-file-transfer, id-et-file-transfer
                ----
                FROM IPMSObjectIdentifiers {joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)
                        object-identifiers(0)}
```

-- *MTS Abstract Service*

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

-- *File Transfer body part*

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

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

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

11

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

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

PathnameandVersion ::= SEQUENCE {
        pathname      [0] Pathname-Attribute,
        file-version  [1] GraphicString OPTIONAL}

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

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

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

ExplicitRelationship ::= ENUMERATED {
        unspecified (0),
        new-file    (1),
        replacement (2),
        extension   (3) }

ContentsTypeParameter ::= Contents-Type-Attribute

Contents-Type-Attribute ::= CHOICE {
        document-type                    [0] SEQUENCE {
            document-type-name     Document-Type-Name,
            parameter            [0] ANY OPTIONAL },
            -- The actual types to be used for values of the parameter field
            -- are defined in the named document type.
        constraint-set-and-abstract-syntax  [1] SEQUENCE {
            constraint-set-name    Constraint-Set-Name,
            abstract-syntax-name   Abstract-Syntax-Name } }

Document-Type-Name ::= OBJECT IDENTIFIER

Constraint-Set-Name ::= OBJECT IDENTIFIER

Abstract-Syntax-Name ::= OBJECT IDENTIFIER

EnvironmentParameter ::= SEQUENCE {
        application-reference   [0] GeneralIdentifier OPTIONAL,
        machine                 [1] GeneralIdentifier OPTIONAL,
        operating-system        [2] OBJECT IDENTIFIER OPTIONAL,
        user-visible-string     [3] SEQUENCE OF GraphicString OPTIONAL }

GeneralIdentifier ::= CHOICE {
        registered-identifier   [0] OBJECT IDENTIFIER,
        descriptive-identifier  [1] SEQUENCE  OF GraphicString }

CompressionParameter ::= SEQUENCE {
        compression-algorithm-id     [0] OBJECT IDENTIFIER,
        compression-algorithm-param  [1] ANY DEFINED BY compression-algorithm-id }
```

```
FileAttributes ::= SEQUENCE {
        pathname                                     Pathname-Attribute OPTIONAL,
        permitted-actions                       [1]  Permitted-Actions-Attribute OPTIONAL,
        storage-account                         [3]  Account-Attribute OPTIONAL,
        date-and-time-of-creation               [4]  Date-and-Time-Attribute OPTIONAL,
        date-and-time-of-last-modification      [5]  Date-and-Time-Attribute OPTIONAL,
        date-and-time-of-last-read-access       [6]  Date-and-Time-Attribute OPTIONAL,
        date-and-time-of-last-attribute-modification [7] Date-and-Time-Attribute OPTIONAL,
        identity-of-creator                     [8]  User-Identity-Attribute OPTIONAL,
        identity-of-last-modifier               [9]  User-Identity-Attribute OPTIONAL,
        identity-of-last-reader                 [10] User-Identity-Attribute OPTIONAL,
        identity-of-last-attribute-modifier     [11] User-Identity-Attribute OPTIONAL,
        object-availability                     [12] Object-Availability-Attribute OPTIONAL,
        object-size                             [13] Object-Size-Attribute OPTIONAL,
        future-object-size                      [14] Object-Size-Attribute OPTIONAL,
        access-control                          [15] Access-Control-Attribute OPTIONAL,
        legal-qualifications                    [16] Legal-Qualification-Attribute OPTIONAL,
        private-use                             [17] Private-Use-Attribute OPTIONAL,
        attribute-extensions                    [22] Attribute-Extensions OPTIONAL }

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

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

Account ::= GraphicString

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

User-Identity ::= GraphicString

Access-Control-Attribute ::= CHOICE {
        no-value-available [0] NULL,
        -- Indicates partial support of this attribute.
        actual-values      [1] SET OF Access-Control-Element }
        -- The semantics of this attribute are described in ISO 8571-2

Access-Control-Element ::= SEQUENCE {
        action-list        [0] Access-Request,
        concurrency-access [1] Concurrency-Access OPTIONAL,
        identity           [2] User-Identity OPTIONAL,
        passwords          [3] Access-Passwords OPTIONAL,
        location           [4] Application-Entity-Title OPTIONAL }

Access-Request ::= BIT STRING {
        read              (0),
        insert            (1),
        replace           (2),
        extend            (3),
        erase             (4),
        read-attribute    (5),
        change-attribute  (6),
        delete-object     (7) }

Access-Passwords ::= SEQUENCE {
        read-password            [0] Password,
        insert-password          [1] Password,
        replace-password         [2] Password,
        extend-password          [3] Password,
        erase-password           [4] Password,
        read-attribute-password  [5] Password,
        change-attribute-password [6] Password,
        delete-password          [7] Password,
        pass-passwords           [8] Pass-Passwords,
        link-password            [9] Password }
```

13

```
Password ::= CHOICE { GraphicString, OCTET STRING }

Pass-Passwords ::= SEQUENCE OF Password

Application-Entity-Title ::= SEQUENCE {
        ap-title      ANY,
        ae-qualifier  ANY }
```

END -- *of IPMSFileTransferBodyPartType*

Page 89

Annex J

In the IMPORTS *of "-- IPMS Heading extensions" add (in alphabetic sequence):*

```
        AutoSubmitted,
```


Page 89

Annex J

In the IMPORTS *of "-- IPMS Object Identifiers" add (in alphabetic sequence):*

```
        id-hat-auto-submitted,
```


Page 92

Annex J

At the end of "-- Heading extensions" add:

```
auto-submitted ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX  AutoSubmitted
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-hat-auto-submitted
```


Page 99

Annex L

Add to Table L.2 (in alphabetic sequence):

Element of Service	Heading Field
Auto-submitted Indication	Extensions (Auto-submitted)