5i’

6.3 The structure for opcode encoding is given in Figure 4-3/T.150.

Figure 4-3/T.150 [T24.150] à traiter comme tableau MEP, p.

For single-byte opcodes the opcode length indicator bit b5 is ZERO. Bits b4 to b1 represent the opcode, i.e. the opcodes are taken from column 2. For two-byte opcodes the opcode length indicator bit b5 of the first byte is ONE. Bits b4 to b1 of the first byte and bits b5 to b1 of the second byte represent the opcode, i.e. the first byte of the opcode is taken from column 3, the second byte is taken from column 2 or 3.

6.4 The general format for operand encoding is given Figure 4-4/T.150.

Figure 4-4/T.150 [T25.150] à traiter comme tableau MEP, p.

The operand part of a primitive may contain one or more operands, each operand consisting of one or more bytes.

6.5 The encoding of the operands may make use of the following DATA TYPES:

— point P

— colour index CI

— integer number I

— real number R

These data types are coded according to the basic format.

6.6 The basic format for operand encoding is given in Figure 4-5/T.150.

Figure 4-5/T.150 [T26.150] à traiter comme tableau MEP, p.

6.7 Each basic format operand is coded as a sequence of one or more bytes.

Bit b6 of each byte is the extension flag. For single byte operands, the extension flag is ZERO. In multiple byte operands, the extension flag is ONE in all bytes, except the last byte, where it is ZERO.

The most significant part of the operand is coded in the first byte. The least significant part of the operand is coded in the last byte.

In data types P, I and R, bit 5 of the first byte represents the sign bit. Bit 5 = 0 corresponds to positive values. Following data bits represent a binary number. Bit 1 of the last byte is to be considered as the unit of this binary representation.

Data type CI is coded in one single byte (b6 = 0). Bits 5 to 1 give the binary representation of colour indexes.

The coding proposed here for data types P, CI, I and R although derived from Recommendation T.101, Annex C, is a simplified version of the encoding method for these data types, which is only valid after adequate initialization of the protocol description primitives.

6.8 The position of a single point, as well as the position of the first point of a sequence is given by absolute coordinate values x0 and y0, expressed in grid units GU. The encoding structure is given in Figure 4-6/T.150.

6.9 If the coordinate value fits in a single byte, the extension flag is set to ZERO. In that case the x-value is contained in one byte, the y-value is contained in the subsequent byte(s).

6.10 If the coding of a coordinate value requires more than one byte, the complete position information is contained in two contiguous series of bytes. The first series contains the x-value, the second series contains the y-value.

6.11 Each such series consists of contiguous bytes. The extension flag of all bytes in one series, except the last byte, is set to ONE.

The extension flag of the last byte in the series is set to ZERO.

7 Incremental mode coding format

7.1 For incremental mode, the presentation elements trace and closed area are coded, according to the following sequence:

— first point’s position;

— DCC introducer;

— incremental sequence.

Figure 4-6/T.150 [T27.150] à traiter comme tableau MEP, p.

7.2 The position of the first point is coded as defined in §§ 6.8 to 6.11.

7.3 DCC is the abbreviation of differential chain code. The DCC introducer is required in order to preserve compatibility with Recommendation T.101.

7.4 The DCC introducer consists of two bytes, see Figure 4-7/T.150.

Figure 4-7/T.150 [T28.150] à traiter comme tableau MEP, p.

7.5 The format for encoding of the incremental sequence is given in Figure 4-8/T.150.

Figure 4-8/T.150 [T29.150] à traiter comme tableau MEP, p.

7.6 The incremental sequence encoding uses variable length words. To accommodate these words in a sequence of bytes as given in Figure 4-8/T.150, the bit positions b6 to b1 of successive bytes are used as if they constitute a continuous bit channel. The first bit of the first variable length word is placed at b6, and so on.

7.7 The end of the incremental sequence is identified by the end of block code. The remaining bit positions between end of block code and the next byte boundary have no meaning. They will be ignored.

8 Displacement mode coding format

8.1 For displacement mode, the presentation elements trace, closed area and marker are coded according to the following sequence:

— first points position;

— following points.

8.2 For points after the first point in a point list, each displacement is measured with respect to the preceding point of the point list. These displacements are coded as the first point of the list of points.

9 Encoding of the primitives

9.1 The opcodes are defined in Table 4-4/T.150. (The notation x/y means column x, row y, in a 8×16 code table.)

9.2 The notational conventions used are defined in Table 4-5/T.150.

In the further §§ 9.3 to 9.5 the encoding of each primitive is defined as well as the order of the parameters, along with their specific data type.

9.3 The presentation elements trace, closed area and marker are encoded as follows:

Trace

<Trace opcode: 2/0> <point: point list> (2)

OR

<Trace opcode: 2/0> <point: first point>

<DCC introducer: 5/0, 4/1> <Incremental sequence>

Closed area

<Closed area opcode: 2/1> <point: point list> (3)

OR

<Closed area opcode: 2/1>

<Point: first point> <DCC introducer: 5/0, 4/1>

<Incremental sequence>

Marker

<Marker opcode: 3/2, 2/11, 5/2> <point: position>

OR

<Marker opcode: 3/2, 2/11, 5/2> <point: first point> (1)

<DCC introducer: 5/0 4/1> <Incremental sequence>

Clear

<Clear opcode: 3/2, 2/0, 4/0>

H.T. [T30.150]

TABLE 4-4/T.150

Incremental Trace Coding opcodes

Element	Code			
	byte 1	byte 2	byte 3	
Presentation elements	Trace	2/0	—	—
	Closed area	2/1	—	—
	Marker	3/2	2/11	5/2
	Clear	3/2	2/0	4/0
Attribute setting	Set trace thickness	3/1	2/1	
	Set trace texture	3/1	2/2	
	Set trace colour index	3/1	2/0	
	{			
Set closed area interior style				
}	3/1	2/5		
	Set closed area style index	3/1	2/6	
	Set closed area colour index	3/1	2/4	
	Set marker type	3/1	2/12	
	Set marker size	3/1	2/13	
	Set marker colour index	3/1	2/11	
Protocol descriptor	Set domain ring	3/2	2/4	
	Set co-ordinate precision	3/2	2/9	

Tableau 4-4/T.150 [T30.150], p.7

H.T. [T31.150]

TABLE 4-5/T.150

Notational conventions

Item	Meaning
<symbols>	1 occurrence
<symbols> (n)	{
n or more occurrences, with n ≥" 1	
}	
[comments]	Explanation of a production
<x | | >	{
Construction x with meaning y.	
}	

Tableau 4-5/T.150 [T31.150], p.8

9.4 The attribute setting primitives are encoded as follows:

Trace thickness

<Set trace thickness opcode: 3/1, 2/1>

<real = trace thickness scale factor>

Trace texture

<Set trace texture opcode: 3/1, 2/2>

<integer: trace texture> =

<integer: 0> [SOLID]

<integer: 1> [DASHED]

<integer: 2> [DOTTED]

<integer: 3> [DASHED DOTTED]

<all other values> [RESERVED]

Trace colour

<Set trace colour index opcode: 3/1, 2/0>

<colour index: trace colour index> =

<index: 0> [black]

<index: 1> [red]

<index: 2> [green]

<index: 3> [yellow]

<index: 4> [blue]

<index: 5> [magenta]

<index: 6> [cyan]

<index: 7> [white]

Closed area interior style

<Set closed area interior style opcode: 3/1, 2/5>

<integer: fill area interior style>

<integer: 0> [HOLLOW]

<integer: 1> [SOLID]

<integer: 2> [PATTERN]

<integer: 3> [HATCH]

<all other values> [RESERVED]

Closed area style index

<Set closed area style index opcode: 3/1, 2/6>

<integer: closed area style index> = interior style HATCH

<integer: 0> [vertical lines]

<integer: 1> [horizontal lines]

<integer: 2> [45 degrees lines]

<integer: 3> [—45 degrees lines]

<integer: 4> [closed lines, vertical and horizontal]

<integer: 5> [crossed lines, 45 and —45 degrees]

<all other values> [reserved]

Closed area colour index

<Set closed area colour index opcode: 3/1, 2/4>

<colour index: closed area colour index> =

<index: 0> [black]

<index: 1> [red]

<index: 2> [green]

<index: 3> [yellow]

<index: 4> [blue]

<index: 5> [magenta]

<index: 6> [cyan]

<index: 7> [white]

Marker type

<Set marker type opcode: 3/1, 2/12>

<integer: market type> =

<integer: 0> [DOT]

<integer: 1> [PLUS SIGN]

<integer: 2> [ASTERISK]

<integer: 3> [CIRCLE]

<integer: 4> [DIAGONAL CROSS]

<all other values> [RESERVED]

Marker size

<Set marker size scale factor opcode: 3/1, 2/13>

<real: marker size scale factor>

Marker colour

<Set marker colour index opcode: 3/1, 2/11>

<colour index: marker colour index> =

<index: 0> [black]

<index: 1> [red]

<index: 2> [green]

<index: 3> [yellow]

<index: 4> [blue]

<index: 5> [magenta]

<index: 6> [cyan]

<index: 7> [white]

9.5 The protocol descriptor primitives are encoded as follows:

Set domain ring

<Set domain ring opcode: 3/2, 2/4>

<integer: angular resolution factor>

<integer: basic radius of the ring>

Set coordinate precision

<Set coordinate precision opcode: 3/2, 2/9>

<integer: magnitude code> [4]

<integer: granularity code> [1 —9, —10, —11]

<integer: default exponent> [1 —9, —10, —11]

<integer: explicit exponent allowed> [1]

9.6 *Remark 1* — The default value for ‘‘granularity code’’ and ‘‘default exponent’’ is —9.

All the described coding is correct if the values for granularity and for default exponent are equal, and if the value of ‘‘explicit exponent allowed’’ is 1 (i.e. forbidden).

Remark 2 — The primitive set coordinate precision has no effect on reals (e.g. thickness scale factor). Reals are expressed (by default) in fractions of 2 ** —9.

10 Example of differential chain coding

The trace of handwritten information is shown in Figure 4-9/T.150, where (P1, P2, P3) are the sampled points. These points are encoded in the incremental mode; the value of the ring radius is R = 2 and the value of the ring angular resolution factor is p = 0, so the number of reference points on the ring is N = 8 * R/(2 ** p) = 16. On Figure 4-9/T.150, for each point, the corresponding ring with several reference points is shown.

After coding, the new list of points is (Q1, Q2, Q3, Q4, Q5). The coordinate and reference points of Pi and Qj are shown on Table 4-6/T.150. The difference chain code bitstream is shown on Figure 4-10/T.150. This bitstream with the appropriate DCC header could be a block.

The initial trace can also be directly encoded in the displacement mode. Figure 4-11/T.150 shows how the list of points (P_1, P_2, P_3) is encoded in this mode.

H.T. [T32.150]

TABLE 4-6/T.150

T.150 coordinate values and reference point number

	X	Y		X	Y	reference point number
P1	10	10	Q1 Q2	10 12	10 12	— +2
P2	13	14	Q3 Q4	13 14	14 12	+1 —6
P3	14	10	Q5	14	10	—1

Tableau 4-6/T.150 [T32.150], p.9

Figure 4-9/T.150, p.10

Figure 4-10/T.150 [T33.150] à traiter comme tableau MEP, p.11

Figure 4-11/T.150 [T34.150] à traiter comme tableau MEP, p.12

Recommendation T.300

GENERAL PRINCIPLES OF TELEMATIC INTERWORKING

The establishment in various countries of telematic services, computer-based store-and-forward message services and other services creates a need to produce standards to facilitate international message exchange between subscribers to such services.

The CCITT,

considering

(a) the need to transfer messages of different types having a large varietry of formats;

(b) that within the X Series of Recommendations services and optional user facilities for public data networks are defined;

(c) that the F Series of Recommendations defines telematic services and that the T Series of Recommendations defines terminal equipment and control procedures for telematic services;

(d) that a set of Recommendations describes various aspects of message handling systems: X.400 Series,

unanimously declares

that this Recommendation describes the general principles for telematic interworking.

CONTENTS

0 *Introduction*

1 *Scope and field of application*

2 *References*

3 *Definitions*

4 *Symboles and abbreviations*

5 *Conventions*

6 *Telematic interworking model*

6.1 Overview

6.2 Telematic interworking involving the MTS

6.3 Telematic interworking not involving the MTS

6.4 Physical configuration — Definition of the telematic interworking facility (TIF)

7 *The telematic interworking system*

7.1 Definition of the TISs

7.2 TIS involving the MTS

7.3 TIS not involving the MTS

8 *Refinement of the TLMA principles of telematic access protocol*

0 Introduction

This Recommendation is the first in a series of Recommendations dealing with telematic interworking.

Telematic interworking is the generic name for a set of applications provided to telematic users. Each of these applications is called a telematic interworking application (TIA), and involves a subset of the following:

— allowing efficient interchange of information between telematic terminals which cannot interact directly;

— providing additional features to standardized telematic services, such as document storage and multiaddressing;

— providing access to or participation in CCITT defined services such as telex, interpersonnal messaging, directory services, etc.

Two TIAs are defined in the present set of Recommendations, namely:

— participation of telematic users in the IPM service;

— teletex to telex interworking.

Other TIAs are for further study.

Some TIAs may be defined to be operated on a standalone basis, e.g. teletex to telex interworking.

1 Scope and field of application

This Recommendation defines the general principles for telematic interworking. It defines the principles of the telematic access protocols as the protocols used by the telematic terminal to participate in telematic interworking applications.

This Recommendation makes use of the concepts defined for message handling. The architectural basis and foundation for message handling is defined in the X.400 Series of Recommendations.

The other Recommendations in the T.300 Series define the telematic interworking applications and the protocols used by telematic terminals to make these applications available to their users. Only the protocols dealing with the telematic access to CCITT-defined services are being considered.

2 References

This Recommendation cites the documents listed below:

— Rec. T.330: Telematic access to interpersonal messaging system

— Rec. X.400: Message handling systems: System and service overview

— Rec. X.402: Message handling systems: Overall architecture

— Rec. X.420: Message handling systems: Interpersonal messaging system.

3 Definitions

This Recommendation makes use of terms defined in Recommendations X.400, X.402 and X.420.

4 Symbols and abbreviations

AU Access unit

C Conditional/consumer

CF Conversion facility

IPM Interpersonal messaging

IPMS Interpersonal messaging system

IPM-UA Interpersonal messaging user agent

M Multiple

MS Message store

MTA Message transfer agent

MTS Message transfer system

PDS Physical delivery system

PTTXAU Public teletex access unit

TIA Telematic interworking application

TIAS Telematic interworking abstract service

TIU Telematic interworking unit

TLM Telematic

TLMA Telematic agent

TLMAU Telematic access unit

TLM-TER Telematic terminal

TTX Teletex

UA User agent

5 Conventions

This Recommendation makes use of no specific convention.

6 Telematic interworking model

6.1 *Overview*

The telematic interworking model serves as a tool to aid in the development of Recommendations on telematic interworking. It depicts different interworking scenarios, and the use of the message handling model for telematic interworking.

The model is applicable in two cases, the first one involving the MTS, the second one not involving the MTS. The model provides only a functional description and does not mandate any specific implementation or interfaces.

6.2 *Telematic interworking involving the MTS*

Figure 1/T.300 depicts the telematic interworking model when the MTS is involved.

It allows:

a) telematic to telematic interworking, through the use of the MTS for relaying the telematic information;

b) participation of a telematic user in the IPM service (telematic to IPM-UA interworking through the MTS);

c) participation of a telematic user in any other CCITT-defined application in the field of message handling (telematic to other UA interworking);

d) telematic to other CCITT-defined services interworking through the MTS and the appropriate access units.

Figure 1/T.300, p.

6.3 *Telematic interworking not involving the MTS*

Figure 2/T.300 depicts the telematic interworking model when the MTS is not involved.

Figure 2/T.300, p.

In this case, TLMAUs and AUs are the same as in § 6.2. The TIU may provide the appropriate subset of the MTS service that will enable information to be conveyed between TLMAUs, or between TLMAUs and AUs. The TIU performs, when necessary, the relevant conversion and dispatching functions.

6.4 *Physical configurations — Definition of the telematic interworking facility (TIF)*

A telematic interworking facility (TIF) is a real system incorporating a valid combination of functional units within telematic interworking. The present clause defines the valid combinations.

6.4.1 *TIF involving the MTS*

In the case of telematic interworking involving the MTS, valid combinations of functional units are depicted in Table 1/T.300.

H.T. [T1.300]

TABLE 1/T.300

Physical confirgurations of a TIF involving the MTS

	Functional units						
	TLMAU	MTA	MS	UA	TLX AU	PDAU	Other AU
TIF	M	1	[M]	[M]	[M]	[1]	[M]

M Multiple

[| Optional

MS Message store

Tableau 1/T.300 [T1.300], p.

Two rules apply to define a TIF in this case:

— At least one TLMAU is present.

— TLMAUs are necessarily co-resident with the MTA they are in relation with.

The definition of other entities present in a TIF (UAs, MSs, TLXAUs, PDAUs, other AUs) and their relations with the MTA is outside the scope of the T.300 Series of Recommendations.

6.4.2 *TIF not involving the MTS*

This case corresponds to a stand-alone TIF, that is a real-system interacting with telematic terminals, and optionally with other CCITT-defined systems (e.g. telex system), but not with MHS nor with other TIFs.

The valid combinations of functional units are depicted in Table 2/T.300.

H.T. [T2.300]

TABLE 2/T.300

Physical configurations of a TIF not involving the MTS

	Functional units			
	TLMAU	TIU	TLXAU	Other AU
TIF	M	1	[M]	[M]

M Multiple

[| Optional

Tableau 2/T.300 [T2.300], p.

7 The telematic interworking system

Every TIA is provided to the user by a system called telematic interworking system (TIS). This clause provides an abstract model of the TIS.

7.1 *Definition of the TISs*

The TIS and associated users are modelled as objects, as depicted in Figure 3/T.300.

Figure 3/T.300, p.

A user and the TIS are paired through the use of one or more ports. At each of these ports, one or more abstract operations are made available to the user. The collection of these abstract operations will define the abstract-service (called telematic interworking abstract service — TIAS) provided by the TIS.

All ports and operations are application-dependent, and thus are described in appropriate (application-specific) Recommendations.

One application of TIS that is defined is the IPMS. IPM-ports and operations are described in Recommendations X.420 and T.330.

The refinement of the TISs leads to two cases, the first one involving the MTS, the second one not involving the MTS.

7.2 *TIS involving the MTS*

The TIS may be refined according to Figure 4/T.300.

Figure 4/T.300, p.

The refinement or the TIS exposes the following component objects:

— The MTS, as defined in Recommendations X.402 and X.411.

— The TLMA, providing the user of a telematic terminal with the telematic interworking abstract service (TIAS).

— The UA, which provide TIAS to users not using a telematic terminal. One application is the IPM-UA, in Recommendation X.420.

— The AUs, which allow intercommunication with other CCITT-defined services. An example of such an AU is the PDAU, defined in the X.400 Series of Recommendations.

The description of UAs and AUs is outside the scope of the T.300 Series of Recommendations.

The ports and operations between the MTS and the other objects listed above are defined in the X.400 Series of Recommendations.

7.3 *TIS not involving the MTS*

This case is for further study.

8 Refinement of the TLMA principles of telematic access protocols

The refinement of a TLMA exposes two component objects: the telematic terminal (TLM) and the telematic access unit (TLMAU).

A TLM is functional object corresponding to a terminal as defined in the relevant T Series of Recommendations.

The TLMAU is defined in Recommendation T.330 in the case of telematic access to IPMS.

The TLM and TLMAU are linked together through one or more port and a set of operations describing them.

Ports and operations are application-dependent and thus are described in appropriate (application-specific) Recommendations.

The realization of these operations involves some transfer of information between the TLM and the TLMAU. It is called a telematic access protocol (see Figure 5/T.300).

Figure 5/T.300, p.

Recommendation T.330

TELEMATIC ACCESS TO INTERPERSONAL MESSAGE SYSTEM

(Melbourne, 1988)

The establishment in various countries of telematic services and computer-based store-and-forward message service in association with public data networks creates a need to produce standards to facilitate international message exchange between subscribers to

such services.

The CCITT,

considering

(a) the need for interpersonal messaging and message transfer services;

(b) the need to transfer messages of different types having a large variety of formats;

(c) that within the X Series of Recommendations services and optional user facilities for public data networks are defined;

(d) that the F Series of Recommendations defines telematic services and that the T Series of Recommendations defines terminal equipment and control procedures for telematic services;

(e) that a set of Recommendation describes various aspects of message handling systems: X.400 Series;

(f) that Recommendation T.300 describes general principles of telematic interworking,

unanimously declares

that this Recommendation describes the access protocol to be used by telematic terminals when making additional use of the interpersonal messaging system.

CONTENTS

0 *Introduction*

1 *Scope and field of application*

2 *References*

3 *Definitions*

4 *Abbreviations*

5 *Conventions*

6 *Overview of telematic access to IPMS*

6.1 Abstract model

6.2 Functional model

6.3 Access for registered and non-registered users

7 *IPMS in the context of telematic interworking*

7.1 Objects and ports description

7.2 Origination, reception and management ports, services and operations

7.3 Miscellanea port services and operations

8 *Refinement of the TLMA object*

8.1 Object and ports description

8.2 The mhs-doc-xfer port operations

9 *Abstract errors*

10 *Realization of abstract operations*

10.1 Description of TAPDU

10.2 Operation of the TLMAU

11 *Formats and coding of TAPDU*

11.1 Principles

11.2 Structure and format of TAPDU

11.3 Coding of TAPDU

11.4 Format of TAPDU

11.5 Reference between TAPDU components and its coding format

12 *Error recovery*

13 *Control procedures*

13.1 Session control procedures

13.2 Document control procedures

13.3 Log-on procedures

Annex A — Formal definition of TLMA abstract service

Annex B — Format of TAPDU components

Annex C — Element ID list

Annex D — Element of service for TTX/IPM service intercommunications

0 Introduction

Recommendation T.330 is one of a series of Recommendations dealing with telematic interworking. Telematic interworking is the generic name for a set of applications provided to telematic users. Each of those applications is called a telematic interworking application (TIA).

Access to and participating in interpersonal messaging system (IPMS) are one of the telematic interworking applications. This Recommendation aims at specifying this application.

1 Scope and field of application

This Recommendation defines the abstract service provided by the telematic agent (TLMA) which is defined as an object of IPMS. It specifies not only abstract operations provided by TLMAU but also access protocol (P5) to be used between a TLMAU and a telematic (TLM) terminal, when participating in and accessing the IPMS. The P5 access protocol is a generalized access protocol; it is applicable to other applications such as network based storage for the teletex service. The TLM terminals being considered in this Recommendation are teletex, G4 facsimile and mixed mode terminals. The use of other types of TLM terminals are for further study.

Other Recommendations in the series contain description on telematic interworking model, the functions of the telematic access unit (TLMAU), and telematic access protocol to specific services, such as telematic, telex, directory, etc. Recommendation T.300 outlines the principles of telematic interworking procedures.

Section 6 of this Recommendation defines overview of telematic access to IPMS provided by TLMA object. Section 7 defines the IPMS in the context of telematic interworking. Section 8 refines the TLMA object and defines abstract operations at a specific port of TLMAU and TLM terminal. Section 9 defines abstract errors used in telematic interworking. Section 10 specifies an access protocol (P5). Section 11 specifies formatting and coding rule of protocol. Section 12 specifies an error recovery mechanism. Section 13 specifies control procedures.

The purpose of a TLMAU is to aid the user of a TLM terminal in gaining access to the features of the IPMS. The TLMAU, which is associated with a message transfer system (MTS), provides the TLM terminal with access to the IPMS.

The TLMAU may also provide a *document storage (DS)* facility to accept delivery of messages from the MTS for TLM users. Document storage is basically defined as a TLM terminal storage extension facility located in the TLMAU allowing reservation of a specific amount of storage for an individual user. Users of TLM terminals may also be registered as users of DS.

2 References

This Recommendation cites the documents listed below.

2.1 *Telematic interworking*

— Rec. T.300: General principles of telematic interworking.

2.2 *Message handling systems*

— Rec. X.400: Message handling systems: System and service overview

— Rec. X.402: Message handling systems: Overall architecture

— Rec. X.407: Message handling systems: Abstract service definition conventions

— Rec. X.411: Message handling systems: Message transfer system: Abstract service definition and procedures

— Rec. X.413: Message handling systems: Message store: Abstract service definition

— Rec. X.419: Message handling systems: Protocol specifications

— Rec. X.420: Message handling systems: Interpersonal messaging system

2.3 *Control procedures*

— Rec. T.62: Control procedures for Teletex and Group 4 facsimile services

2.4 *ASN.1 coding*

— Rec. X.208: Specification of abstract syntax notation one (ASN.1)

— Rec. X.219: Remote operation

2.5 *Address*

— Rec. X.121: International numbering plan for public data networks

2.6 *Character repertoires*

— Rec. T.61: Character repertoire and coded character sets for the international Teletex service

2.7 *Intercommunication*

— Rec. F.422: Intercommunication between Teletex service and IPM service.

— Rec. F.203: Network based storage for the Teletex service.

3 Definitions

This Recommendation uses the terms many of those used in Recommendations X.402, X.411 and X.420.

In addition to the above terms, this Recommendation uses as terms the names of abstract objects, ports, operations and errors; the names of ASN.1 data types; the names of the information item types and values this Recommendation specifies.

4 Abbreviations

ASN.1 Abstract syntax notation one

AU Access unit

C Conditional/consumer

CDC Command document continue

CF Conversion facility

CSCC Command session change control

CSS Command session start

DN Delivery status notification

DS Document storage

G3 Group 3 facsimile

G4 Group 4 facsimile

ID Identity

IP Interpersonal

IPM Interpersonal messaging

IPMAS Interpersonal messaging abstract service

IPME Interpersonal messaging environment

IPMS Interpersonal messaging system

IPM-UA Interpersonal messaging user agent

IPN Interpersonal notification

M Mandatory

MS Message store

MT Message transfer

MTA Message transfer agent

MTAS Message transfer abstract service

MTS Message transfer system

NDN Non-delivery status notification

NL New line

NRN Non-receipt notification

O/R Originator/receipt

PDAU Physical delivery access unit

PTTXAU Public Teletex access unit

P5 Telematic access protocol

RN Receipt status notification

S Supplier

TAPDU Telematic access protocol data unit

TIA Telematic interworking application

TID Terminal identification

TLM Telematic

TLMA Telematic agent

TLMAU Telematic access unit

TLM-TER Telematic terminal

TLXAU Telex access unit

TTX Teletex

UA User agent

5 Conventions

This Recommendation uses the descriptive conventions identified below.

5.1 *ASN.1*

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

a) to specify the functional objects, the OBJECT and REFINE macros and associated conventions of Recommendation X.407;

b) to specify the information objects (and other data types and values of all kinds), ASN.1 itself;

c) to specify the abstract service, the PORT and ABSTRACT-BIND, -UNBIND, -OPERATION, and -ERROR macros and associated conventions of Recommendation X.407.

5.2 *Grade*

Whenever this Recommendation describes a class of data structure (e.g. Headings) having components (e.g. fields), each component is categorized as one of the following grades:

a) Mandatory (M): A mandatory component shall be present in every member of the class.

b) Conditional (C): A conditional component shall be present in a member of the class as dictated by this Recommendation.

6 Overview of telematic access to IPMS

6.1 *Abstract model*

This Recommendation makes use of the message handling abstract service definitions conventions defined in Recommendation X.407. These conventions provide a descriptive tool for the specification of information processing tasks in abstract terms. This ensures that a tasks functional requirements are stated independently of its realization.

6.2 *Functional model*

This section provides a functional model of telematic access to IPMS. The purpose of this model is to provide a general description of the functional entities, which are then explicitly defined using the definitions and conventions found in Recommendation X.407, and further refined as necessary, in following sections (see Figure 1/T.330).

Figure 1/T.330, p.

The functional model comprises the following functional entities:

— Telematic agent (TLMA): Logical entity only which comprises the TLMAU and the telematic terminal. The TLMA is useful as an object in the refinement of the IPMS.

— Telematic access unit (TLMAU): Functional entity which provides all of the interworking functions between telematic codes and protocols and IPMS codes and protocols. The TLMAU also supports the DS functionality.

— Telematic terminal (TLM-TER): The telematic terminal.

— Access unit (AU): Functional entity which provides access to message handling applications for indirect users of the MTS.

— Document storage (DS): Extension of the telematic terminal storage capabilities. The TLMAU may optionally, on a subscription basis, deliver messages to a DS. The terminal may then retrieve the message for the document storage when convenient.

— Message store (MS): Functional entity which provides single direct user of message handling with capabilities for message storage. Although the MS and DS provide a similar functionality, there is no relationship between the two.

— Message transfer system (MTS): Functional entity which conveys information objects between individual users and members of distribution lists.

— User agent (UA): Functional entity by means of which a direct user engages in message handling.

6.3 *Access for registered and non-registered users*

Two types of access to the IPM service are defined within this Recommendation. Registered users of the IPM service who wish to use telematic terminal equipment to access the IPM service are provided with complete IPM service functionality with any full implementation of this Recommendation.

Telematic terminal equipment users who are not registered IPM service subscribers but who wish to direct a message to an IPM service user are provided with a subset of the functionality defined within this Recommendation, in accordance with Recommendation F.422 and Annex D of this Recommendation. This functionality is referred to as a public teletex access unit (PTTXAU).

7 IPMS in the context of telematic interworking

7.1 *Objects and ports description*

The refinement of the IPMS is found in Recommendation X.420 (interpersonal messaging system). The IPMS refinement describes secondary objects, one of which is the telematic agent (TLMA) which is paired to the MTS by the import and export ports.

The TLMA is visible to the telematic user through four ports, namely: origination, reception, management and miscellanea. The origination, reception and management port services and operations are described fully in Recommendation X.420. The miscellanea port services and operations are described in this Recommendation. The import and export port services and operations are described in Recommendation X.411.

H.T. [T1.330]

tlma

OBJECT

tlma

PORTS { | rigination

[S],

tlma

PORTS { | fR

reception

[S],

tlma

PORTS { | fR

management

[S],

tlma

PORTS { | fR

miscellanea

[S],

tlma

PORTS { | fR

import

[C],

tlma

PORTS { | fR

export

[C] }

tlma

::= id-ot-tlma

H.T. [T2.330]

tlm-user

OBJECT

tlma

PORTS { | rigination

[C],

tlma

PORTS { | fR

reception

[C],

tlma

PORTS { | fR

management

[C],

tlma

PORTS { | fR

miscellanea

[C] }

tlma

::= id-ot-tlm-user

H.T. [T3.330]

origination PORT

tlma

CONSUMER INVOKERS { | riginateProbe,

tlma

CONSUMER INVOKERS { | fR

OriginateIPM,

tlma

CONSUMER INVOKERS { | fR

OriginateRN,

tlma

CONSUMER INVOKERS { | fR

CancelIPM }

tlma

PORTS { | fR

::= id-pt-origination

reception PORT

tlma

CONSUMER INVOKERS { | eceiveReport,

tlma

CONSUMER INVOKERS { | fR

ReceiveIPM,

tlma

CONSUMER INVOKERS { | fR

ReceiveRN,

tlma

CONSUMER INVOKERS { | fR

ReceiveNRN }

tlma

PORTS { | fR

::= id-pt-reception

management PORT

tlma

CONSUMER INVOKERS { | hangeAutoDiscard,

tlma

CONSUMER INVOKERS { | fR

ChangeAutoAcknowledgment,

tlma

CONSUMER INVOKERS { | fR

ChangeAutoForwarding }

tlma

PORTS { | fR

::= id-pt-management

H.T. [T4.330]

miscellanea PORT

tlma

SUPPLIER PERFORMS { | hangeSubscriptionProfile,

tlma

SUPPLIER PERFORMS { | fR

DSList,

tlma

SUPPLIER PERFORMS { | fR

DSDelete,

tlma

SUPPLIER PERFORMS { | fR

DSFetch,

tlma

SUPPLIER PERFORMS { | fR

MessageStatus }

tlma

PORTS { | fR

::= id-pt-miscellanea

H.T. [T5.330]

ChangeSubscriptionProfile ::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | s-mode

[0] DSMode OPTIONAL,

tlma

ARGUMENT SET { | fR

error-recovery-mode

[1]

ErrorRecoveryMode OPTIONAL,

tlma

ARGUMENT SET { | fR

message-delete-mode

[2]

MessageDeleteMode OPTIONAL }

tlma

RESULT { }

tlma

ERRORS { | ame-error,

tlma

ERRORS { | fR

ds-error,

tlma

ERRORS { | fR

subscription-profile-error }

H.T. [T6.330]

DSList

::= ABSTRACT-OPERATION

tlma

ARGUMENT { }

tlma

RESULT SET { | 0] SET OF ListReport OPTIONAL }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

ListReport

::= SET { | etrieval-id

[0]

RetrievalIdentifier,

ListReport

::= SET { | fR

message-type

[1]

MessageType,

ListReport

::= SET { | fR

priority

[2]

Priority OPTIONAL,

ListReport

::= SET { | fR

message-length

[3]

MessageLength OPTIONAL,

ListReport

::= SET { | fR

originator-name

[4]

OrName OPTIONAL }

H.T. [T7.330]

DSDelete

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | elector [0] SET OF RetrievalIdentifier }

tlma

RESULT { }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

H.T. [T8.330]

DSFetch

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET OF { | etrieval-id

[0]

RetrievalIdentifier,

tlma

ARGUMENT SET OF { | fR

delete-after-output

[1]

DeleteAfterOutput OPTIONAL }

tlma

RESULT SET { | etrieval-id

[0] RetrievalIdentifier,

tlma

RESULT SET { | fR

message-report

[1]

MessageReport }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

H.T. [T9.330]

MessageStatus

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | 0] QueryIdentifier OPTIONAL }

tlma

RESULT SET { | eport-time

[0] DateandTime,

tlma

RESULT SET { | fR

reported-message-id

[1]

MessageIdentifier,

tlma

RESULT SET { | eported-message-id

[2]

SEQUENCE OF StatusInfo }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

message-status-error }

QueryIdentifier

::= CHOICE { | ubmission-id

[0] MessageIdentifier,

QueryIdentifier

::=

CHOICE { | fR

correlation-info

[1] CallIdentification }

StatusInfo

::= SET { | tatus

[0] Status,

StatusInfo

::= SET { | fR

per-recipient-info

[1]

PerRecipientReportDeliveryFields OPTIONAL }

H.T. [T10.330]

tlma-refinement REFINE tlma AS

tlmau

mhs-doc-xfer

[S] PAIRED with { | lm-ter }

tlm-ter

origination

[S] VISIBLE

tlm-ter

reception

[S] VISIBLE

tlm-ter

management

[S] VISIBLE

tlm-ter

miscellanea

[S] VISIBLE

tlm-ter

::= id-ref-secondary

H.T. [T11.330]

tlm-ter

OBJECT

tlma

PORTS { | rigination

[S],

tlma

PORTS { | fR

reception

[S],

tlma

PORTS { | fR

management

[S],

tlma

PORTS { | fR

miscellanea

[S],

tlma

PORTS { | fR

mhs-doc-xfer

[C] }

tlma

::= id-ot-tlm-ter

tlmau

OBJECT

tlma

PORTS { | hs-doc-xfer

[S],

tlma

PORTS { | fR

import

[C],

tlma

PORTS { | fR

export

[C] }

tlma

::= id-ot-tlm-user

H.T. [T12.330]

mhs-doc-xfer PORT

tlma

SUPPLIER PERFORMS { | essageSend,

tlma

SUPPLIER PERFORMS { | fR

MessageProbe,

tlma

SUPPLIER PERFORMS { | fR

ExplicitReceive,

tlma

SUPPLIER PERFORMS { | fR

MessageCancel,

tlma

SUPPLIER PERFORMS { | fR

Register,

tlma

SUPPLIER PERFORMS { | fR

DSList,

tlma

SUPPLIER PERFORMS { | fR

DSDelete,

tlma

SUPPLIER PERFORMS { | fR

DSFetch,

tlma

SUPPLIER PERFORMS { | fR

MessageStatus }

tlma

CONSUMER PERFORMS { | essageDeliver,

tlma

CONSUMER PERFORMS { | fR

ReceiptStatusNotice,

tlma

CONSUMER PERFORMS { | fR

DeliveryStatusNotice }

tlma

::= id-pt-mhs-doc-xfer

H.T. [T13.330]

TABLE 1/T.330

Operations of mhs-doc-xfer port

{
IPMS ports and telematic specific port
}
	mhs-doc-xfer port

Port	Abstract operation	Invoker	Performer	Abstract operation	Invoker	Performer
Origination (1) OriginateIPM (2) OriginateProbe (3) OriginateRN (4) CancelIPM }	{ TLM-User	 TLM-TER	 {			
(1) MessageSend (2) MessageProbe (3) ExplicitReceive (4) MessageCancel }	 TLM-TER	 TLMAU				
Reception (1) ReceiveIPM (2) ReceiveRN (3) ReceiveNRN (4) ReceiveReport }	{ TLM-TER	 User	 {			
(1) MessageDeliver (2) ReceiptStatusNotice (3) ReceiptStatusNotice (4) DeliveryStatusNotice }	 TLMAU	 TLM-TER				
Management (1) ChangeAutoDiscard (2) ChangeAutoAcknowledgment (3) ChangeAutoForwarding }	{ TLM-User	 TLM-TER	 {			
(1) Register (2) Register (3) Register }	 TLM-TER	 TLMAU				
Miscellanea (1) ChangeSubscriptionProfile (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	{ TLM-User	 TLM-TER	 {			
(1) Register (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	 TLM-TER	 TLMAU				

Tableau [T1.330], p.

The IPMS comprises any number of TLMA.

TLM users are communicants in telematic interworking. A TLM user originates or receives information objects whose types are specified in Recommendation X.420 and this Recommendation.

H.T. [T2.330]

tlm-user

OBJECT

tlma

PORTS { | rigination

[C],

tlma

PORTS { | fR

reception

[C],

tlma

PORTS { | fR

management

[C],

tlma

PORTS { | fR

miscellanea

[C] }

tlma

::= id-ot-tlm-user

H.T. [T3.330]

origination PORT

tlma

CONSUMER INVOKERS { | riginateProbe,

tlma

CONSUMER INVOKERS { | fR

OriginateIPM,

tlma

CONSUMER INVOKERS { | fR

OriginateRN,

tlma

CONSUMER INVOKERS { | fR

CancelIPM }

tlma

PORTS { | fR

::= id-pt-origination

reception PORT

tlma

CONSUMER INVOKERS { | eceiveReport,

tlma

CONSUMER INVOKERS { | fR

ReceiveIPM,

tlma

CONSUMER INVOKERS { | fR

ReceiveRN,

tlma

CONSUMER INVOKERS { | fR

ReceiveNRN }

tlma

PORTS { | fR

::= id-pt-reception

management PORT

tlma

CONSUMER INVOKERS { | hangeAutoDiscard,

tlma

CONSUMER INVOKERS { | fR

ChangeAutoAcknowledgment,

tlma

CONSUMER INVOKERS { | fR

ChangeAutoForwarding }

tlma

PORTS { | fR

::= id-pt-management

H.T. [T4.330]

miscellanea PORT

tlma

SUPPLIER PERFORMS { | hangeSubscriptionProfile,

tlma

SUPPLIER PERFORMS { | fR

DSList,

tlma

SUPPLIER PERFORMS { | fR

DSDelete,

tlma

SUPPLIER PERFORMS { | fR

DSFetch,

tlma

SUPPLIER PERFORMS { | fR

MessageStatus }

tlma

PORTS { | fR

::= id-pt-miscellanea

H.T. [T5.330]

ChangeSubscriptionProfile ::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | s-mode

[0] DSMode OPTIONAL,

tlma

ARGUMENT SET { | fR

error-recovery-mode

[1]

ErrorRecoveryMode OPTIONAL,

tlma

ARGUMENT SET { | fR

message-delete-mode

[2]

MessageDeleteMode OPTIONAL }

tlma

RESULT { }

tlma

ERRORS { | ame-error,

tlma

ERRORS { | fR

ds-error,

tlma

ERRORS { | fR

subscription-profile-error }

H.T. [T6.330]

DSList

::= ABSTRACT-OPERATION

tlma

ARGUMENT { }

tlma

RESULT SET { | 0] SET OF ListReport OPTIONAL }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

ListReport

::= SET { | etrieval-id

[0]

RetrievalIdentifier,

ListReport

::= SET { | fR

message-type

[1]

MessageType,

ListReport

::= SET { | fR

priority

[2]

Priority OPTIONAL,

ListReport

::= SET { | fR

message-length

[3]

MessageLength OPTIONAL,

ListReport

::= SET { | fR

originator-name

[4]

OrName OPTIONAL }

H.T. [T7.330]

DSDelete

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | elector [0] SET OF RetrievalIdentifier }

tlma

RESULT { }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

H.T. [T8.330]

DSFetch

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET OF { | etrieval-id

[0]

RetrievalIdentifier,

tlma

ARGUMENT SET OF { | fR

delete-after-output

[1]

DeleteAfterOutput OPTIONAL }

tlma

RESULT SET { | etrieval-id

[0] RetrievalIdentifier,

tlma

RESULT SET { | fR

message-report

[1]

MessageReport }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

H.T. [T9.330]

MessageStatus

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | 0] QueryIdentifier OPTIONAL }

tlma

RESULT SET { | eport-time

[0] DateandTime,

tlma

RESULT SET { | fR

reported-message-id

[1]

MessageIdentifier,

tlma

RESULT SET { | eported-message-id

[2]

SEQUENCE OF StatusInfo }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

message-status-error }

QueryIdentifier

::= CHOICE { | ubmission-id

[0] MessageIdentifier,

QueryIdentifier

::=

CHOICE { | fR

correlation-info

[1] CallIdentification }

StatusInfo

::= SET { | tatus

[0] Status,

StatusInfo

::= SET { | fR

per-recipient-info

[1]

PerRecipientReportDeliveryFields OPTIONAL }

H.T. [T10.330]

tlma-refinement REFINE tlma AS

tlmau

mhs-doc-xfer

[S] PAIRED with { | lm-ter }

tlm-ter

origination

[S] VISIBLE

tlm-ter

reception

[S] VISIBLE

tlm-ter

management

[S] VISIBLE

tlm-ter

miscellanea

[S] VISIBLE

tlm-ter

::= id-ref-secondary

H.T. [T11.330]

tlm-ter

OBJECT

tlma

PORTS { | rigination

[S],

tlma

PORTS { | fR

reception

[S],

tlma

PORTS { | fR

management

[S],

tlma

PORTS { | fR

miscellanea

[S],

tlma

PORTS { | fR

mhs-doc-xfer

[C] }

tlma

::= id-ot-tlm-ter

tlmau

OBJECT

tlma

PORTS { | hs-doc-xfer

[S],

tlma

PORTS { | fR

import

[C],

tlma

PORTS { | fR

export

[C] }

tlma

::= id-ot-tlm-user

H.T. [T12.330]

mhs-doc-xfer PORT

tlma

SUPPLIER PERFORMS { | essageSend,

tlma

SUPPLIER PERFORMS { | fR

MessageProbe,

tlma

SUPPLIER PERFORMS { | fR

ExplicitReceive,

tlma

SUPPLIER PERFORMS { | fR

MessageCancel,

tlma

SUPPLIER PERFORMS { | fR

Register,

tlma

SUPPLIER PERFORMS { | fR

DSList,

tlma

SUPPLIER PERFORMS { | fR

DSDelete,

tlma

SUPPLIER PERFORMS { | fR

DSFetch,

tlma

SUPPLIER PERFORMS { | fR

MessageStatus }

tlma

CONSUMER PERFORMS { | essageDeliver,

tlma

CONSUMER PERFORMS { | fR

ReceiptStatusNotice,

tlma

CONSUMER PERFORMS { | fR

DeliveryStatusNotice }

tlma

::= id-pt-mhs-doc-xfer

H.T. [T13.330]

TABLE 1/T.330

Operations of mhs-doc-xfer port

{
IPMS ports and telematic specific port
}
	mhs-doc-xfer port

Port	Abstract operation	Invoker	Performer	Abstract operation	Invoker	Performer
Origination (1) OriginateIPM (2) OriginateProbe (3) OriginateRN (4) CancelIPM }	{ TLM-User	 TLM-TER	 {			
(1) MessageSend (2) MessageProbe (3) ExplicitReceive (4) MessageCancel }	 TLM-TER	 TLMAU				
Reception (1) ReceiveIPM (2) ReceiveRN (3) ReceiveNRN (4) ReceiveReport }	{ TLM-TER	 User	 {			
(1) MessageDeliver (2) ReceiptStatusNotice (3) ReceiptStatusNotice (4) DeliveryStatusNotice }	 TLMAU	 TLM-TER				
Management (1) ChangeAutoDiscard (2) ChangeAutoAcknowledgment (3) ChangeAutoForwarding }	{ TLM-User	 TLM-TER	 {			
(1) Register (2) Register (3) Register }	 TLM-TER	 TLMAU				
Miscellanea (1) ChangeSubscriptionProfile (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	{ TLM-User	 TLM-TER	 {			
(1) Register (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	 TLM-TER	 TLMAU				

Tableau [T2.330], p.

A telematic user is associated with the TLMA by means of the origination, reception, management and miscellanea ports. A telematic user is a supplier [S] of no ports and a consumer [C] of all TLMA ports. The TLMA is a supplier of all TLMA ports and consumer of no ports.

The general access to IPMS is illustrated in Figure 2/T.330.

Figure 2/T.330, p.

An interpersonal messaging user agent (IPM-UA) is a secondary object that provides the interpersonal messaging abstract service (IPMAS) to a single IPM user. An IPM-UA is a specialized instance of the more general object, UA. An IPM-UA performs its function with help from the MTS.

A telematic agent (TLMA) is an object that provides the abstract service which comprises IPMAS and telematic specific abstract service, to a single TLM user. A TLMA is an instance of the more general object UA. A TLMA performs its function with help from the MTS.

A message transfer system (MTS), upon which all other IPMS components relay, is the provider of the message transfer abstract service (MTAS). It performs its function without assistance.

An interpersonal messaging system (IPMS) is the object by means of which all users communicate in interpersonal messaging.

The access unit (AU) could be a physical delivery access unit (PDAU), or telex access unit (TLXAU). The descriptions of these objects found in relevant Recommendations.

7.2 *Origination, reception and management ports, services and operations*

The abstract operations available at these ports, as described in X.420, are:

H.T. [T3.330]

origination PORT

tlma

CONSUMER INVOKERS { | riginateProbe,

tlma

CONSUMER INVOKERS { | fR

OriginateIPM,

tlma

CONSUMER INVOKERS { | fR

OriginateRN,

tlma

CONSUMER INVOKERS { | fR

CancelIPM }

tlma

PORTS { | fR

::= id-pt-origination

reception PORT

tlma

CONSUMER INVOKERS { | eceiveReport,

tlma

CONSUMER INVOKERS { | fR

ReceiveIPM,

tlma

CONSUMER INVOKERS { | fR

ReceiveRN,

tlma

CONSUMER INVOKERS { | fR

ReceiveNRN }

tlma

PORTS { | fR

::= id-pt-reception

management PORT

tlma

CONSUMER INVOKERS { | hangeAutoDiscard,

tlma

CONSUMER INVOKERS { | fR

ChangeAutoAcknowledgment,

tlma

CONSUMER INVOKERS { | fR

ChangeAutoForwarding }

tlma

PORTS { | fR

::= id-pt-management

H.T. [T4.330]

miscellanea PORT

tlma

SUPPLIER PERFORMS { | hangeSubscriptionProfile,

tlma

SUPPLIER PERFORMS { | fR

DSList,

tlma

SUPPLIER PERFORMS { | fR

DSDelete,

tlma

SUPPLIER PERFORMS { | fR

DSFetch,

tlma

SUPPLIER PERFORMS { | fR

MessageStatus }

tlma

PORTS { | fR

::= id-pt-miscellanea

H.T. [T5.330]

ChangeSubscriptionProfile ::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | s-mode

[0] DSMode OPTIONAL,

tlma

ARGUMENT SET { | fR

error-recovery-mode

[1]

ErrorRecoveryMode OPTIONAL,

tlma

ARGUMENT SET { | fR

message-delete-mode

[2]

MessageDeleteMode OPTIONAL }

tlma

RESULT { }

tlma

ERRORS { | ame-error,

tlma

ERRORS { | fR

ds-error,

tlma

ERRORS { | fR

subscription-profile-error }

H.T. [T6.330]

DSList

::= ABSTRACT-OPERATION

tlma

ARGUMENT { }

tlma

RESULT SET { | 0] SET OF ListReport OPTIONAL }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

ListReport

::= SET { | etrieval-id

[0]

RetrievalIdentifier,

ListReport

::= SET { | fR

message-type

[1]

MessageType,

ListReport

::= SET { | fR

priority

[2]

Priority OPTIONAL,

ListReport

::= SET { | fR

message-length

[3]

MessageLength OPTIONAL,

ListReport

::= SET { | fR

originator-name

[4]

OrName OPTIONAL }

H.T. [T7.330]

DSDelete

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | elector [0] SET OF RetrievalIdentifier }

tlma

RESULT { }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

H.T. [T8.330]

DSFetch

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET OF { | etrieval-id

[0]

RetrievalIdentifier,

tlma

ARGUMENT SET OF { | fR

delete-after-output

[1]

DeleteAfterOutput OPTIONAL }

tlma

RESULT SET { | etrieval-id

[0] RetrievalIdentifier,

tlma

RESULT SET { | fR

message-report

[1]

MessageReport }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

H.T. [T9.330]

MessageStatus

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | 0] QueryIdentifier OPTIONAL }

tlma

RESULT SET { | eport-time

[0] DateandTime,

tlma

RESULT SET { | fR

reported-message-id

[1]

MessageIdentifier,

tlma

RESULT SET { | eported-message-id

[2]

SEQUENCE OF StatusInfo }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

message-status-error }

QueryIdentifier

::= CHOICE { | ubmission-id

[0] MessageIdentifier,

QueryIdentifier

::=

CHOICE { | fR

correlation-info

[1] CallIdentification }

StatusInfo

::= SET { | tatus

[0] Status,

StatusInfo

::= SET { | fR

per-recipient-info

[1]

PerRecipientReportDeliveryFields OPTIONAL }

H.T. [T10.330]

tlma-refinement REFINE tlma AS

tlmau

mhs-doc-xfer

[S] PAIRED with { | lm-ter }

tlm-ter

origination

[S] VISIBLE

tlm-ter

reception

[S] VISIBLE

tlm-ter

management

[S] VISIBLE

tlm-ter

miscellanea

[S] VISIBLE

tlm-ter

::= id-ref-secondary

H.T. [T11.330]

tlm-ter

OBJECT

tlma

PORTS { | rigination

[S],

tlma

PORTS { | fR

reception

[S],

tlma

PORTS { | fR

management

[S],

tlma

PORTS { | fR

miscellanea

[S],

tlma

PORTS { | fR

mhs-doc-xfer

[C] }

tlma

::= id-ot-tlm-ter

tlmau

OBJECT

tlma

PORTS { | hs-doc-xfer

[S],

tlma

PORTS { | fR

import

[C],

tlma

PORTS { | fR

export

[C] }

tlma

::= id-ot-tlm-user

H.T. [T12.330]

mhs-doc-xfer PORT

tlma

SUPPLIER PERFORMS { | essageSend,

tlma

SUPPLIER PERFORMS { | fR

MessageProbe,

tlma

SUPPLIER PERFORMS { | fR

ExplicitReceive,

tlma

SUPPLIER PERFORMS { | fR

MessageCancel,

tlma

SUPPLIER PERFORMS { | fR

Register,

tlma

SUPPLIER PERFORMS { | fR

DSList,

tlma

SUPPLIER PERFORMS { | fR

DSDelete,

tlma

SUPPLIER PERFORMS { | fR

DSFetch,

tlma

SUPPLIER PERFORMS { | fR

MessageStatus }

tlma

CONSUMER PERFORMS { | essageDeliver,

tlma

CONSUMER PERFORMS { | fR

ReceiptStatusNotice,

tlma

CONSUMER PERFORMS { | fR

DeliveryStatusNotice }

tlma

::= id-pt-mhs-doc-xfer

H.T. [T13.330]

TABLE 1/T.330

Operations of mhs-doc-xfer port

{
IPMS ports and telematic specific port
}
	mhs-doc-xfer port

Port	Abstract operation	Invoker	Performer	Abstract operation	Invoker	Performer
Origination (1) OriginateIPM (2) OriginateProbe (3) OriginateRN (4) CancelIPM }	{ TLM-User	 TLM-TER	 {			
(1) MessageSend (2) MessageProbe (3) ExplicitReceive (4) MessageCancel }	 TLM-TER	 TLMAU				
Reception (1) ReceiveIPM (2) ReceiveRN (3) ReceiveNRN (4) ReceiveReport }	{ TLM-TER	 User	 {			
(1) MessageDeliver (2) ReceiptStatusNotice (3) ReceiptStatusNotice (4) DeliveryStatusNotice }	 TLMAU	 TLM-TER				
Management (1) ChangeAutoDiscard (2) ChangeAutoAcknowledgment (3) ChangeAutoForwarding }	{ TLM-User	 TLM-TER	 {			
(1) Register (2) Register (3) Register }	 TLM-TER	 TLMAU				
Miscellanea (1) ChangeSubscriptionProfile (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	{ TLM-User	 TLM-TER	 {			
(1) Register (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	 TLM-TER	 TLMAU				

Tableau [T3.330], p.

The abstract operations are fully described in Recommendation X.420.

7.3 *Miscellanea port services and operations*

Besides IPM abstract services, the following abstract services are available at the miscellanea port. They are provided by the TLMA object as the miscellanea abstract services.

H.T. [T4.330]

miscellanea PORT

tlma

SUPPLIER PERFORMS { | hangeSubscriptionProfile,

tlma

SUPPLIER PERFORMS { | fR

DSList,

tlma

SUPPLIER PERFORMS { | fR

DSDelete,

tlma

SUPPLIER PERFORMS { | fR

DSFetch,

tlma

SUPPLIER PERFORMS { | fR

MessageStatus }

tlma

PORTS { | fR

::= id-pt-miscellanea

H.T. [T5.330]

ChangeSubscriptionProfile ::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | s-mode

[0] DSMode OPTIONAL,

tlma

ARGUMENT SET { | fR

error-recovery-mode

[1]

ErrorRecoveryMode OPTIONAL,

tlma

ARGUMENT SET { | fR

message-delete-mode

[2]

MessageDeleteMode OPTIONAL }

tlma

RESULT { }

tlma

ERRORS { | ame-error,

tlma

ERRORS { | fR

ds-error,

tlma

ERRORS { | fR

subscription-profile-error }

H.T. [T6.330]

DSList

::= ABSTRACT-OPERATION

tlma

ARGUMENT { }

tlma

RESULT SET { | 0] SET OF ListReport OPTIONAL }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

ListReport

::= SET { | etrieval-id

[0]

RetrievalIdentifier,

ListReport

::= SET { | fR

message-type

[1]

MessageType,

ListReport

::= SET { | fR

priority

[2]

Priority OPTIONAL,

ListReport

::= SET { | fR

message-length

[3]

MessageLength OPTIONAL,

ListReport

::= SET { | fR

originator-name

[4]

OrName OPTIONAL }

H.T. [T7.330]

DSDelete

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | elector [0] SET OF RetrievalIdentifier }

tlma

RESULT { }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

H.T. [T8.330]

DSFetch

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET OF { | etrieval-id

[0]

RetrievalIdentifier,

tlma

ARGUMENT SET OF { | fR

delete-after-output

[1]

DeleteAfterOutput OPTIONAL }

tlma

RESULT SET { | etrieval-id

[0] RetrievalIdentifier,

tlma

RESULT SET { | fR

message-report

[1]

MessageReport }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

H.T. [T9.330]

MessageStatus

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | 0] QueryIdentifier OPTIONAL }

tlma

RESULT SET { | eport-time

[0] DateandTime,

tlma

RESULT SET { | fR

reported-message-id

[1]

MessageIdentifier,

tlma

RESULT SET { | eported-message-id

[2]

SEQUENCE OF StatusInfo }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

message-status-error }

QueryIdentifier

::= CHOICE { | ubmission-id

[0] MessageIdentifier,

QueryIdentifier

::=

CHOICE { | fR

correlation-info

[1] CallIdentification }

StatusInfo

::= SET { | tatus

[0] Status,

StatusInfo

::= SET { | fR

per-recipient-info

[1]

PerRecipientReportDeliveryFields OPTIONAL }

H.T. [T10.330]

tlma-refinement REFINE tlma AS

tlmau

mhs-doc-xfer

[S] PAIRED with { | lm-ter }

tlm-ter

origination

[S] VISIBLE

tlm-ter

reception

[S] VISIBLE

tlm-ter

management

[S] VISIBLE

tlm-ter

miscellanea

[S] VISIBLE

tlm-ter

::= id-ref-secondary

H.T. [T11.330]

tlm-ter

OBJECT

tlma

PORTS { | rigination

[S],

tlma

PORTS { | fR

reception

[S],

tlma

PORTS { | fR

management

[S],

tlma

PORTS { | fR

miscellanea

[S],

tlma

PORTS { | fR

mhs-doc-xfer

[C] }

tlma

::= id-ot-tlm-ter

tlmau

OBJECT

tlma

PORTS { | hs-doc-xfer

[S],

tlma

PORTS { | fR

import

[C],

tlma

PORTS { | fR

export

[C] }

tlma

::= id-ot-tlm-user

H.T. [T12.330]

mhs-doc-xfer PORT

tlma

SUPPLIER PERFORMS { | essageSend,

tlma

SUPPLIER PERFORMS { | fR

MessageProbe,

tlma

SUPPLIER PERFORMS { | fR

ExplicitReceive,

tlma

SUPPLIER PERFORMS { | fR

MessageCancel,

tlma

SUPPLIER PERFORMS { | fR

Register,

tlma

SUPPLIER PERFORMS { | fR

DSList,

tlma

SUPPLIER PERFORMS { | fR

DSDelete,

tlma

SUPPLIER PERFORMS { | fR

DSFetch,

tlma

SUPPLIER PERFORMS { | fR

MessageStatus }

tlma

CONSUMER PERFORMS { | essageDeliver,

tlma

CONSUMER PERFORMS { | fR

ReceiptStatusNotice,

tlma

CONSUMER PERFORMS { | fR

DeliveryStatusNotice }

tlma

::= id-pt-mhs-doc-xfer

H.T. [T13.330]

TABLE 1/T.330

Operations of mhs-doc-xfer port

{
IPMS ports and telematic specific port
}
	mhs-doc-xfer port

Port	Abstract operation	Invoker	Performer	Abstract operation	Invoker	Performer
Origination (1) OriginateIPM (2) OriginateProbe (3) OriginateRN (4) CancelIPM }	{ TLM-User	 TLM-TER	 {			
(1) MessageSend (2) MessageProbe (3) ExplicitReceive (4) MessageCancel }	 TLM-TER	 TLMAU				
Reception (1) ReceiveIPM (2) ReceiveRN (3) ReceiveNRN (4) ReceiveReport }	{ TLM-TER	 User	 {			
(1) MessageDeliver (2) ReceiptStatusNotice (3) ReceiptStatusNotice (4) DeliveryStatusNotice }	 TLMAU	 TLM-TER				
Management (1) ChangeAutoDiscard (2) ChangeAutoAcknowledgment (3) ChangeAutoForwarding }	{ TLM-User	 TLM-TER	 {			
(1) Register (2) Register (3) Register }	 TLM-TER	 TLMAU				
Miscellanea (1) ChangeSubscriptionProfile (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	{ TLM-User	 TLM-TER	 {			
(1) Register (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	 TLM-TER	 TLMAU				

Tableau [T4.330], p.

7.3.1 *ChangeSubscriptionProfile*

The ChangeSubscriptionProfile abstract operation enables a user to change the registered subscription profile which specifies relationship with the TLMAU, such as DS mode, error recovery mode and message delete mode.

H.T. [T5.330]

ChangeSubscriptionProfile ::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | s-mode

[0] DSMode OPTIONAL,

tlma

ARGUMENT SET { | fR

error-recovery-mode

[1]

ErrorRecoveryMode OPTIONAL,

tlma

ARGUMENT SET { | fR

message-delete-mode

[2]

MessageDeleteMode OPTIONAL }

tlma

RESULT { }

tlma

ERRORS { | ame-error,

tlma

ERRORS { | fR

ds-error,

tlma

ERRORS { | fR

subscription-profile-error }

H.T. [T6.330]

DSList

::= ABSTRACT-OPERATION

tlma

ARGUMENT { }

tlma

RESULT SET { | 0] SET OF ListReport OPTIONAL }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

ListReport

::= SET { | etrieval-id

[0]

RetrievalIdentifier,

ListReport

::= SET { | fR

message-type

[1]

MessageType,

ListReport

::= SET { | fR

priority

[2]

Priority OPTIONAL,

ListReport

::= SET { | fR

message-length

[3]

MessageLength OPTIONAL,

ListReport

::= SET { | fR

originator-name

[4]

OrName OPTIONAL }

H.T. [T7.330]

DSDelete

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | elector [0] SET OF RetrievalIdentifier }

tlma

RESULT { }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

H.T. [T8.330]

DSFetch

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET OF { | etrieval-id

[0]

RetrievalIdentifier,

tlma

ARGUMENT SET OF { | fR

delete-after-output

[1]

DeleteAfterOutput OPTIONAL }

tlma

RESULT SET { | etrieval-id

[0] RetrievalIdentifier,

tlma

RESULT SET { | fR

message-report

[1]

MessageReport }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

H.T. [T9.330]

MessageStatus

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | 0] QueryIdentifier OPTIONAL }

tlma

RESULT SET { | eport-time

[0] DateandTime,

tlma

RESULT SET { | fR

reported-message-id

[1]

MessageIdentifier,

tlma

RESULT SET { | eported-message-id

[2]

SEQUENCE OF StatusInfo }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

message-status-error }

QueryIdentifier

::= CHOICE { | ubmission-id

[0] MessageIdentifier,

QueryIdentifier

::=

CHOICE { | fR

correlation-info

[1] CallIdentification }

StatusInfo

::= SET { | tatus

[0] Status,

StatusInfo

::= SET { | fR

per-recipient-info

[1]

PerRecipientReportDeliveryFields OPTIONAL }

H.T. [T10.330]

tlma-refinement REFINE tlma AS

tlmau

mhs-doc-xfer

[S] PAIRED with { | lm-ter }

tlm-ter

origination

[S] VISIBLE

tlm-ter

reception

[S] VISIBLE

tlm-ter

management

[S] VISIBLE

tlm-ter

miscellanea

[S] VISIBLE

tlm-ter

::= id-ref-secondary

H.T. [T11.330]

tlm-ter

OBJECT

tlma

PORTS { | rigination

[S],

tlma

PORTS { | fR

reception

[S],

tlma

PORTS { | fR

management

[S],

tlma

PORTS { | fR

miscellanea

[S],

tlma

PORTS { | fR

mhs-doc-xfer

[C] }

tlma

::= id-ot-tlm-ter

tlmau

OBJECT

tlma

PORTS { | hs-doc-xfer

[S],

tlma

PORTS { | fR

import

[C],

tlma

PORTS { | fR

export

[C] }

tlma

::= id-ot-tlm-user

H.T. [T12.330]

mhs-doc-xfer PORT

tlma

SUPPLIER PERFORMS { | essageSend,

tlma

SUPPLIER PERFORMS { | fR

MessageProbe,

tlma

SUPPLIER PERFORMS { | fR

ExplicitReceive,

tlma

SUPPLIER PERFORMS { | fR

MessageCancel,

tlma

SUPPLIER PERFORMS { | fR

Register,

tlma

SUPPLIER PERFORMS { | fR

DSList,

tlma

SUPPLIER PERFORMS { | fR

DSDelete,

tlma

SUPPLIER PERFORMS { | fR

DSFetch,

tlma

SUPPLIER PERFORMS { | fR

MessageStatus }

tlma

CONSUMER PERFORMS { | essageDeliver,

tlma

CONSUMER PERFORMS { | fR

ReceiptStatusNotice,

tlma

CONSUMER PERFORMS { | fR

DeliveryStatusNotice }

tlma

::= id-pt-mhs-doc-xfer

H.T. [T13.330]

TABLE 1/T.330

Operations of mhs-doc-xfer port

{
IPMS ports and telematic specific port
}
	mhs-doc-xfer port

Port	Abstract operation	Invoker	Performer	Abstract operation	Invoker	Performer
Origination (1) OriginateIPM (2) OriginateProbe (3) OriginateRN (4) CancelIPM }	{ TLM-User	 TLM-TER	 {			
(1) MessageSend (2) MessageProbe (3) ExplicitReceive (4) MessageCancel }	 TLM-TER	 TLMAU				
Reception (1) ReceiveIPM (2) ReceiveRN (3) ReceiveNRN (4) ReceiveReport }	{ TLM-TER	 User	 {			
(1) MessageDeliver (2) ReceiptStatusNotice (3) ReceiptStatusNotice (4) DeliveryStatusNotice }	 TLMAU	 TLM-TER				
Management (1) ChangeAutoDiscard (2) ChangeAutoAcknowledgment (3) ChangeAutoForwarding }	{ TLM-User	 TLM-TER	 {			
(1) Register (2) Register (3) Register }	 TLM-TER	 TLMAU				
Miscellanea (1) ChangeSubscriptionProfile (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	{ TLM-User	 TLM-TER	 {			
(1) Register (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	 TLM-TER	 TLMAU				

Tableau [T5.330], p.

7.3.1.1 *Arguments of ChangeSubscriptionProfile*

This abstract operation has the following arguments:

a) DS-mode (C): The document storage mode to be applied. One of the following values:

1) retrieval: In the mode, the TLMAU holds the messages in the DS until they are explicitly deleted by the user;

2) auto output: In this mode, the TLMAU tries to output messages under user subscribed conditions after they are delivered to the DS.

b) Error-recovery-mode (C): This mode, whose recovery mechanism is defined in § 12 of this Recommendation has to be applied. (Recovery-1, 2 or 3.)

c) Message-delete-mode (C): Mode to be applied. One of the following values:

1) auto delete: In this mode, the messages in the DS are deleted as soon as they are output to the user by the performance of the DS fetch abstract operation with no delete-after-output argument (in case of retrieval mode), or by the automatically output (in case of auto-output mode);

2) manual delete: In this mode, the messages in the DS are held until the DS delete abstract operation or DS fetch abstract operation whose delete-after-output argument is ‘‘delete after output’’, will be carried out.

7.3.1.2 *Results of ChangeSubscriptionProfile*

This abstract operation has no results.

7.3.1.3 *Error of ChangeSubscriptionProfile*

This abstract operation has name-error, ds-error and subscription-profile error. These abstract errors are commonly described in § 9.

7.3.2 *DSList*

The DSList abstract operation enables a user to get a list of messages (IPMs, IPNs or reports) currently held in the document storage (DS).

H.T. [T6.330]

DSList

::= ABSTRACT-OPERATION

tlma

ARGUMENT { }

tlma

RESULT SET { | 0] SET OF ListReport OPTIONAL }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

ListReport

::= SET { | etrieval-id

[0]

RetrievalIdentifier,

ListReport

::= SET { | fR

message-type

[1]

MessageType,

ListReport

::= SET { | fR

priority

[2]

Priority OPTIONAL,

ListReport

::= SET { | fR

message-length

[3]

MessageLength OPTIONAL,

ListReport

::= SET { | fR

originator-name

[4]

OrName OPTIONAL }

H.T. [T7.330]

DSDelete

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | elector [0] SET OF RetrievalIdentifier }

tlma

RESULT { }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

H.T. [T8.330]

DSFetch

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET OF { | etrieval-id

[0]

RetrievalIdentifier,

tlma

ARGUMENT SET OF { | fR

delete-after-output

[1]

DeleteAfterOutput OPTIONAL }

tlma

RESULT SET { | etrieval-id

[0] RetrievalIdentifier,

tlma

RESULT SET { | fR

message-report

[1]

MessageReport }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

H.T. [T9.330]

MessageStatus

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | 0] QueryIdentifier OPTIONAL }

tlma

RESULT SET { | eport-time

[0] DateandTime,

tlma

RESULT SET { | fR

reported-message-id

[1]

MessageIdentifier,

tlma

RESULT SET { | eported-message-id

[2]

SEQUENCE OF StatusInfo }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

message-status-error }

QueryIdentifier

::= CHOICE { | ubmission-id

[0] MessageIdentifier,

QueryIdentifier

::=

CHOICE { | fR

correlation-info

[1] CallIdentification }

StatusInfo

::= SET { | tatus

[0] Status,

StatusInfo

::= SET { | fR

per-recipient-info

[1]

PerRecipientReportDeliveryFields OPTIONAL }

H.T. [T10.330]

tlma-refinement REFINE tlma AS

tlmau

mhs-doc-xfer

[S] PAIRED with { | lm-ter }

tlm-ter

origination

[S] VISIBLE

tlm-ter

reception

[S] VISIBLE

tlm-ter

management

[S] VISIBLE

tlm-ter

miscellanea

[S] VISIBLE

tlm-ter

::= id-ref-secondary

H.T. [T11.330]

tlm-ter

OBJECT

tlma

PORTS { | rigination

[S],

tlma

PORTS { | fR

reception

[S],

tlma

PORTS { | fR

management

[S],

tlma

PORTS { | fR

miscellanea

[S],

tlma

PORTS { | fR

mhs-doc-xfer

[C] }

tlma

::= id-ot-tlm-ter

tlmau

OBJECT

tlma

PORTS { | hs-doc-xfer

[S],

tlma

PORTS { | fR

import

[C],

tlma

PORTS { | fR

export

[C] }

tlma

::= id-ot-tlm-user

H.T. [T12.330]

mhs-doc-xfer PORT

tlma

SUPPLIER PERFORMS { | essageSend,

tlma

SUPPLIER PERFORMS { | fR

MessageProbe,

tlma

SUPPLIER PERFORMS { | fR

ExplicitReceive,

tlma

SUPPLIER PERFORMS { | fR

MessageCancel,

tlma

SUPPLIER PERFORMS { | fR

Register,

tlma

SUPPLIER PERFORMS { | fR

DSList,

tlma

SUPPLIER PERFORMS { | fR

DSDelete,

tlma

SUPPLIER PERFORMS { | fR

DSFetch,

tlma

SUPPLIER PERFORMS { | fR

MessageStatus }

tlma

CONSUMER PERFORMS { | essageDeliver,

tlma

CONSUMER PERFORMS { | fR

ReceiptStatusNotice,

tlma

CONSUMER PERFORMS { | fR

DeliveryStatusNotice }

tlma

::= id-pt-mhs-doc-xfer

H.T. [T13.330]

TABLE 1/T.330

Operations of mhs-doc-xfer port

{
IPMS ports and telematic specific port
}
	mhs-doc-xfer port

Port	Abstract operation	Invoker	Performer	Abstract operation	Invoker	Performer
Origination (1) OriginateIPM (2) OriginateProbe (3) OriginateRN (4) CancelIPM }	{ TLM-User	 TLM-TER	 {			
(1) MessageSend (2) MessageProbe (3) ExplicitReceive (4) MessageCancel }	 TLM-TER	 TLMAU				
Reception (1) ReceiveIPM (2) ReceiveRN (3) ReceiveNRN (4) ReceiveReport }	{ TLM-TER	 User	 {			
(1) MessageDeliver (2) ReceiptStatusNotice (3) ReceiptStatusNotice (4) DeliveryStatusNotice }	 TLMAU	 TLM-TER				
Management (1) ChangeAutoDiscard (2) ChangeAutoAcknowledgment (3) ChangeAutoForwarding }	{ TLM-User	 TLM-TER	 {			
(1) Register (2) Register (3) Register }	 TLM-TER	 TLMAU				
Miscellanea (1) ChangeSubscriptionProfile (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	{ TLM-User	 TLM-TER	 {			
(1) Register (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	 TLM-TER	 TLMAU				

Tableau [T6.330], p.

7.3.2.1 *Argument of DSList*

This abstract operation has no argument.

7.3.2.2 *Results of DSList*

This abstract-operation has the following results:

a) List-report: The characteristics of message held in DS.

1) Retrieval-id (M): The retrieval-id assigned to the message in DS.

2) Message-type (M): The type of message (IPM, RN, NRN or report).

3) Priority (C): The priority of the message (normal, non-urgent or urgent).

4) Message-length (C): The length of the message in octet.

5) Originator-name (C): The originator name of the message.

7.3.2.3 *Errors of DSList*

This abstract operation has subscription-error, name-error and ds-error. These abstract errors are described in § 9.

7.3.3 *DSDelete*

The DSDelete abstract operation enable a user to delete one or more specified messages in DS.

H.T. [T7.330]

DSDelete

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | elector [0] SET OF RetrievalIdentifier }

tlma

RESULT { }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

H.T. [T8.330]

DSFetch

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET OF { | etrieval-id

[0]

RetrievalIdentifier,

tlma

ARGUMENT SET OF { | fR

delete-after-output

[1]

DeleteAfterOutput OPTIONAL }

tlma

RESULT SET { | etrieval-id

[0] RetrievalIdentifier,

tlma

RESULT SET { | fR

message-report

[1]

MessageReport }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

H.T. [T9.330]

MessageStatus

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | 0] QueryIdentifier OPTIONAL }

tlma

RESULT SET { | eport-time

[0] DateandTime,

tlma

RESULT SET { | fR

reported-message-id

[1]

MessageIdentifier,

tlma

RESULT SET { | eported-message-id

[2]

SEQUENCE OF StatusInfo }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

message-status-error }

QueryIdentifier

::= CHOICE { | ubmission-id

[0] MessageIdentifier,

QueryIdentifier

::=

CHOICE { | fR

correlation-info

[1] CallIdentification }

StatusInfo

::= SET { | tatus

[0] Status,

StatusInfo

::= SET { | fR

per-recipient-info

[1]

PerRecipientReportDeliveryFields OPTIONAL }

H.T. [T10.330]

tlma-refinement REFINE tlma AS

tlmau

mhs-doc-xfer

[S] PAIRED with { | lm-ter }

tlm-ter

origination

[S] VISIBLE

tlm-ter

reception

[S] VISIBLE

tlm-ter

management

[S] VISIBLE

tlm-ter

miscellanea

[S] VISIBLE

tlm-ter

::= id-ref-secondary

H.T. [T11.330]

tlm-ter

OBJECT

tlma

PORTS { | rigination

[S],

tlma

PORTS { | fR

reception

[S],

tlma

PORTS { | fR

management

[S],

tlma

PORTS { | fR

miscellanea

[S],

tlma

PORTS { | fR

mhs-doc-xfer

[C] }

tlma

::= id-ot-tlm-ter

tlmau

OBJECT

tlma

PORTS { | hs-doc-xfer

[S],

tlma

PORTS { | fR

import

[C],

tlma

PORTS { | fR

export

[C] }

tlma

::= id-ot-tlm-user

H.T. [T12.330]

mhs-doc-xfer PORT

tlma

SUPPLIER PERFORMS { | essageSend,

tlma

SUPPLIER PERFORMS { | fR

MessageProbe,

tlma

SUPPLIER PERFORMS { | fR

ExplicitReceive,

tlma

SUPPLIER PERFORMS { | fR

MessageCancel,

tlma

SUPPLIER PERFORMS { | fR

Register,

tlma

SUPPLIER PERFORMS { | fR

DSList,

tlma

SUPPLIER PERFORMS { | fR

DSDelete,

tlma

SUPPLIER PERFORMS { | fR

DSFetch,

tlma

SUPPLIER PERFORMS { | fR

MessageStatus }

tlma

CONSUMER PERFORMS { | essageDeliver,

tlma

CONSUMER PERFORMS { | fR

ReceiptStatusNotice,

tlma

CONSUMER PERFORMS { | fR

DeliveryStatusNotice }

tlma

::= id-pt-mhs-doc-xfer

H.T. [T13.330]

TABLE 1/T.330

Operations of mhs-doc-xfer port

{
IPMS ports and telematic specific port
}
	mhs-doc-xfer port

Port	Abstract operation	Invoker	Performer	Abstract operation	Invoker	Performer
Origination (1) OriginateIPM (2) OriginateProbe (3) OriginateRN (4) CancelIPM }	{ TLM-User	 TLM-TER	 {			
(1) MessageSend (2) MessageProbe (3) ExplicitReceive (4) MessageCancel }	 TLM-TER	 TLMAU				
Reception (1) ReceiveIPM (2) ReceiveRN (3) ReceiveNRN (4) ReceiveReport }	{ TLM-TER	 User	 {			
(1) MessageDeliver (2) ReceiptStatusNotice (3) ReceiptStatusNotice (4) DeliveryStatusNotice }	 TLMAU	 TLM-TER				
Management (1) ChangeAutoDiscard (2) ChangeAutoAcknowledgment (3) ChangeAutoForwarding }	{ TLM-User	 TLM-TER	 {			
(1) Register (2) Register (3) Register }	 TLM-TER	 TLMAU				
Miscellanea (1) ChangeSubscriptionProfile (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	{ TLM-User	 TLM-TER	 {			
(1) Register (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	 TLM-TER	 TLMAU				

Tableau [T7.330], p.

7.3.3.1 *Arguments of DSDelete*

This abstract operation has the following arguments:

a) Selector (M): The selector is the list of the retrieval-id of messages that have to be deleted.

7.3.3.2 *Results of DSDelete*

This abstract operation has no results.

7.3.3.3 *Errors of DSDelete*

This abstract operation has subscription-error, name-error and ds-error. These abstract errors are described in § 9.

7.3.4 *DSFetch*

The DSFetch abstract operation enables a user to get one or more specified messages (IPMs, IPNs or reports) from DS.

H.T. [T8.330]

DSFetch

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET OF { | etrieval-id

[0]

RetrievalIdentifier,

tlma

ARGUMENT SET OF { | fR

delete-after-output

[1]

DeleteAfterOutput OPTIONAL }

tlma

RESULT SET { | etrieval-id

[0] RetrievalIdentifier,

tlma

RESULT SET { | fR

message-report

[1]

MessageReport }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

ds-error }

H.T. [T9.330]

MessageStatus

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | 0] QueryIdentifier OPTIONAL }

tlma

RESULT SET { | eport-time

[0] DateandTime,

tlma

RESULT SET { | fR

reported-message-id

[1]

MessageIdentifier,

tlma

RESULT SET { | eported-message-id

[2]

SEQUENCE OF StatusInfo }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

message-status-error }

QueryIdentifier

::= CHOICE { | ubmission-id

[0] MessageIdentifier,

QueryIdentifier

::=

CHOICE { | fR

correlation-info

[1] CallIdentification }

StatusInfo

::= SET { | tatus

[0] Status,

StatusInfo

::= SET { | fR

per-recipient-info

[1]

PerRecipientReportDeliveryFields OPTIONAL }

H.T. [T10.330]

tlma-refinement REFINE tlma AS

tlmau

mhs-doc-xfer

[S] PAIRED with { | lm-ter }

tlm-ter

origination

[S] VISIBLE

tlm-ter

reception

[S] VISIBLE

tlm-ter

management

[S] VISIBLE

tlm-ter

miscellanea

[S] VISIBLE

tlm-ter

::= id-ref-secondary

H.T. [T11.330]

tlm-ter

OBJECT

tlma

PORTS { | rigination

[S],

tlma

PORTS { | fR

reception

[S],

tlma

PORTS { | fR

management

[S],

tlma

PORTS { | fR

miscellanea

[S],

tlma

PORTS { | fR

mhs-doc-xfer

[C] }

tlma

::= id-ot-tlm-ter

tlmau

OBJECT

tlma

PORTS { | hs-doc-xfer

[S],

tlma

PORTS { | fR

import

[C],

tlma

PORTS { | fR

export

[C] }

tlma

::= id-ot-tlm-user

H.T. [T12.330]

mhs-doc-xfer PORT

tlma

SUPPLIER PERFORMS { | essageSend,

tlma

SUPPLIER PERFORMS { | fR

MessageProbe,

tlma

SUPPLIER PERFORMS { | fR

ExplicitReceive,

tlma

SUPPLIER PERFORMS { | fR

MessageCancel,

tlma

SUPPLIER PERFORMS { | fR

Register,

tlma

SUPPLIER PERFORMS { | fR

DSList,

tlma

SUPPLIER PERFORMS { | fR

DSDelete,

tlma

SUPPLIER PERFORMS { | fR

DSFetch,

tlma

SUPPLIER PERFORMS { | fR

MessageStatus }

tlma

CONSUMER PERFORMS { | essageDeliver,

tlma

CONSUMER PERFORMS { | fR

ReceiptStatusNotice,

tlma

CONSUMER PERFORMS { | fR

DeliveryStatusNotice }

tlma

::= id-pt-mhs-doc-xfer

H.T. [T13.330]

TABLE 1/T.330

Operations of mhs-doc-xfer port

{
IPMS ports and telematic specific port
}
	mhs-doc-xfer port

Port	Abstract operation	Invoker	Performer	Abstract operation	Invoker	Performer
Origination (1) OriginateIPM (2) OriginateProbe (3) OriginateRN (4) CancelIPM } (1) MessageSend (2) MessageProbe (3) ExplicitReceive (4) MessageCancel }	{ TLM-User TLM-TER	 TLM-TER TLMAU	 { 			
Reception (1) ReceiveIPM (2) ReceiveRN (3) ReceiveNRN (4) ReceiveReport } (1) MessageDeliver (2) ReceiptStatusNotice (3) ReceiptStatusNotice (4) DeliveryStatusNotice }	{ TLM-TER TLMAU	 User TLM-TER	 { 			
Management (1) ChangeAutoDiscard (2) ChangeAutoAcknowledgment (3) ChangeAutoForwarding } (1) Register (2) Register (3) Register }	{ TLM-User TLM-TER	 TLM-TER TLMAU	 { 			
Miscellanea (1) ChangeSubscriptionProfile (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus } (1) Register (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	{ TLM-User TLM-TER	 TLM-TER TLMAU	 { 			

Tableau [T8.330], p.

7.3.4.1 *Arguments of DSFetch*

This abstract operation has the following arguments:

a) Retrieval-id (M): The retrieval-id assigned to the message in DS.

b) Delete-after-output (C): This value indicates whether or not the message is deleted after retrieval. If this argument does not exist, registered mode, message-delete-mode, is applied.

7.3.4.2 *Results of DSFetch*

This abstract-operation has the following results:

a) Retrieval-id (M): The retrieval-id assigned to the message that was reported.

b) Message report (M): Envelope and content of reported message IPM, RN, NRN or report), assigned by retrieval-id.

7.3.4.3 *Errors of DSFetch*

This abstract operation has subscription-error, name-error and ds-error. These abstract errors are described in § 9.

7.3.5 *MessageStatus*

The MessageStatus abstract operation enables a user to get an information on the actual status of the previously submitted IPM.

H.T. [T9.330]

MessageStatus

::= ABSTRACT-OPERATION

tlma

ARGUMENT SET { | 0] QueryIdentifier OPTIONAL }

tlma

RESULT SET { | eport-time

[0] DateandTime,

tlma

RESULT SET { | fR

reported-message-id

[1]

MessageIdentifier,

tlma

RESULT SET { | eported-message-id

[2]

SEQUENCE OF StatusInfo }

tlma

ERRORS { | ubscription-error,

tlma

ERRORS { | fR

name-error,

tlma

ERRORS { | fR

message-status-error }

QueryIdentifier

::= CHOICE { | ubmission-id

[0] MessageIdentifier,

QueryIdentifier

::=

CHOICE { | fR

correlation-info

[1] CallIdentification }

StatusInfo

::= SET { | tatus

[0] Status,

StatusInfo

::= SET { | fR

per-recipient-info

[1]

PerRecipientReportDeliveryFields OPTIONAL }

H.T. [T10.330]

tlma-refinement REFINE tlma AS

tlmau

mhs-doc-xfer

[S] PAIRED with { | lm-ter }

tlm-ter

origination

[S] VISIBLE

tlm-ter

reception

[S] VISIBLE

tlm-ter

management

[S] VISIBLE

tlm-ter

miscellanea

[S] VISIBLE

tlm-ter

::= id-ref-secondary

H.T. [T11.330]

tlm-ter

OBJECT

tlma

PORTS { | rigination

[S],

tlma

PORTS { | fR

reception

[S],

tlma

PORTS { | fR

management

[S],

tlma

PORTS { | fR

miscellanea

[S],

tlma

PORTS { | fR

mhs-doc-xfer

[C] }

tlma

::= id-ot-tlm-ter

tlmau

OBJECT

tlma

PORTS { | hs-doc-xfer

[S],

tlma

PORTS { | fR

import

[C],

tlma

PORTS { | fR

export

[C] }

tlma

::= id-ot-tlm-user

H.T. [T12.330]

mhs-doc-xfer PORT

tlma

SUPPLIER PERFORMS { | essageSend,

tlma

SUPPLIER PERFORMS { | fR

MessageProbe,

tlma

SUPPLIER PERFORMS { | fR

ExplicitReceive,

tlma

SUPPLIER PERFORMS { | fR

MessageCancel,

tlma

SUPPLIER PERFORMS { | fR

Register,

tlma

SUPPLIER PERFORMS { | fR

DSList,

tlma

SUPPLIER PERFORMS { | fR

DSDelete,

tlma

SUPPLIER PERFORMS { | fR

DSFetch,

tlma

SUPPLIER PERFORMS { | fR

MessageStatus }

tlma

CONSUMER PERFORMS { | essageDeliver,

tlma

CONSUMER PERFORMS { | fR

ReceiptStatusNotice,

tlma

CONSUMER PERFORMS { | fR

DeliveryStatusNotice }

tlma

::= id-pt-mhs-doc-xfer

H.T. [T13.330]

TABLE 1/T.330

Operations of mhs-doc-xfer port

{
IPMS ports and telematic specific port
}
	mhs-doc-xfer port

Port	Abstract operation	Invoker	Performer	Abstract operation	Invoker	Performer
Origination (1) OriginateIPM (2) OriginateProbe (3) OriginateRN (4) CancelIPM } (1) MessageSend (2) MessageProbe (3) ExplicitReceive (4) MessageCancel }	{ TLM-User TLM-TER	 TLM-TER TLMAU	 {			
Reception (1) ReceiveIPM (2) ReceiveRN (3) ReceiveNRN (4) ReceiveReport } (1) MessageDeliver (2) ReceiptStatusNotice (3) ReceiptStatusNotice (4) DeliveryStatusNotice }	{ TLM-TER TLMAU	 User TLM-TER	 {			
Management (1) ChangeAutoDiscard (2) ChangeAutoAcknowledgment (3) ChangeAutoForwarding } (1) Register (2) Register (3) Register }	{ TLM-User TLM-TER	 TLM-TER TLMAU	 {			
Miscellanea (1) ChangeSubscriptionProfile (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus } (1) Register (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	{ TLM-User TLM-TER	 TLM-TER TLMAU	 {			

Tableau [T9.330], p.

7.3.5.1 *Arguments of MessageStatus*

This abstract operation has the following arguments:

a) Query-identifier (C): This identifier enables the TLMAU to identify the message whose status is being reported. Two types of query-identifiers are available:

1) submission-id (C): The message-id of the originated message whose status wants to query, returned as a result of the OriginateIPM abstract operation;

2) correlation-info (C): The call-identification of the originated message whose status wants to query.

7.3.5.2 *Results of MessageStatus*

This abstract operation has the following results:

a) Report-time (M): The date and time the report is made.

b) Message-id (M): The message-identifier of the originated message whose status is being reported, returned as a result of the OriginateIPM abstract operation.

c) Status-info (M): The status information of previously submitted messages.

1) Status: The status of the previously submitted IPM (in-process, delivered or non-delivered).

2) Per-recipient-info: Information about subject-message's status with respect to particular intended-recipients. A sequence of MTS per-recipient-field items, one for each recipient. This component does not exist until status component become delivered or non-delivered.

7.3.5.3 *Errors of MessageStatus*

This abstract operation has subscription-error, name-error and message-status-error. These abstract errors are described in § 9.

8 Refinement of the TLMA object

8.1 *Object and ports description*

In this Recommendation, the TLMA is refined further into secondary objects namely: the TLMA and the TLM-TER object.

H.T. [T10.330]

tlma-refinement REFINE tlma AS

tlmau

mhs-doc-xfer

[S] PAIRED with { | lm-ter }

tlm-ter

origination

[S] VISIBLE

tlm-ter

reception

[S] VISIBLE

tlm-ter

management

[S] VISIBLE

tlm-ter

miscellanea

[S] VISIBLE

tlm-ter

::= id-ref-secondary

H.T. [T11.330]

tlm-ter

OBJECT

tlma

PORTS { | rigination

[S],

tlma

PORTS { | fR

reception

[S],

tlma

PORTS { | fR

management

[S],

tlma

PORTS { | fR

miscellanea

[S],

tlma

PORTS { | fR

mhs-doc-xfer

[C] }

tlma

::= id-ot-tlm-ter

tlmau

OBJECT

tlma

PORTS { | hs-doc-xfer

[S],

tlma

PORTS { | fR

import

[C],

tlma

PORTS { | fR

export

[C] }

tlma

::= id-ot-tlm-user

H.T. [T12.330]

mhs-doc-xfer PORT

tlma

SUPPLIER PERFORMS { | essageSend,

tlma

SUPPLIER PERFORMS { | fR

MessageProbe,

tlma

SUPPLIER PERFORMS { | fR

ExplicitReceive,

tlma

SUPPLIER PERFORMS { | fR

MessageCancel,

tlma

SUPPLIER PERFORMS { | fR

Register,

tlma

SUPPLIER PERFORMS { | fR

DSList,

tlma

SUPPLIER PERFORMS { | fR

DSDelete,

tlma

SUPPLIER PERFORMS { | fR

DSFetch,

tlma

SUPPLIER PERFORMS { | fR

MessageStatus }

tlma

CONSUMER PERFORMS { | essageDeliver,

tlma

CONSUMER PERFORMS { | fR

ReceiptStatusNotice,

tlma

CONSUMER PERFORMS { | fR

DeliveryStatusNotice }

tlma

::= id-pt-mhs-doc-xfer

H.T. [T13.330]

TABLE 1/T.330

Operations of mhs-doc-xfer port

{
IPMS ports and telematic specific port
}
	mhs-doc-xfer port

Port	Abstract operation	Invoker	Performer	Abstract operation	Invoker	Performer
Origination (1) OriginateIPM (2) OriginateProbe (3) OriginateRN (4) CancelIPM } (1) MessageSend (2) MessageProbe (3) ExplicitReceive (4) MessageCancel }	{ TLM-User TLM-TER	 TLM-TER TLMAU	 {			
Reception (1) ReceiveIPM (2) ReceiveRN (3) ReceiveNRN (4) ReceiveReport } (1) MessageDeliver (2) ReceiptStatusNotice (3) ReceiptStatusNotice (4) DeliveryStatusNotice }	{ TLM-TER TLMAU	 User TLM-TER	 {			
Management (1) ChangeAutoDiscard (2) ChangeAutoAcknowledgment (3) ChangeAutoForwarding } (1) Register (2) Register (3) Register }	{ TLM-User TLM-TER	 TLM-TER TLMAU	 {			
Miscellanea (1) ChangeSubscriptionProfile (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus } (1) Register (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	{ TLM-User TLM-TER	 TLM-TER TLMAU	 {			

Tableau [T10.330], p.

The mhs-doc-xfer is a port that enables the interaction of the TLM-TER and the TLMAU.

Figure 3/T.330 illustrates refinement of TLMA.

Figure 3/T.330, p.

A telematic access unit (TLMAU) is a secondary object to the TLMA object. It provides a TLM-TER with access to any TLM user within the interpersonal messaging environment. (IPME: see Recommendation X.420.)

The TLM-TER is a secondary object to the TLMA object.

TLM-TERs are communicants in telematic interworking. A TLM-TER sends or receives documents, embodying information objects whose types are specified in Recommendation X.420 and this Recommendation.

TLM-TER shall be addressable by at least a Network address (see Recommendation X.402), and may also be addressed by one or more other forms of ORName.

H.T. [T11.330]

tlm-ter

OBJECT

tlma

PORTS { | rigination

[S],

tlma

PORTS { | fR

reception

[S],

tlma

PORTS { | fR

management

[S],

tlma

PORTS { | fR

miscellanea

[S],

tlma

PORTS { | fR

mhs-doc-xfer

[C] }

tlma

::= id-ot-tlm-ter

tlmau

OBJECT

tlma

PORTS { | hs-doc-xfer

[S],

tlma

PORTS { | fR

import

[C],

tlma

PORTS { | fR

export

[C] }

tlma

::= id-ot-tlm-user

H.T. [T12.330]

mhs-doc-xfer PORT

tlma

SUPPLIER PERFORMS { | essageSend,

tlma

SUPPLIER PERFORMS { | fR

MessageProbe,

tlma

SUPPLIER PERFORMS { | fR

ExplicitReceive,

tlma

SUPPLIER PERFORMS { | fR

MessageCancel,

tlma

SUPPLIER PERFORMS { | fR

Register,

tlma

SUPPLIER PERFORMS { | fR

DSList,

tlma

SUPPLIER PERFORMS { | fR

DSDelete,

tlma

SUPPLIER PERFORMS { | fR

DSFetch,

tlma

SUPPLIER PERFORMS { | fR

MessageStatus }

tlma

CONSUMER PERFORMS { | essageDeliver,

tlma

CONSUMER PERFORMS { | fR

ReceiptStatusNotice,

tlma

CONSUMER PERFORMS { | fR

DeliveryStatusNotice }

tlma

::= id-pt-mhs-doc-xfer

H.T. [T13.330]

TABLE 1/T.330

Operations of mhs-doc-xfer port

{
IPMS ports and telematic specific port
}
	mhs-doc-xfer port

Port	Abstract operation	Invoker	Performer	Abstract operation	Invoker	Performer
Origination (1) OriginateIPM (2) OriginateProbe (3) OriginateRN (4) CancelIPM } (1) MessageSend (2) MessageProbe (3) ExplicitReceive (4) MessageCancel }	{ TLM-User TLM-TER	 TLM-TER TLMAU	 {			
Reception (1) ReceiveIPM (2) ReceiveRN (3) ReceiveNRN (4) ReceiveReport } (1) MessageDeliver (2) ReceiptStatusNotice (3) ReceiptStatusNotice (4) DeliveryStatusNotice }	{ TLM-TER TLMAU	 User TLM-TER	 {			
Management (1) ChangeAutoDiscard (2) ChangeAutoAcknowledgment (3) ChangeAutoForwarding } (1) Register (2) Register (3) Register }	{ TLM-User TLM-TER	 TLM-TER TLMAU	 {			
Miscellanea (1) ChangeSubscriptionProfile (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus } (1) Register (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	{ TLM-User TLM-TER	 TLM-TER TLMAU	 {			

Tableau [T11.330], p.

The TLMA comprises one TLM terminal and one TLMAU.

8.2 *The mhs-doc-xfer port operations*

The following abstract operations are available at the mhs-doc-xfer port. The correspondence between mhs-doc-xfer port abstract operations and IPMS ports plus telematic specific port abstract operations are described in Table 1/T.330.

In this Recommendation TLM terminals implicitly bind a certain port at the time that the session is established and implicitly unbind a certain port at the time the session is released because Recommendation T.62 session procedure does not have association control.

H.T. [T12.330]

mhs-doc-xfer PORT

tlma

SUPPLIER PERFORMS { | essageSend,

tlma

SUPPLIER PERFORMS { | fR

MessageProbe,

tlma

SUPPLIER PERFORMS { | fR

ExplicitReceive,

tlma

SUPPLIER PERFORMS { | fR

MessageCancel,

tlma

SUPPLIER PERFORMS { | fR

Register,

tlma

SUPPLIER PERFORMS { | fR

DSList,

tlma

SUPPLIER PERFORMS { | fR

DSDelete,

tlma

SUPPLIER PERFORMS { | fR

DSFetch,

tlma

SUPPLIER PERFORMS { | fR

MessageStatus }

tlma

CONSUMER PERFORMS { | essageDeliver,

tlma

CONSUMER PERFORMS { | fR

ReceiptStatusNotice,

tlma

CONSUMER PERFORMS { | fR

DeliveryStatusNotice }

tlma

::= id-pt-mhs-doc-xfer

H.T. [T13.330]

TABLE 1/T.330

Operations of mhs-doc-xfer port

{
IPMS ports and telematic specific port
}
	mhs-doc-xfer port

Port	Abstract operation	Invoker	Performer	Abstract operation	Invoker	Performer
Origination (1) OriginateIPM (2) OriginateProbe (3) OriginateRN (4) CancelIPM } (1) MessageSend (2) MessageProbe (3) ExplicitReceive (4) MessageCancel }	{ TLM-User TLM-TER	 TLM-TER TLMAU	{ 			
Reception (1) ReceiveIPM (2) ReceiveRN (3) ReceiveNRN (4) ReceiveReport } (1) MessageDeliver (2) ReceiptStatusNotice (3) ReceiptStatusNotice (4) DeliveryStatusNotice }	{ TLM-TER TLMAU	 User TLM-TER	{ 			
Management (1) ChangeAutoDiscard (2) ChangeAutoAcknowledgment (3) ChangeAutoForwarding } (1) Register (2) Register (3) Register }	{ TLM-User TLM-TER	 TLM-TER TLMAU	{ 			
Miscellanea (1) ChangeSubscriptionProfile (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus } (1) Register (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	{ TLM-User TLM-TER	 TLM-TER TLMAU	{ 			

Tableau [T12.330], p.

H.T. [T13.330]

TABLE 1/T.330

Operations of mhs-doc-xfer port

{
IPMS ports and telematic specific port
}
	
	mhs-doc-xfer port

Port	Abstract operation	Invoker	Performer	Abstract operation	Invoker	Performer
Origination (1) OriginateIPM (2) OriginateProbe (3) OriginateRN (4) CancelIPM }	{ TLM-User	 TLM-TER	 {			
(1) MessageSend (2) MessageProbe (3) ExplicitReceive (4) MessageCancel }	 TLM-TER	 TLMAU				
Reception (1) ReceiveIPM (2) ReceiveRN (3) ReceiveNRN (4) ReceiveReport }	{ TLM-TER	 User	 {			
(1) MessageDeliver (2) ReceiptStatusNotice (3) ReceiptStatusNotice (4) DeliveryStatusNotice }	 TLMAU	 TLM-TER				
Management (1) ChangeAutoDiscard (2) ChangeAutoAcknowledgment (3) ChangeAutoForwarding }	{ TLM-User	 TLM-TER	 {			
(1) Register (2) Register (3) Register }	 TLM-TER	 TLMAU				
Miscellanea (1) ChangeSubscriptionProfile (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	{ TLM-User	 TLM-TER	 {			
(1) Register (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus }	 TLM-TER	 TLMAU				

Tableau 1/T.330 [T13.330], p.

8.2.1 *MessageSend*

MessageSend is the abstract operation at mhs-doc-xfer port that is invoked by TLM terminal to perform OriginateIPM abstract operation at TLM terminal. This abstract operation is used to submit the IPM from TLM terminal to TLMAU.

The description of OriginateIPM abstract operation is in Recommendation X.420.

8.2.2 *MessageProbe*

MessageProbe is the abstract operation at mhs-doc-xfer port that is invoked by TLM terminal to perform OriginateProbe abstract operation at TLM terminal. This abstract operation is used to determine whether or not this IPM could be delivered to one or more recipients.

The description of OriginateProbe abstract operation is in Recommendation X.420.

8.2.3 *ExplicitReceive*

ExplicitReceive is the abstract operation at mhs-doc-xfer port that is invoked by TLM terminal perform OriginateRN abstract operation at TLM terminal. This abstract operation is used to be originated by the actual-recipient of the subject IPM of whom RN is requested by means of notification-requests component of the subject IPM's recipient-specification.

The description of OriginateRN abstract operation is in Recommendation X.420.

8.2.4 *MessageCancel*

MessageCancel is the abstract operation at mhs-doc-xfer port that is invoked by TLM terminal to perform CancelIPM abstract operation at TLM terminal. This abstract operation is used to cancel if it can the delivery of previously originated message whose content is an IPM and for which deferred delivery was requested. There is no result in MessageCancel abstract operation.

The description of CancelIPM abstract operation is in Recommendation X.420.

8.2.5 *MessageDeliver*

MessageDeliver is the abstract operation at mhs-doc-xfer port that is invoked by TLMAU to perform ReceiveIPM at TLM terminal. This abstract operation is used to deliver the IPM from TLMAU to TLM terminal. There is no result or error in MessageDeliver abstract operation.

The description of ReceiveIPM abstract operation is in Recommendation X.420.

8.2.6 *ReceiptStatusNotice*

ReceiptStatusNotice is the abstract operation at mhs-doc-xfer port that is invoked by TLMAU to perform ReceiveRN or ReceiveNRN abstract operation at TLM terminal. This abstract operation is used to report the IPN that was invoked by an IPM originated by means of the MessageSend abstract operation. There is no result or error in ReceiptStatusNotice abstract operation.

The description of ReceiveRN or ReceiveNRN abstract operation is in Recommendation X.420.

8.2.7 *DeliveryStatusNotice*

DeliveryStatusNotice is the abstract operation at mhs-doc-xfer port that invoked by TLMAU to perform ReceiveReport abstract operation at TLM terminal. This abstract operation is used to deliver the DN that was invoked by a IPM originated by means of the MessageSend abstract operation. There is no result or error in DeliveryStatusNotice abstract operation.

The description of ReceiveReport abstract operation is in Recommendation X.420.

8.2.8 *Register*

Register is the abstract operation at mhs-doc-xfer port that is invoked by TLM terminal to perform all management port's abstract operations and ChangeSubscriptionProfile mode abstract operation. This abstract operation is used to register or change the parameters that will be kept on the parameter list of TLMAU.

The description of all management port's abstract operations is in Recommendation X.420 and ChangeSubscriptionProfile abstract operation found in § 7.3.1 of this Recommendation.

8.2.9 *DSList*

DSList is the abstract operation at mhs-doc-xfer port that is invoked by TLM terminal to perform DSList abstract operation at TLM terminal. This abstract operation is used to request the status list of a previously delivered IPMs, RNs, NRNs or reports.

The description of DSList abstract operation is in § 7.3.2 of this Recommendation.

8.2.10 *DSDelete*

DSDelete is the abstract operation at mhs-doc-xfer port that is invoked by TLM terminal to perform DSDelete abstract operation at TLM terminal, and is used to delete one or more messages from the DS. There is no result in DSDelete abstract operation.

The description of DSDelete abstract operation is in § 7.3.3 of this Recommendation.

8.2.11 *DSFetch*

DSFetch is the abstract operation at mhs-doc-xfer port that is invoked by TLM terminal to perform DSFetch abstract operation, and is used to fetch one specified message (IPM, RN, NRN or report), from the DS.

The description of DSFetch abstract operation is in § 7.3.4 of this Recommendation.

8.2.12 *MessageStatus*

MessageStatus is the abstract operation at mhs-doc-xfer port that invoked by TLM terminal to perform MessageStatus abstract operation. This abstract operation is used to know the status of previously submitted IPM by means of MessageSend abstract operation.

The description of MessageStatus abstract operation is in § 7.3.5 of this Recommendation.

9 Abstract errors

The abstract errors that may be reported in response to the invocation of abstract operations at the IPM's origination, reception and management ports are subscription error, name error and cancellation error, and in miscellanea port, subscription profile error, DS error and message status error. They are defined and described in the present section.

a) Subscription error

The subscription error abstract error reports that the user has not subscribed to one or more of the element of service implicit in his invocation of the abstract operation when performance is aborted.

The description of abstract error macro and abstract errors of subscription error is in Recommendation X.420.

b) Name error

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

The description of abstract error macro and abstract errors of name error is in Recommendation X.420.

c) Cancellation error

The cancellation error abstract error reports that the user's request to cancel the delivery of a message cannot be performed.

The description of abstract error macro and abstract errors of cancellation error is in Recommendation X.420.

d) Subscription profile error

The user’s request to change his subscription-prpfile cannot be performed, because one or more arguments proposed are inacceptable.

H.T. [T14.330]

subscription-profile-error

ABSTRACT-ERROR

tlma

PARAMETER SET { | roblem [0] SubscriptionProfileProblem }

tlma

::= 0

H.T. [T15.330]

SubscriptionProfileProblem ::= CHOICE { | 0] not-changed }

H.T. [T16.330]

ds-error

ABSTRACT-ERROR

tlma

PARAMETER SET { | roblem [0] DSProblem }

tlma

::= 1

H.T. [T17.330]

DSProblem

::= CHOICE { | 0] no-message-in-ds,

DSProblem

::= CHOICE { | fR

[1] ds-not-supported,

DSProblem

::= CHOICE { | fR

[2] ds-not-subscribed,

DSProblem

::= CHOICE { | fR

[3] retrieval-identifier-invalid,

DSProblem

::= CHOICE { | fR

[4] parameter-invalid }

H.T. [T18.330]

message-status-error

ABSTRACT-ERROR

tlma

PARAMETER SET { | roblem [0] MessageStatusProblem }

tlma

::= 2

H.T. [T19.330]

MessageStatusProblem ::= CHOICE { | 0] query-identifier-invalid }

Tableau [T14.330], p.

This abstract error has the following parameters:

1) Problem (M): The specific subscription profile related problem encountered.

H.T. [T15.330]

SubscriptionProfileProblem ::= CHOICE { | 0] not-changed }

H.T. [T16.330]

ds-error

ABSTRACT-ERROR

tlma

PARAMETER SET { | roblem [0] DSProblem }

tlma

::= 1

H.T. [T17.330]

DSProblem

::= CHOICE { | 0] no-message-in-ds,

DSProblem

::= CHOICE { | fR

[1] ds-not-supported,

DSProblem

::= CHOICE { | fR

[2] ds-not-subscribed,

DSProblem

::= CHOICE { | fR

[3] retrieval-identifier-invalid,

DSProblem

::= CHOICE { | fR

[4] parameter-invalid }

H.T. [T18.330]

message-status-error

ABSTRACT-ERROR

tlma

PARAMETER SET { | roblem [0] MessageStatusProblem }

tlma

::= 2

H.T. [T19.330]

MessageStatusProblem ::= CHOICE { | 0] query-identifier-invalid }

Tableau [T15.330], p.

This parameter may assume any one of the following values:

— not-changed: One or more subscription-profile arguments proposed are unacceptable, this abstract-operation is not performed.

e) DS error

The argument related DS cannot be performed because one or more arguments are improperly specified.

H.T. [T16.330]

ds-error

ABSTRACT-ERROR

tlma

PARAMETER SET { | roblem [0] DSProblem }

tlma

::= 1

H.T. [T17.330]

DSProblem

::= CHOICE { | 0] no-message-in-ds,

DSProblem

::= CHOICE { | fR

[1] ds-not-supported,

DSProblem

::= CHOICE { | fR

[2] ds-not-subscribed,

DSProblem

::= CHOICE { | fR

[3] retrieval-identifier-invalid,

DSProblem

::= CHOICE { | fR

[4] parameter-invalid }

H.T. [T18.330]

message-status-error

ABSTRACT-ERROR

tlma

PARAMETER SET { | roblem [0] MessageStatusProblem }

tlma

::= 2

H.T. [T19.330]

MessageStatusProblem ::= CHOICE { | 0] query-identifier-invalid }

Tableau [T16.330], p.

This abstract error has the following parameter:

1) Problem (M): The specific DS related problem encountered.

H.T. [T17.330]

DSProblem

::= CHOICE { | 0] no-message-in-ds,

DSProblem

::= CHOICE { | fR

[1] ds-not-supported,

DSProblem

::= CHOICE { | fR

[2] ds-not-subscribed,

DSProblem

::= CHOICE { | fR

[3] retrieval-identifier-invalid,

DSProblem

::= CHOICE { | fR

[4] parameter-invalid }

H.T. [T18.330]

message-status-error

ABSTRACT-ERROR

tlma

PARAMETER SET { | roblem [0] MessageStatusProblem }

tlma

::= 2

H.T. [T19.330]
MessageStatusProblem ::= CHOICE { | 0] query-identifier-invalid }
Tableau [T17.330], p.

This parameter may assume any one of the following values:

— no-message-in-ds: User requests to perform DS related abstract operation when there is no message in DS.

— ds-not-supported: User requests to perform DS related abstract-operation when TLMAU does not provide DS.

— ds-not-subscribed: User requests to perform DS related abstract-operation when he does not subscribe to DS.

— retrieval-identifier-invalid: The retrieval-id proposed is invalid.

— parameter-invalid: One or more arguments proposed are invalid.

f) MessageStatusError

No such message can be assigned by the query-identifier for message status abstract operation.

H.T. [T18.330]
message-status-error
ABSTRACT-ERROR
tlma
PARAMETER SET { | roblem [0] MessageStatusProblem }

tlma
::= 2
H.T. [T19.330]
MessageStatusProblem ::= CHOICE { | 0] query-identifier-invalid }
Tableau [T18.330], p.

This abstract-error has the following parameter:

1) Problem (M): The specific message status related problem encountered.

H.T. [T19.330]
MessageStatusProblem ::= CHOICE { | 0] query-identifier-invalid }
Tableau [T19.330], p.

This parameter may assume any one of the following values:

— query-identifier-invalid: The query-identifier proposal is unacceptable.

MONTAGE: § 10 SUR LE RESTE DE CETTE PAGE