



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

T.330

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

**TERMINAL EQUIPMENT AND PROTOCOLS FOR
TELEMATIC SERVICES**

**TELEMATIC ACCESS TO INTERPERSONAL
MESSAGE SYSTEM**

ITU-T Recommendation T.330

(Extract from the *Blue Book*)

NOTES

1 ITU-T Recommendation T.330 was published in Fascicle VII.5 of the *Blue Book*. This file is an extract from the *Blue Book*. While the presentation and layout of the text might be slightly different from the *Blue Book* version, the contents of the file are identical to the *Blue Book* version and copyright conditions remain unchanged (see below).

2 In this Recommendation, the expression “Administration” is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

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	<i>Introduction</i>
1	<i>Scope and field of application</i>
2	<i>References</i>
3	<i>Definitions</i>
4	<i>Abbreviations</i>
5	<i>Conventions</i>
6	<i>Overview of telematic access to IPMS</i>
	6.1 Abstract model
	6.2 Functional model
	6.3 Access for registered and non-registered users
7	<i>IPMS in the context of telematic interworking</i>
	7.1 Objects and ports description
	7.2 Origination, reception and management ports, services and operations
	7.3 Miscellanea port services and operations
8	<i>Refinement of the TLMA object</i>
	8.1 Object and ports description
	8.2 The mhs-doc-xfer port operations

9	<i>Abstract errors</i>
10	<i>Realization of abstract operations</i>
	10.1 Description of TAPDU
	10.2 Operation of the TLMAU
11	<i>Formats and coding of TAPDU</i>
	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	<i>Error recovery</i>
13	<i>Control procedures</i>
	13.1 Session control procedures
	13.2 Document control procedures
	13.3 Log-on procedures
	<i>Annex A - Formal definition of TLMA abstract service</i>
	<i>Annex B - Format of TAPDU components</i>
	<i>Annex C - Element ID list</i>
	<i>Annex D - Element of service for TTX/IPM service intercommunications</i>

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
S	Message store
T	Message transfer
TA	Message transfer agent
TAS	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:

- to specify the functional objects, the OBJECT and REFINE macros and associated conventions of Recommendation X.407;
- to specify the information objects (and other data types and values of all kinds), ASN.1 itself;
- 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:

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

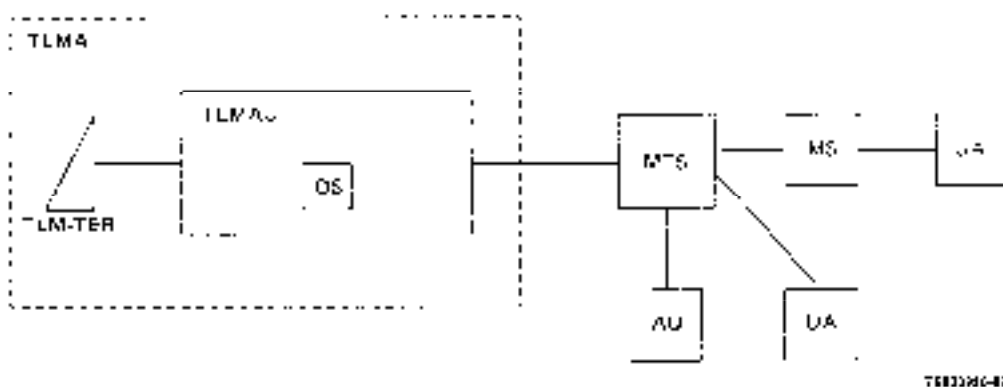


FIGURE 1/T.330

Telematic access to IPMS: Functional model

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.

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

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.


```

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

```

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.

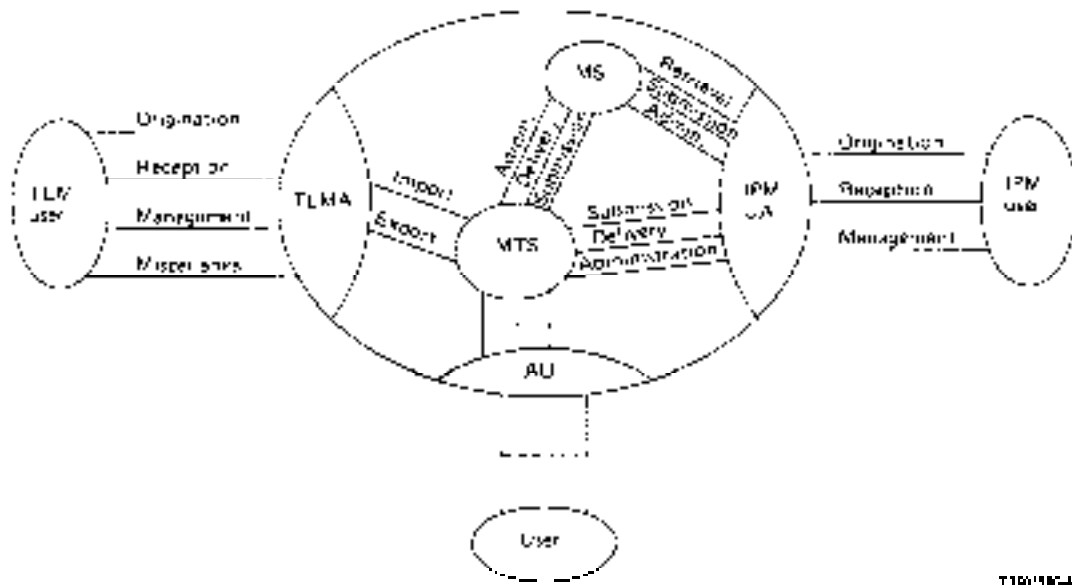


FIGURE 2/T.330
General access to IPMS

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:

origination PORT

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

reception PORT

```
CONSUMER INVOKERS { ReceiveReport,  
                    ReceiveIPM,  
                    ReceiveRN,  
                    ReceiveNRN }  
  
:: = id-pt-reception
```

management PORT

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

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.

miscellanea PORT

```
SUPPLIER PERFORMS { ChangeSubscriptionProfile,  
                    DSLList,  
                    DSDelete,  
                    DSFetch,  
                    MessageStatus }  
  
:: = id-pt-miscellanea
```

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.

ChangeSubscriptionProfile :: = ABSTRACT-OPERATION

```
ARGUMENT SET { ds-mode           [0] DSMode OPTIONAL,  
               error-recovery-mode [1] ErrorRecoveryMode OPTIONAL,  
               message-delete-mode [2] MessageDeleteMode OPTIONAL }  
  
RESULT { }  
  
ERRORS { name-error,  
         ds-error,  
         subscription-profile-error }
```

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).

DSList ::= ABSTRACT-OPERATION

ARGUMENT{ }
 RESULT SET{[0] SET OF ListReport OPTIONAL }
 ERRORS { subscription-error,
 name-error,
 ds-error }

ListReport ::= SET { retrieval-id [0] RetrievalIdentifier,
 message-type [1] MessageType,
 priority [2] Priority OPTIONAL,
 message-length [3] MessageLength OPTIONAL,
 originator-name [4] OrName OPTIONAL }

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.

DSDelete ::= ABSTRACT-OPERATION

ARGUMENT SET { selector [0] SET OF RetrievalIdentifier }
RESULT{ }
ERRORS { subscription-error,
 name-error,
 ds-error }

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.

DSFetch ::= ABSTRACT-OPERATION

ARGUMENT SET OF { retrieval-id [0] RetrievalIdentifier,
 delete-after-output [1] DeleteAfterOutput OPTIONAL }
RESULT SET { retrieval-id [0] RetrievalIdentifier,
 message-report [1] MessageReport }
ERRORS { subscription-error,
 name-error,
 ds-error }

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.

MessageStatus ::= ABSTRACT-OPERATION

ARGUMENT SET { [0] QueryIdentifier OPTIONAL }

RESULT SET { report-time [0] DateandTime,
reported-message-id [1] MessageIdentifier,
[2] SEQUENCE OF StatusInfo }

ERRORS { subscription-error,
name-error,
message-status-error }

QueryIdentifier ::= CHOICE { submission-id [0] MessageIdentifier,
correlation-info [1] CallIdentification }

StatusInfo ::= SET { status [0] Status,
per-recipient-info [1] PerRecipientReportDeliveryFields OPTIONAL }

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.

```

tлма-refinement REFINE tлма AS
  tлмаu          mhs-doc-xfer      [S] PAIRED with { tlm-ter }
  tlm-ter        origination       [S] VISIBLE
                 reception        [S] VISIBLE
                 management       [S] VISIBLE
                 miscellanea      [S] VISIBLE
  ::= id-ref-secondary

```

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.

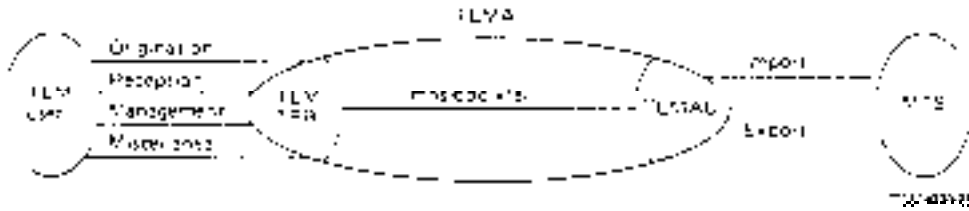


FIGURE 3/T.330

Refinement of TLMA

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.

```
tlm-ter          OBJECT
  PORTS { origination [S],
           reception  [S],
           management [S],
           miscellanea [S],
           mhs-doc-xfer [C] }
  ::= id-ot-tlm-ter
```

```
tlmau           OBJECT
  PORTS { mhs-doc-xfer [S],
           import      [C],
           export      [C] }
  ::= id-ot-tlm-user
```

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.

```
mhs-doc-xfer PORT
  SUPPLIER PERFORMS { MessageSend,
                     MessageProbe,
                     ExplicitReceive,
                     MessageCancel,
                     Register,
                     DSList,
                     DSDelete,
                     DSFetch,
                     MessageStatus }
  CONSUMER PERFORMS { MessageDeliver,
                     ReceiptStatusNotice,
                     DeliveryStatusNotice }
  ::= id-pt-mhs-doc-xfer
```

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) ChangeAutoAcknow- ledgment (3) ChangeAutoForwarding	TLM-User	TLM-TER	(1) Register (2) Register (3) Register	TLM-TER	TLMAU
Miscellanea	(1) ChangeSubscriptionPro- file (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

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 *DSLlist*

DSLlist is the abstract operation at mhs-doc-xfer port that is invoked by TLM terminal to perform DSLlist 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 DSLlist 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 miscellaneous 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-profile cannot be performed, because one or more arguments proposed are unacceptable.

subscription-profile-error ABSTRACT-ERROR

```
PARAMETER SET { problem [0] SubscriptionProfileProblem }
:: = 0
```

This abstract error has the following parameters:

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

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

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.

ds-error ABSTRACT-ERROR

```
PARAMETER SET { problem [0] DSProblem }
:: = 1
```

This abstract error has the following parameter:

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

```
DSProblem :: = CHOICE { [0] no-message-in-ds,
                        [1] ds-not-supported,
                        [2] ds-not-subscribed,
                        [3] retrieval-identifier-invalid,
                        [4] parameter-invalid }
```

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.

```
message-status-error  ABSTRACT-ERROR
                      PARAMETER SET { problem [0] MessageStatusProblem }
                      ::= 2
```

This abstract-error has the following parameter:

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

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

This parameter may assume any one of the following values:

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

10 Realization of abstract operations

How a TLMAU realizes the mhs-doc-xfer port by means of which it interacts with a TLM terminal is specified in this section. But how a TLMA realizes the ports by means of which it interacts with a TLM user and MTS is outside the scope of this Recommendation.

Telematic access protocol for accessing to IPMS, called P5 protocol, is provided to realize the interaction, which means abstract operations performed at the mhs-doc-xfer port, between a TLMAU and a TLM terminal. The concrete interactions, which correspond to abstract operations, are realized as telematic access protocol data units (TAPDUs).

It should be noted that the TLMAU may not support all the conditional TAPDUs and all the optional elements or parameters of a TAPDU. The actual support of the TAPDUs and parameters depends on the application and the version of the colocated MTA.

The relationship between abstract operations at the mhs-doc-xfer port and associated TAPDUs are summarized in Table 2/T.330.

10.1 *Description of TAPDU*

10.1.1 *MessageSend*

A TLM terminal sends a Send-TAPDU to invoke the MessageSend abstract operation. The TLMAU returns a SendAck-TAPDU to report the result of that operation, or may return an Exception-TAPDU (§ 10.1.1.3) to report an abstract error.

TABLE 2/T.330

Relationship between abstract operation and TAPDU

mhs-doc-xfer Abstract operation	TAPDU		Direction of transfer
Operation	TAPDU Name	TLMAU Status	TLM ↔ TLMAU
MessageSend	(O) Send-TAPDU (R) SendAck-TAPDU (E) Exception-TAPDU	M C M	→ ← ←
MessageProbe	(O) Probe-TAPDU (R) ProbeAck-TAPDU (E) Exception-TAPDU	C C C	→ ← ←
ExplicitReceive	(O) ExplicitRN-TAPDU (R) ExplicitRNAck-TAPDU (E) Exception-TAPDU	C C C	→ ← ←
MessageCancel	(O) Cancel-TAPDU (R) – (E) Exception-TAPDU	C – C	→ – ←
MessageDeliver	(O) Deliver-TAPDU	M	←
ReceiptStatusNotice	(O) ReceiptStatusNotice-TAPDU	M	←
DeliveryStatusNotice	(O) DeliveryStatusNotice-TAPDU	M	←
Register	(O) Register-TAPDU (R) RegisterAck-TAPDU (E) Exception-TAPDU	C C C	→ ← ←
DSLlist	(O) DSQuery-TAPDU (R) DSReport-TAPDU (E) Exception-TAPDU	C C C	→ ← ←
DSDelete	(O) MessageDelete-TAPDU (R) – (E) Exception-TAPDU	C – C	→ – ←
DSFetch	(O) OutputRequest-TAPDU (R) OutputMessage-TAPDU (E) Exception-TAPDU	C (Note 1) C (Note 1) C (Note 1)	→ ← ←
MessageStatus	(O) StatusQuery-TAPDU (R) StatusReport-TAPDU (E) Exception-TAPDU	C C C	→ ← ←

O Argument R Result E Error

M Mandatory C Conditional

Note 1 – In cases where TLMAU provides DS, these TAPDU are mandatory.

Note 2 – A message may arrive at a TLM terminal as a result of either a Deliver-TAPDU or OutputMessage-TAPDU. The Deliver-TAPDU is applicable when delivery occurs directly to a TLM terminal. The OutputMessage-TAPDU is only applicable in the case that DS is subscribed.

10.1.1.1 Send-TAPDU

The Send-TAPDU comprises the following elements:

Send-TAPDU	
<pre> Send-TAPDU ::= SEQUENCE { [0] SEQUENCE { send [0] SendTAPDUId, [1] SEQUENCE { quantityOfDocs number-of-docs } -- See Note 1 } [1] SET { [0] SEQUENCE { priority priority-ind } [1] SEQUENCE { perMessageIndicators SEQUENCE { deferred-delivery-time [0] DateandTime OPTIONAL, [1] SET { disclose-recipients [0] DiscloseRecipientsValue OPTIONAL, alternate-recipient-allowed [1] AlternateRecipientAllowedValue OPTIONAL, recipient-reassignment-prohibited [2] ReassignmentValue OPTIONAL } } OPTIONAL, } [2] SEQUENCE { conversion conversion-info } [3] SEQUENCE { contentinfo content-return-request } [4] SEQUENCE { returnAddress postal-address } [5] SEQUENCE { latestdelivery latest-delivery-time } [6] SEQUENCE { to SET OF SEQUENCE { primary-recipient [0] ORDescriptor, [1] RecOptions } } OPTIONAL, } -- See Note 2 } [7] SEQUENCE { cc SET OF SEQUENCE { copy-recipient [0] ORDescriptor, [1] RecOptions } } OPTIONAL, } -- See Note 2 } [8] SEQUENCE { bcc SET OF SEQUENCE { blind-copy-recipient [0] ORDescriptor, [1] RecOptions } } OPTIONAL, } -- See Note 2 } } </pre>	<pre> QuantityOfDocsElementId, NumberOfAssociatedDocuments } OPTIONAL }, PriorityElementId, PriorityValue DEFAULT normal } OPTIONAL, PerMessageIndicatorsElementId, DateandTime OPTIONAL, DiscloseRecipientsValue OPTIONAL, AlternateRecipientAllowedValue OPTIONAL, ReassignmentValue OPTIONAL } } OPTIONAL, ConversionElementId, ConversionInfoValue } OPTIONAL, ContentInfoElementId, ContentReturnRequestValue } OPTIONAL, ReturnAddressElementId, PostalAddressValue OPTIONAL, LatestDeliveryElementId, DateandTime } OPTIONAL }, ToElementId, ORDescriptor, RecOptions } } OPTIONAL, CCElementId, ORDescriptor, RecOptions } } OPTIONAL, BCCElementId, ORDescriptor, RecOptions } } OPTIONAL }, </pre>

Send-TAPDU (*continued*)

-- *Send-TAPDU Definition (continued)*

```

[2] SET {
    [0] SEQUENCE {
        thisIPM                ThisIPMElementId,
        this-ipm-id            IPMIdentifier } OPTIONAL,
    -- See Note 3
    [1] SEQUENCE {
        from                    FromElementId,
        originating-user       ORDescriptor } OPTIONAL,
    [2] SEQUENCE {
        authorizing             AuthorizingElementId,
        SET OF
        authorizing-user       OrDescriptor } OPTIONAL,
    [3] SEQUENCE {
        repliedToIPM           RepliedToIPMElementId,
        replied-to-ipm-id      IPMIdentifier } OPTIONAL,
    [4] SEQUENCE {
        obsoletedIPMs          ObsoletedIPMsElementId,
        SEQUENCE OF
        obsoleted-ipm-id      IPMIdentifier } OPTIONAL,
    [5] SEQUENCE {
        relatedIPMs            RelatedIPMsElementId,
        SEQUENCE OF
        related-ipm-id        IPMIdentifier } OPTIONAL,
    [6] SEQUENCE {
        subject                 SubjectElementId,
        subject-content         SubjectContent } OPTIONAL,
    [7] SEQUENCE {
        contentIndicator        ContentIndicatorElementId,
        SEQUENCE {
        expiry-time [0] DateandTime OPTIONAL,
        [1] SET {
        importance [0] ImportanceValue DEFAULT normal,
        sensitivity [1] SensitivityValue OPTIONAL }}} OPTIONAL,
    [8] SEQUENCE {
        reply                   ReplyElementId,
        SEQUENCE {
        reply-time [0] DateandTime,
        [1] SET OF {
        reply-recipient         ORDescriptor }} OPTIONAL,
    [9] SEQUENCE {
        language                LanguageElementId,
        language-ind            LanguageInd } OPTIONAL,
MsgIncomplete [10] MsgIncompleteElementId OPTIONAL },
    -- Body
    [3] SET {
        [0] SEQUENCE {
            BodyType            BodyTypeElementId,
            SET OF {
            Body-part          BodyPartValue } OPTIONAL }}

```

Send-TAPDU (*end*)

-- *Send-TAPDU Definition (continued)*

-- *Definition of RecOptions*

RecOptions ::= SET {

user-report-request	[1] UserReportRequestValue OPTIONAL,
explicit-conversion	[2] ExplicitConversionValue OPTIONAL OPTIONAL,
rn-request	[3] RNRequestValue OPTIONAL,
nrn-request	[4] NRNRequestValue OPTIONAL,
return-request	[5] ReturnRequestValue OPTIONAL,
reply-request	[6] ReplyRequestValue DEFAULT noReply,
requested-delivery-method	[7] RequestedDelValue OPTIONAL,
terminal-type	[8] TerminalTypeValue OPTIONAL,
physical-forwarding-prohibited	[9] PhyForProhibValue OPTIONAL,
physical-forwarding-address-request	[10] PhyforAdrValue OPTIONAL,
physical-delivery-modes	[11] PhyDelModValue OPTIONAL,
registered-mail-type	[12] RegMailTypeValue OPTIONAL,
recipient-number-for-advice	[13] RecNumAdvValue OPTIONAL,
physical-delivery-report-request	[14] PhyDelRepValue OPTIONAL,
originator-requested-alternate-recipient	[15] OrgRecAltValue OPTIONAL,

Note 1 – This element must be present when ControllInfo is conveyed by a normal document and more than one TAPDU are conveyed in this session.

Note 2 – OR Descriptor must contain an ORAddress and at least one of these addresses must be present.

Note 3 – When this element is omitted, the TLMAU shall construct this component which consists of the following components: originator name, date and time, and if necessary, a sequence number.

10.1.1.2 *SendAck-TAPDU*

The SendAck-TAPDU comprises the following elements:

SendAck-TAPDU	
<pre> SendAck-TAPDU ::= SEQUENCE { [0] SEQUENCE { sendAck [0] SendAckTAPDUId, [1] SEQUENCE { correlationInfo CorrelationInfoElementId, call-id CallIdentification }, -- See Note [1] SET { [0] SEQUENCE { submissionId SubmissionIdElementId, submission-msg-id MessageIdentifier }, [1] SEQUENCE { submissionTime SubmissionTimeElementId, submission-time DateandTime }}} </pre>	

Note – This element is a session connection information that identifies previous Send-TAPDU being reported on.

10.1.1.3 *Exception-TAPDU*

The Exception- TAPDU comprises the following elements:

Exception-TAPDU	
<pre> Exception-TAPDU ::= SEQUENCE { [0] SEQUENCE { exception [0] ExceptionTAPDUId, [1] SEQUENCE { correlationInfo CorrelationInfoElementId, call-id CallIdentification }, -- See Note [2] SEQUENCE { errors ErrorsElementId, error-cause ErrorCauseValue }}} </pre>	

Note – This element is a session connection information that identifies associated TAPDU being reported on e.g. Send-TAPDU.

10.1.2 *MessageProbe*

A TLM terminal sends a Probe-TAPDU to invoke the MessageProbe abstract operation. The TLMAU returns a ProbeAck-TAPDU to report the result of that operation, or may return an Exception-TAPDU (§ 10.1.1.3) to report an abstract error.

10.1.2.1 Probe-TAPDU

The Probe-TAPDU comprises the following elements:

Probe-TAPDU
<pre>Probe-TAPDU ::= SEQUENCE { [0] SEQUENCE { probe [0] ProbeTAPDUId, [1] SEQUENCE { quantityOfDocs QuantityOfDocsElementId, number-of-docs NumberOfAssociatedDocuments } OPTIONAL }, [1] SET { -- Continuation see send-TAPDU. -- Note that only few elements of the Send-TAPDU are relevant for a Probe-TAPDU. -- Not relevant elements will be ignored. -- At least one recipient must be present.</pre>

10.1.2.2 ProbeAck-TAPDU

The ProbeAck-TAPDU comprises the following elements:

ProbeAck-TAPDU
<pre>ProbeAck-TAPDU ::= SEQUENCE { [0] SEQUENCE { probeAck [0] ProbeAckTAPDUId, [1] SEQUENCE { correlationInfo CorrelationInfoElementId, call-id CallIdentification }}, [1] SET { [0] SEQUENCE { probeId ProbeElementId, probe-msg-id MessageIdentifier }, [1] SEQUENCE { submissionTime SubmissionTimeElementId, submission-time DateandTime }}}}</pre>

10.1.3 ExplicitReceive

A TLM terminal sends an ExplicitRN-TAPDU to invoke the ExplicitReceive abstract operation. The TLMAU returns an ExplicitRNAck-TAPDU to report the result of that operation, or may return an Exception-TAPDU (see § 10.1.1.3), to report an abstract error.

10.1.3.1 *ExplicitRN-TAPDU*

The ExplicitRN-TAPDU comprises the following elements:

ExplicitRN-TAPDU	
ExplicitRN-TAPDU ::= SEQUENCE {	
[0]	explicitRN
[1] SET {	ExplicitRNTAPDUId,
[0] SEQUENCE {	RecipientsElementId,
recipients	ORName },
recipient-name	
[1] SEQUENCE {	PriorityElementId,
priority	PriorityValue DEFAULT normal } OPTIONAL,
priority-ind	
[2] SEQUENCE {	SubjectIPMElementId,
subjectIPM	IPMIdentifier { OPTIONAL,
subject-ipm-id	
[3] SEQUENCE {	IPNOriginatorElementId,
IPNOriginator	ORDescriptor } OPTIONAL,
ipn-originating-user	
[4] SEQUENCE {	TimeOfReceiptElementId,
timeOfReceipt	DateandTime } OPTIONAL,
receipt-time	
[5] SEQUENCE {	ConvertedInfoTypesElementId,
convertedInfoTypes	SET OF
eIT	EITValue } OPTIONAL }

Note – If receipt-time element defined in Receipt is omitted, TLMAU extracts one from the CES of the session in which this TAPDU was transferred. This may differ from the time of actual receipt of IPM.

10.1.3.2 *ExplicitRNack-TAPDU*

The ExplicitRNack-TAPDU comprises the following elements:

ExplicitRNack-TAPDU	
ExplicitRN-TAPDU ::= SEQUENCE {	
[0] SEQUENCE {	
explicitRNack [0] ExplicitRNTAPDUId,	
[1] SEQUENCE {	
correlationInfo	CorrelationInfoElementId,
call-id	CallIdentification }},
[1] SET {	
[0] SEQUENCE {	
submissionId	SubmissionElementId,
submission-msg-id	MessageIdentifier }
[1] SEQUENCE {	
submissionTime	SubmissionTimeElementId,
submission-time	DateandTime }}}

10.1.4 *MessageCancel*

A TLM terminal sends a Cancel-TAPDU to invoke the MessageCancel abstract operation. The TLMAU returns no TAPDU to report the result of that operation, or may return an Exception-TAPDU (see § 10.1.1.3), to report an abstract error.

10.1.4.1 *Cancel-TAPDU*

The Cancel-TAPDU comprises the following elements:

Cancel-TAPDU	
Cancel-TAPDU ::= SEQUENCE {	
cancel [0] CancelTAPDUId,	
[1] SEQUENCE {	
submissionId	SubmissionIdElementId,
submission-msg-id	MessageIdentifier } OPTIONAL,
[2] SEQUENCE {	
correlation-Info	Correlation-InfoElementId,
call-id	CallIdentification } OPTIONAL }
<i>-- one of these must be present.</i>	

10.1.5 MessageDeliver

A TLMAU sends a Deliver-TAPDU to invoke the MessageDeliver abstract operation.

10.1.5.1 Deliver-TAPDU

The Deliver-TAPDU comprises the following elements:

Deliver-TAPDU	
<pre> Deliver-TAPDU ::= SEQUENCE { [0] SEQUENCE { deliver [0] DeliverTAPDUId, [1] SEQUENCE { quantityOfDocs number-of-docs } } -- MTS parameters [1] SET { [0] SEQUENCE { priority priority-ind } [1] SEQUENCE { originator originator-name } [2] SEQUENCE { thisRecipient this-recipient-name } [3] SEQUENCE { orgIntendedRecipient org-intended-recipient-name } [4] SEQUENCE { otherRecipients SET OF otherRecipient-name } [5] SEQUENCE { redirectedfrom SEQUENCE OF redirected-from } [6] SEQUENCE { submissionTime submission-time } [7] SEQUENCE { deliveryId delivery-msg-id } [8] SEQUENCE { conversionIndication SET{ [0] SET OF eIT } conversion-prohibited [1] ConversionProhibitedValue OPTIONAL } } OPTIONAL, [9] SEQUENCE { convertedInfoTypes SET OF eIT } } </pre>	<pre> QuantityOfDocsElementId, NumberOfAssociatedDocuments } OPTIONAL }, PriorityElementId, PriorityValue DEFAULT normal } OPTIONAL }, OriginatorElementId, ORName } OPTIONAL, ThisRecipientElementId, ORName }, OrgIntendedRecipientElementId, ORName } OPTIONAL, OtherRecipientsElementId, ORName } OPTIONAL, RedirectedFromElementId, ORName } OPTIONAL, SubmissionTimeElementId, DateandTime }, DeliveryElementId, MessageIdentifier } OPTIONAL, ConversionIndicationElementId, EITValue OPTIONAL } } ConvertedInfoTypesElementId, EITValue } }, </pre>

Deliver-TAPDU (continued)

-- Deliver-TAPDU Definition (continued)

-- IPMS parameters

```
[2] SET {
    [0] SEQUENCE {
        thisIPM                                ThisIPMElementId,
        this-ipm-id                            IPMIdentifier },
    [1] SEQUENCE {
        from                                    FromElementId,
        originating-user                       OrDescriptor } OPTIONAL,
    [2] SEQUENCE {
        authorizing                            AuthorizingElementId,
        SET OF
        authorizing-user                       ORDescriptor } OPTIONAL,
    [3] SEQUENCE {
        to                                      ToElementId,
        SET OF SEQUENCE {
        primary-recipient [0] ORDescriptor,
        [1] NotificationSpecification }} OPTIONAL,
    [4] SEQUENCE {
        cc                                      CCElementId,
        SET OF SEQUENCE {
        copy-recipient [0] ORDescriptor,
        [1] NotificationSpecification }} OPTIONAL,
    [5] SEQUENCE {
        bcc                                      BCCElementId,
        SET OF SEQUENCE {
        blind-copy-recipient [0] ORDescriptor,
        [1] NotificationSpecification }} OPTIONAL,
    [6] SEQUENCE {
        repliedToIPM                            RepliedToIPMElementId,
        replied-to-ipm-id                       IPMIdentifier } OPTIONAL,
    [7] SEQUENCE {
        obsoletedIPMs                           ObsoletedIPMsElementId,
        SET OF
        obsoleted-ipm-id                       IPMIdentifier } OPTIONAL,
    [8] SEQUENCE {
        relatedIPMs                             RelatedIPMsElementId,
        SET OF
        related-ipm-id                         IPMIdentifier } OPTIONAL,
    [9] SEQUENCE {
        subject                                  SubjectElementId,
        subject-content                         SubjectContent } OPTIONAL,
    [10] SEQUENCE {
        contentIndicator                        ContentIndicatorElementId,
        SEQUENCE{
        expiry-time [0] DateandTime OPTIONAL,
        [1] SET {
        importance [0] ImportanceValue DEFAULT normal,
        sensitivity [1] SensitivityValue OPTIONAL,
        auto-forwarded [2] AutoForwardedValue DEFAULT
        notAutoForward }}} OPTIONAL,
```

Deliver-TAPDU (*end*)

-- *Deliver-TAPDU Definition (continued)*

```

    [11] SEQUENCE {
        reply                               ReplyElementId,
            SEQUENCE {
                reply-time [0] DateandTime,
                [1] SET OF
                reply-recipient             ORDescriptor } } OPTIONAL,
    [12] SEQUENCE {
        language                             LanguageElementId,
        language-ind                         LanguageInd } OPTIONAL,
    MsgIncomplete [13] MsgIncompleteElementId OPTIONAL },
    -- Body
    [3] SEQUENCE {
        [0] SEQUENCE {
            bodyType                         BodyTypeElementId,
            body-part                         BodyPartValue } OPTIONAL,
        [1] SEQUENCE {
            forwardedInfo                     ForwardedInfoElementId,
                SEQUENCE {
                    forwarded-time [0] DateandTime,
                    [1] DeliveryEnvelope } } OPTIONAL,
    -- Delivery Envelope contains same set of MTS parameters of Deliver-TAPDU } }

```

-- *Definition of Notification Specification*

```

Notification Specification ::= SET {
    rn-request          [0] RNRequestValue OPTIONAL,
    nrn-request        [1] NRNRequestValue OPTIONAL,
    return-request     [2] ReturnRequestValue OPTIONAL,
    reply-request      [3] ReplyRequestValue DEFAULT
                    noReply }

```

10.1.6 *ReceiptStatusNotice*

A TLMAU terminal sends a ReceiptStatusNotice-TAPDU to invoke the ReceiptStatusNotice abstract operation.

10.1.6.1 *ReceiptStatusNotice-TAPDU*

The ReceiptStatusNotice-TAPDU comprises the following elements:

ReceiptStatusNotice-TAPDU

```

ReceiptStatusNotice-TAPDU ::= SEQUENCE {
  [0] SEQUENCE {
    receiptStatusNotice [0] ReceiptStatusNoticeTAPDUId,
      [1] SEQUENCE {
        quantityOfDocs          QuantityOfDocsElementId,
        number-of-docs          NumberOfAssociatedDocuments } OPTIONAL },
    -- MTS parameters
  [1] SET {
    [0] SEQUENCE {
      priority          PriorityElementId,
      priority-ind      PriorityValue }
    [1] SEQUENCE {
      deliveryId        DeliveryIdElementId,
      delivery-id       MessageIdentifier } OPTIONAL,
    [2] SEQUENCE {
      originator        OriginatorElementId,
      originator-name   ORName } OPTIONAL,
    [3] SEQUENCE {
      thisRecipient     ThisRecipientElementId,
      this-recipient-name ORName },
    [4] SEQUENCE {
      submissionTime    SubmissionTimeElementId,
      submissionTime    DateandTime },
    [5] SEQUENCE {
      timeOfDelivery    TimeOfDeliveryElementId,
      delivery-time      DateandTime },
    [6] SEQUENCE {
      conversionIndication ConversionIndicationElementId,
      SET {
        [0] SET OF
          eIT          EITValue } OPTIONAL }}
    [7] SEQUENCE {
      convertedInfotypes  ConvertedInfoTypesElementId,
      SET OF
        eIt          EITValue }},
    --IPMS parameters
  [2] SET {
    [0] SEQUENCE {
      notificationType  NotificationTypeElementId,
      report-type       ReportTypeValue },
    [1] SEQUENCE {
      subjectIPM        SubjectIPMElementId,
      subject-ipm-id    IPMIdentifier },
    [2] SEQUENCE {
      IPNOriginator     IPNOriginatorElementId,
      ipn-originating-user ORDescriptor } OPTIONAL,
    [3] SEQUENCE {
      preferredRecipient PreferredRecipientElementId,
      preferred-recipient ORDescriptor } OPTIONAL,
  }
}

```

ReceiptStatusNotice-TAPDU (*end*)

-- *ReceiptStatusNotice-TAPDU Definition (continued)*

```

[4] SET {
    [0] SEQUENCE {
        timeOfReceipt           TimeOfReceiptElementId,
        receipt-time           DateandTime },
    [1] SEQUENCE {
        typeOfReceipt           TypeOfReceiptElementId,
        type-of-receipt         TypeOfReceiptValue DEFAULT manual } OPTIONAL,
    [2] SEQUENCE {
        supplReceiptInfo        SupplReceiptInfoElementId,
        suppl-receipt-info      SupplementaryInformation } OPTIONAL } OPTIONAL,
[5] SET {
    [0] SEQUENCE {
        nonReceiptInfo          NonReceiptInfoElementId,
        SET {
            non-receipt-reason [0] NonReceiptReasonValue,
            discard-reason    [1] DiscardReasonValue OPTIONAL }} OPTIONAL,
    [1] SEQUENCE {
        comments                CommentElementId,
        comments                Comment },
messageReturnedInd [2] MessageReturnedIndElementId OPTIONAL }}}

```

10.1.7 *DeliveryStatusNotice*

A TLMAU terminal sends a *DeliveryStatusNotice-TAPDU* to invoke the *DeliveryStatusNotice* abstract operation.

10.1.7.1 *DeliveryStatusNotice-TAPDU*

The *DeliveryStatusNotice-TAPDU* comprises the following elements:

DeliveryStatusNotice-TAPDU

```

DeliveryStatusNotice-TAPDU ::= SEQUENCE {
  [0] SEQUENCE {
    deliveryStatusNotice [0] DeliveryStatusNoticeTAPDUId,
      [1] SEQUENCE {
        quantityOfDocs      QuantityOfDocsElementId,
        number-of-docs      NumberOfAssociatedDocuments } OPTIONAL },
      [2] SEQUENCE {
        correlationInfo      CorrelationInfoElementId,
        call-id              CallIdentification }},
  [1] SET {
    [0] SEQUENCE {
      submissionId          SubmissionIdElementId,
      submission-msg-id     MessageIdentifier } OPTIONAL,
    [1] SEQUENCE {
      probeId              ProbeIdElementId,
      submission-msg-id     MessageIdentifier } OPTIONAL,
    [2] SET OF {
      [0] SEQUENCE {
        reportedRecipient    ReportedRecipientElementId,
        reported-recipient-name ORName },
      [1] SEQUENCE {
        notificationType     NotificationTypeElementId,
        report-type          ReportTypeValue },
      [2] SEQUENCE {
        intendedRecipient     IntendedRecipientElementId,
        intended-recipient-name ORName },
      [3] SEQUENCE {
        convertedInfoTypes    ConvertedInfoTypesElementId,
        SET OF
        eIT                  EITValue },
      [4] SET {
        -- In case of Delivery Notification,
        -- this set of element shall be present.
        [0] SEQUENCE {
          timeOfDelivery      TimeOfDeliveryElementId,
          delivery-time       DateandTime },
        [1] SEQUENCE {
          typeOfUA            TypeOfUAElementId,
          type-of-ua          TypeOfUA DEFAULT public } OPTIONAL,
        [2] SEQUENCE {
          supplInfo           SupplInfoElementId,
          suppl-info          SupplementaryInformation } OPTIONAL,
          [5] SET {
            -- In case of Non Delivery Notification,
            -- this set of element shall be present.
            [0] SEQUENCE {
              nonDeliveryReason NonDeliveryReasonElementId,
              SET {
                reason-code    [0] ReasonCodeValue,
                diagnostic-code [1] DiagnosticCodeValue OPTIONAL }}} OPTIONAL },
        contentReturned [3] ContentReturnedElementId OPTIONAL }}

```

10.1.8 Register

A TLM terminal sends a Register-TAPDU to invoke the register abstract operation. The TLMAU returns a RegisterAck-TAPDU, if necessary, to report the result of that operation, or may return an Exception-TAPDU (see § 10.1.1.3) to report an abstract error.

10.1.8.1 Register-TAPDU

The Register-TAPDU comprises the following elements:

Register-TAPDU	
Register-TAPDU ::= SEQUENCE {	
[0]	register TAPDUIdValue,
[1] SET {	
[0] SET {	
[0] SEQUENCE {	
expiredDiscard	ExpiredDiscardElementId,
discard-ipm	DiscardValue DEFAULT discard } OPTIONAL,
[1] SEQUENCE {	
obsoleteDiscard	ObsoleteDiscardElementId,
discard-ipm	DiscardValue DEFAULT discard } OPTIONAL },
[1] SET {	
[0] SEQUENCE {	
autoFWDIPMs	AutoFWDIPMsElementId,
auto-fwd-ipms	AutoFWDIPMsValue DEFAULT not-auto-forward } OPTIONAL,
[1] SEQUENCE {	
autoFWDRecipients	AutoFWDRecipientsElementId,
SET OF {	
auto-fwd-recipient-name	ORName } } OPTIONAL,
[2] SEQUENCE {	
autoFWDHeading	AutoFWDHeadingElementId,
auto-fwd-heading	AutoFWDHeading } OPTIONAL,
<i>-- For further study</i>	
[3] SEQUENCE {	
autoFWDComment	AutoFWDCommentElementId,
auto-fwd-comment	AutoFWDComment } OPTIONAL },
[2] SET {	
[0] SEQUENCE {	
dsMode	DSModeElementId,
ds-mode	DSModeValue } OPTIONAL,
[1] SEQUENCE {	
tLMAUOperation	TLMAUOperationElementId,
SET {	
error-recovery-mode	[0] ErrorRecoveryModeValue OPTIONAL,
auto-acknowledgment	[1] AutoAcknowledgment DEFAULT manual } } OPTIONAL,
[2] SEQUENCE {	
supplRecipientInfo	SupplRecipientInfoElementId,
suppl-recipient-info	SupplementaryInformation } OPTIONAL,
[3] SEQUENCE {	
autoOutput	AutoOutputElementId,
SET {	
frequency	[0] Frequency OPTIONAL,
output-time	[1] DateandTime OPTIONAL } } OPTIONAL,
[4] SEQUENCE {	
messageDeleteMode	MessageDeleteModeElementId,
message-delete-mode	MessageDeleteModeValue DEFAULT auto-delete } OPTIONAL } }

10.1.8.2 RegisterAck-TAPDU

The RegisterAck-TAPDU comprises the following elements:

RegisterAck-TAPDU	
RegisterAck-TAPDU ::=	RegisterAckTAPDUId
registerAck	

10.1.9 DSList

A TLM terminal sends a DSQuery-TAPDU to invoke the DSList abstract operation. The TLMAU returns a DSReport-TAPDU to report the result of that operation, or may return an Exception-TAPDU (see § 10.1.1.3) to report an abstract error.

10.1.9.1 DSQuery-TAPDU

The DSQuery-TAPDU comprises the following elements:

DSQuery-TAPDU	
DSQuery-TAPDU ::=	DSQueryTAPDUId
dsQuery	

10.1.9.2 DSReport-TAPDU

The DSReport-TAPDU comprises the following elements:

DSReport-TAPDU	
DSReport-TAPDU ::= SEQUENCE {	
[0]	DSReportTAPDUId,
[1] SET OF {	
[0] SEQUENCE {	RetrievalIdElementId,
retrievalId	RetrievalIdentifier },
[1] SEQUENCE {	MessageTypeElementId,
messageType	MessageTypeValue },
[2] SEQUENCE {	OriginatorElementId,
originator	ORName } OPTIONAL,
[3] SEQUENCE {	PriorityElementId,
priority	PriorityValue DEFAULT normal } OPTIONAL,
[4] SEQUENCE {	MessageLengthElementId,
messageLength	MessageLength } OPTIONAL}}

10.1.10 *DSDelete*

A TLM terminal sends a MessageDelete-TAPDU to invoke the DSDelete abstract operation. The TLMAU returns no TAPDU to report the result of that operation, or may return an Exception-TAPDU (see § 10.1.1.3) to report an abstract error.

10.1.10.1 *MessageDelete-TAPDU*

The MessageDelete-TAPDU comprises the following elements:

MessageDelete-TAPDU	
<pre> MessageDelete-TAPDU ::= SEQUENCE { messageDelete [0] MessageDeleteTAPDUId, [1] SEQUENCE { SET OF { messageSelector MessageSelectorElementId, retrieval-id RetrievalIdentifier }}} </pre>	

10.1.11 *DSFetch*

A TLM terminal sends an OutputRequest-TAPDU to invoke the DSFetch abstract operation. The TLMAU returns an OutputMessage-TAPDU to report the result of that operation, or may return an Exception-TAPDU (see § 10.1.1.3) to report an abstract error.

The OutputMessage-TAPDU is sent by TLMAU to be output the message from DS. This TAPDU is triggered by one of the following events:

- 1) some rule (not defined in this Recommendation) which causes TLMAU to establish a connection to the TLM terminal and to send a message at a specific time, for example, the TLM terminal has registered its times of availability with TLMAU;
- 2) the TLM terminal establishes a connection to TLMAU and initiates a CSCC which is taken as an implicit request for output by TLMAU;
- 3) receipt of an OutputRequest-TAPDU.

10.1.11.1 *OutputRequest-TAPDU*

The OutputRequest-TAPDU comprises the following elements:

OutputRequest-TAPDU	
<pre> OutputRequest-TAPDU ::= SEQUENCE { [0] outputRequest OutputRequestTAPDUId, [1] SET OF SEQUENCE { [0] SEQUENCE { retrievalId RetrievalIdElementId, retrieval-id RetrievalIdentifier }, [1] SEQUENCE { deleteAfterOutput DeleteAfterOutputElementId, delete-after-output DeleteAfterOutputValue } OPTIONAL }} </pre>	

10.1.11.2 OutputMessage-TAPDU

The OutputMessage-TAPDU comprises the following elements:

OutputMessage-TAPDU	
<pre> OutputMessage-TAPDU ::= SEQUENCE { [0] SEQUENCE { outputMessage [0] OutputMessageTAPDUId, [1] SEQUENCE { quantityOfDocs QuantityOfDocsElementId, number-of-docs NumberOfAssociatedDocuments } OPTIONAL }, [1] SET OF SEQUENCE { [0] SEQUENCE { retrievalId RetrievalIdElementId, retrieval-id RetrievalIdentifier } [1] SEQUENCE { messageType MessageTypeElementId, message-type MessageTypeValue } [2] SEQUENCE { timeOfDelivery TimeOfDeliveryElementId, delivery-time DateandTime } -- The remaining Components of this TAPDU are identical to the components in the Deliver, DeliverStatusNotice and ReceiptStatusNotice-TAPDU. The actual components to be used depend upon the MessageType parameter value specified in the MessageType component. }} </pre>	

Note – The RetrievalIdentifier is an identifier which identifies a message in DS.

10.1.12 MessageState

A TLM terminal sends a StatusQuery-TAPDU to invoke the MessageState abstract operation. The TLMAU returns a StatusReport-TAPDU to report the result of that operation, or returns an Exception-TAPDU to report an abstract error.

10.1.12.1 StatusQuery-TAPDU

The StatusQuery-TAPDU comprises the following elements:

StatusQuery-TAPDU	
<pre> StatusQuery-TAPDU ::= SEQUENCE { [0] statusQuery StatusQueryTAPDUId, [1] SET { [0] SEQUENCE { submissionId SubmissionIdElementId, submission-msg-id MessageIdentifier } OPTIONAL, -- See Note [1] SEQUENCE { correlationInfo CorrelationInfoElementId, call-id CallIdentification } OPTIONAL }} -- See Note </pre>	

Note – If none of these are present all outstanding (in operation) operations will be reported.

10.1.12.2 *StatusReport-TAPDU*

The StatusReport-TAPDU comprises the following elements:

StatusReport-TAPDU	
<pre> StatusReport-TAPDU ::= SEQUENCE { [0] SEQUENCE { statusReport [0] StatusReportTAPDUId [1] SEQUENCE { correlationInfo CorrelationInfoElementId, call-id CallIdentification }}, [1] SET { [0] SEQUENCE { timeOfReport TimeOfReportElementId, report-time DateandTime }, [1] SEQUENCE { reportedMessageId ReportedMessageIdElementId, reported-message-id MessageTypeValue } [2] SET OF SEQUENCE { [0] SEQUENCE { actualRecipient ActualRecipientElementId, actual-recipient-name ORName }, [1] SEQUENCE { messageStatus [1] MessageStatusElementId, status StatusValue }, [2] SET { timeOfDelivery [0] SEQUENCE { TimeOfDeliveryElementId, DateandTime }, typeOfUA [1] SEQUENCE { TypeOfUAElementId, TypeOfUA DEFAULT public } OPTIONAL } OPTIONAL }, [3] SEQUENCE { nonDeliveryReason NonDeliveryReasonElementId, SET { reason-code [0] ReasonCodeValue, diagnostic-code [1] DiagnosticCodeValue } OPTIONAL }}, } </pre>	
	<p><i>-- In case of DN, this set -- of element shall be present.</i></p>
	<p><i>-- In case of DN, this set -- of element shall be present.</i></p>

10.2 Operation of the TLMAU

The section describes how the TLMAU will provide the TLM abstract service. The TLM abstract operations have been defined by abstract operations, sometimes with associated results or errors. These abstract operations, results and errors are realized via the exchange of TAPDUs between the TLM terminal and the TLMAU.

The realization of the abstract operations for the import and export ports linking the TLMAU and the MTS, is beyond the scope of this Recommendation. For the purpose of this section import and export operations will be considered to be similar to submission and delivery port operations.

10.2.1 MessageSend

The MessageSend operation will be provided by the TLMAU via the Send-, SendAck- and Exception-TAPDUs:

Upon receipt of Send-TAPDU by the TLMAU, the TLMAU will take the following actions:

- 1) The TLMAU will invoke the MTAS import abstract operation MessageSubmission with the following argument values:

Source of MTAS MessageSubmission arguments

MessageSubmission argument	CorrespondingSend-TAPDU component		Operation
	Element name	Value name	
originator-name	–	–	Authenticate User (Note 1)
original-encoded-information-types	–	–	set by TLMAU to EITs of submitted IPM's body
content-type	–	–	set by TLMAU to "IPMS"
content-identifier	–	–	Local matter
content-correlator	–	–	Local matter
recipient-name	to, cc, bcc	primary-, copy-, blind-copy-recipient	constructed by TLMAU

Note 1 – Authenticate User is constructed from TID obtained from CSS terminal ID.

Note 2 – The IPM submitted as the content is constructed by the TLMAU. Send-TAPDU components representing IPMS elements are mapped onto the corresponding IPMS application protocol data unit (APDU) elements.

Note 3 – A multi-document messages will be submitted as an IP message with a multi-part body, each body part corresponding to a submitted document.

Note 4 – When this-IPM of IPMS element is omitted, the TLMAU shall construct this component which consists of the following components: originator name, Date and Time and, if necessary, a sequence number.

Other message submission arguments have a corresponding Send-TAPDU component. If this component is omitted, the default value applies.

- 2) If the MessageSubmission operation results in an error or if an error is detected in the Send-TAPDU, the TLMAU will return an Exception-TAPDU to the originating TLM terminal.
- 3) The TLMAU will, when required, return a SendAck-TAPDU to the originating TLM terminal following the successful completion of the MessageSubmission operation. The values of the SendAck-TAPDU will be set as follows:

Source of SendAck-TAPDU components

SendAck-TAPDU component		Source
Element name	Value name	
correlationInfo	call-id	CallIdentification that identifies previous Send-TAPDU being reported on
submissionId	submission-msg-id	MTS message-submission-identifier
submissionTime	submission-time	MTS message-submission-time

- 4) The TLMAU will maintain a one-to-one correlation between MTS message-submission-identifiers and correlation information values to facilitate status query.

10.2.2 *MessageProbe*

The MessageProbe operation is provided by the TLMAU via the Probe-, ProbeAck- and Exception-TAPDUs.

Upon receipt of the Probe-TAPDU by the TLMAU, the TLMAU will take the following actions:

- 1) The TLMAU will invoke the MTAS import abstract operation ProbeSubmission with the following argument values:
 Source of MTAS ProbeSubmission arguments.
 See § 10.2.1 - MessageSubmission arguments.
- 2) If the Probe operation results in an error or if an error is detected in the Probe-TAPDU, the TLMAN will return an Exception-TAPDU to the originator.
- 3) The TLMAU will, when required, return a ProbeAck-TAPDU to the originator, following the successful completion of the probe operation. The values of the ProbeAck-TAPDU will be set as follows:

Source of ProbeAck-TAPDU components

ProbeAck-TAPDU component		Source
Element name	Value name	
correlationInfo	call-id	CallIdentification that identifies previous Probe-TAPDU being reported on
probeId	probe-msg-id	MTS probe-submission-identifier
submissionTime	submission-time	MTS probe-submission-time

10.2.3 *ExplicitReceive*

The *ExplicitReceive* operation is provided by the TLMAU via the *ExplicitRN-*, *ExplicitRNack-* and *Exception-TAPDU*s.

Upon receipt of the *ExplicitRN-TAPDU* the TLMAU will take the following actions:

- 1) The TLMAU will invoke the MTAS import abstract operation *MessageSubmission* with the following argument values:

Source of MTAS *MessageSubmission* components

MessageSubmission argument	Corresponding ExplicitRN-TAPDU component		Operation
	Element name	Value name	
originator-name	–	–	authenticate user
original-encoded-information-types	–	–	set by TLMAU to “unspecified”
content-type	–	–	set by TLMAU to “IPMS”
content-identifier	–	–	Local matter
priority	priority	priority-ind	
per-message-indicators disclose-recipients conversion-prohibited alternate-recipient-allowed content-return-request	– – – –	– – – –	set to “disclosure-of recipient-prohibited” set to “conversion-prohibited” set to “alternate-recipient-prohibited” set to “content-return-not-requested”
recipient-name	recipients	recipient-name	
originator-report-request	–	–	set by TLMAU to “no report”
content	–	–	identified as IPN

Note 1 – The IPN submitted as the content is constructed by the TLMAU. *ExplicitRN-TAPDU* elements representing IPMS elements are mapped onto the corresponding IPMS APDU elements.

Note 2 – If receipt-time is omitted, the TLMAU extracts the Receipt time from the CSS of the session in which this TAPDU was transferred to. This time may differ from the time of actual receipt of IPM.

Note 3 – Set acknowledgment-mode of IPN to "manual".

- 2) If the Message-Submission operation results in an error, or if an error is detected in the ExplicitRN-TAPDU, the TLMAU will return an Exception-TAPDU to the originator.
- 3) The TLMAU will, when required, return a ExplicitRNack-TAPDU to the originator, following the successful completion of the MessageSubmission operation. The values of the ExplicitRNack-TAPDU will be set as follows:

Source of ExplicitRNack-TAPDU components

ExplicitRNack-TAPDU component		Source
Element name	Value name	
correlationInfo	call-id	CallIdentification that identifies previous ExplicitRN-TAPDU being reported on
submissionId	submission-msg-id	MTS message-submission-identifier
submissionTime	submission-time	MTS message-submission-time

10.2.4 *MessageCancel*

The MessageCancel operation is provided by the TLMAU via the Cancel- and Exception-TAPDUs.

Upon receipt of the Cancel-TAPDU by the TLMAU, the TLMAU will take the following actions:

The TLMAU will invoke the MTAS abstract operation CancelDeferredDelivery with the following argument value:

Source of CancelDeferredDelivery arguments

CancelDeferredDelivery argument	Corresponding Cancel-TAPDU component		Operation
	Element name	Value name	
message-submission-identifier	submissionId	submission-msg-id	

If the CancelDeferredDelivery operation results in an error, or if an error is detected in the Cancel-TAPDU, the TLMAU will return an Exception-TAPDU to the originating TLM terminal.

10.2.5 MessageDeliver

The MessageDeliver operation is provided by the TLMAU via the Deliver-TAPDU.

When the MTAS abstract operation MessageDelivery is invoked by the MTS with an IPM as the MTS message content, the TLMAU will take the following actions:

- 1) The TLMAU will construct a Deliver-TAPDU for transmission to the destination TLM terminal with the following element values:

Source of Deliver-TAPDU component

Deliver-TAPDU component		Corresponding MessageDelivery argument	Operation
Element name	Value name		
quantityOfDocs	number-of-docs	—	when control Information is conveyed by a normal document, set number of associated documents in Deliver-TAPDU
priority	priority-ind	priority	
originator	originator-name	originator-name	
thisRecipient	this-recipient-name	this-recipient-name	
intendedRecipient	intended-recipient-name	originally-intended-recipient-name	
otherRecipients	other-recipient-name	other-recipient-names	
submissionTime	submission-time	message-submission-time	
timeOfDelivery	delivery-time	message-delivery-time	
deliveryId	delivery-msg-id	message-delivery-identifier	
conversionIndication	eIT	original-encoded-information-types	
conversionIndication	conversion-prohibited	delivery-flags	
convertedInfoTypes	eIT	converted-encoded-information-types	

Note 1 – The IPM received by TLMAU is used to construct the Deliver-TAPDU, Deliver-TAPDU elements, representing MTS and IPMS elements of service, are constructed by the TLMAU from the MessageDeliver operation arguments and IPMS application protocol data unit (APDU) values as indicated above.

Note 2 – Multi-part body message will be sent to the destination TLM terminal by the TLMAU as a multi-document message, each document corresponding to an IP message body part.

- 2) If the TLMAU is unable to deliver the constructed Deliver-TAPDU to the designation TLM terminal, then an IPN will be constructed for return to the IPMS originator. This IPN will be submitted according to § 10.2.6.
- 3) The definition of the export port MessageDeliver abstract operation should include a result argument indicating successful delivery or non-delivery. The MTS would then return delivery notifications to the originators of messages routed through a TLMAU only after the result value was indicated.

10.2.6 ReceiptStatusNotice

The ReceiptStatusNotice operation is provided by the TLMAU via the ReceiptStatusNotice-TAPDU.

When the MTAS abstract operation MessageDelivery is invoked by the MTS with an IPN as the IPMS content, the TLMAU will take the following actions:

- 1) The TLMAU will construct a ReceiptStatusNotice-TAPDU for transmission to the destination TLM terminal with the following element values:

Source of ReceiptStatusNotice-TAPDU components

ReceiptStatusNotice-TAPDU component		Corresponding MessageDelivery and receiveRN/NRN argument	Operation
Element name	Value name		
quantityOfDocs	number-of-docs	—	when control Information is conveyed by a normal document, set number of associated documents in ReceiptStatusNotice-TAPDU
priority	priority-ind	priority	
deliveryId	delivery-id	message-delivery-identifier	
originator	originator-name	originator-identifier	if this element is omitted, this argument should be constructed from TID obtains from CSS
thisRecipient	this-recipient-name	this-recipient-identifier	
submissionTime	submission-time	message-submission-time	
timeOfDelivery	delivery-time	message-delivery-time	
conversionIndication	eIT	original-encoded-information-types	
conversionIndication	conversion-prohibited	delivery-flags	
convertedInfoTypes	eIT	converted-encoded-information-types	
notificationType	report-type	choice	
subjectIPM	subject-imps-id	subject-ipm	
iPNOriginator	ipn-originating-user	ipn-originator	
preferredRecipient	preferred-recipient	ipm-preferred-recipient	
timeOfReceipt	receipt-time	receipt-time	
typeOfReceipt	type-of-receipt	acknowledgment-mode	
supplReceiptInfo	suppl-receipt-info	suppl-receipt-info	
nonReceiptInfo	non-receipt-reason	non-receipt-reason	
nonReceiptInfo	discard-reason	discard-reason	
comment	comments	auto-forward-comment	
messageReturnedInd	—	returned-ipm	

Note – What is received by the TLMAU is used to construct the ReceiptStatusNotice-TAPDU. ReceiptStatusNotice-TAPDU elements representing MTS and IPMS elements of service, are constructed by the TLMAU from the MessageDeliver operation arguments and IPMS APDU values, as indicated above.

10.2.7 *DeliveryStatusNotice*

The DeliveryStatusNotice operations are provided by the TLMAU via the DeliveryStatusNotice-TAPDU.

When the MTS abstract operation ReportDelivery is invoked by the MTS, the TLMAU will take the following actions:

- 1) The TLMAU will construct a DeliveryStatusNotice-TAPDU for transmission to the destination TLM terminal with the following element values:

Source of DeliveryStatusNotice-TAPDU components

DeliveryStatusNotice-TAPDU component		Corresponding ReportDelivery argument	Operation
Element name	Value name		
quantityOfDocs	number-of-docs	–	when control Information is conveyed by a normal document, set number of associated documents in DeliverStatusNotice-TAPDU
correlationInfo	call-id	–	CallIdentification that identifies previous Send-TAPDU being reported on
priority	priority-ind	priority	
SubmissionId	submission-id	subject-identifier	
probeId	submission-id	subject-identifier	
reportedRecipient	reported-recipient-name	actual-recipient-identifier	
notificationType	report-type	report	
intendedRecipient	intended-recipient-name	originally-intended-recipient	
convertedInfoType	eIT	converted-encoded-information-types	
timeOfDelivery	delivery-time	message-delivery-time	
typeOfUA	type-of-ua	type-of-MTS-user	
nonDeliveryReason	reason-code	non-delivery-reason-code	
nonDeliveryReason	diagnostic-code	non-delivery-diagnostic-code	
contentReturned		returned-content	

- 2) When required, the TLMAU will accumulate notifications pertaining to a single Send-TAPDU and construct a single DeliveryStatusNotice-TAPDU from multiple ReportDelivery operations.

10.2.8 *Register*

The register operation is provided by the TLMAU via the register-, RegisterAck- and Exception-TAPDUs.

Upon receipt of the Register-TAPDU, the TLMAU will take the following actions:

- If a message delete mode was selected. the TLMAU will subsequently operate according to the new mode with respect to messages output from the DS of the TLM terminal originating the Register-TAPDU,

- If an error recovery mode was selected, the TLMAU will subsequently handle error recovery according to the selected criteria for all transactions with the originator of the Register-TAPDU.
- If a DS mode was selected, the TLMAU will subsequently either hold for retrieval, or auto output messages in the DS of the originator of the Register-TAPDU according to the DS mode selected in this TAPDU.
- If the auto discard mode was enabled by the Register-TAPDU, then the TLMAU will commence automatic deletion of messages in the DS belonging to the originator of the Register-TAPDU when they are obsoleted by subsequent received IPM's.
- If the auto acknowledgement function was enabled by the Register-TAPDU, then the TLMAU will automatically format and submit receipt notifications for subsequent IP messages directed to the originator of the Register-TAPDU. These notifications will be submitted, either following successful delivery of the IP message to the TLM terminal, or upon deposit of the IP message in the TLM terminal's DS.
- If an error is detected with the Register-TAPDU, the TLMAU will return an Exception-TAPDU to the originator.

10.2.9 *DSList*

The DSList operation is implemented by the TLMAU as an internal operation and does not involve the MTS. The DS list operation is provided via the DSQuery-, DSReport- and Exception-TAPDUs as follows:

Upon receipt of the DSQuery-TAPDU by the TLMAU, the TLMAU will take the following actions:

- The TLMAU will prepare a DSReport-TAPDU for return to the originator. If there are no messages in DS, the DSReport-TAPDU will indicate this.
- If an error is detected with the DSQuery-TAPDU, the TLMAU will return an Exception-TAPDU to the originator.

10.2.10 *DSDelete*

The DSDelete operation is implemented by the TLMAU as an internal operation and does not involve the MTS. The DS Delete operation is provided via the DSDelete- and Exception-TAPDUs as follows:

- The TLMAU will delete the indicated message(s) from the DS.
- If an error is detected with the DSDelete-TAPDU or the message indicated is not available for deletion, the TLMAU will return an Exception-TAPDU to the originator.

10.2.11 *DSFetch*

The DSFetch operation is implemented by the TLMAU as an internal operation and does not involve the MTS. The DSFetch operation is provided via the OutputRequest-, OutputMessage- and Exception-TAPDUs as follows:

Upon receipt of the OutputRequest-TAPDU by the TLMAU, the TLMAU will take the following actions:

- For each message indicated in the OutputRequest-TAPDU and found in the DS, the TLMAU will prepare and return an OutputMessage-TAPDU.
- If the delete-after-output function was indicated in the OutputMessage-TAPDU the TLMAU will delete the indicated message(s) from the DS after output.
- If the "auto delete" message delete mode is subscribed to then the TLMAU will delete the indicated message(s) from the DS after output regardless of whether the delete-after-output function was selected in the OutputRequest-TAPDU.
- If an error is detected with the OutputRequest-TAPDU or the message(s) indicated were not available for output, the TLMAU will return an Exception-TAPDU to the originator. If some of a list of indicated messages are available in DS then the TLMAU will output those available, and then return an Exception-TAPDU for those not available or incorrectly indicated.
- If the "auto output" DS mode is subscribed to then the output and associated deletion functions will be executed when the user subscribed conditions are met.

10.2.12 *OutputMessage*

The delivery-time in this TAPDU is the time when the DS received the message.

10.2.13 *MessageStatus*

The MessageStatus operation is implemented by the TLMAU as an internal operation and does not involve the MTS. This operation is applicable only when the TLMAU accumulates notifications for previously submitted multi-address messages. The operations provided by the TLMAU via the StatusQuery-, StatusReport-and Exception-TAPDUs.

Upon receipt of the StatusQuery-TAPDU by the TLMAU, the TLMAU will take the following actions:

- The TLMAU will construct a StatusReport-TAPDU from accumulated notifications pertaining to the message identified in the StatusRequest-TAPDU.
- The TLMAU will not allow StatusQuery operation for ReceiptStatusNotice.
- If an error is detected with the StatusReport-TAPDU or there is no record of the message indicated, the TLMAU will return an Exception-TAPDU to the originator.

11 Formats and encoding of TAPDU

11.1 *Principles*

Elements of a telematic access protocol data unit (TAPDU) shall be coded using human-readable graphic characters of Recommendation T.61 coding scheme. Other coding rules such as machine-readable coding are for further study.

11.2 *Structure of TAPDU*

- 1) A TAPDU is composed of one or more documents. The first one contains control information optionally followed by one or more documents with text (message body information). Within one session one or more TAPDU may be conveyed.
- 2) Control information is conveyed in either a control document or a normal document.
- 3) The control information is subdivided into a TAPDUs and elements each containing a number field and/or name field, and optionally one or more element value fields. An element number field, which is language independent, and the element name, which is language dependent, uniquely identify an element. In case of international access, the element number field must always be present.
- 4) The value fields of an element may contain the same TAPDU information types or different TAPDU information types. The element value Fields (called components) are categorized as follows:
 - components with pre-defined values, i.e. components with a specific, enumerable set of known, unique values (predefined value);
 - components with a wide range of values which are not pre-defined (general value).
- 5) There are two different types of component fields:
 - primitive component;
 - constructor component.
- 6) Each primitive component contains only one parameter. Each constructor component contains more than one parameter.
- 7) A parameter contains a parameter value, optionally preceded by a Parameter-Id. which identifies the parameters.
- 8) The formal description of the structure of a TAPDU is shown in Table 3/T.330.
- 9) A line may contain an Element-Id field and component fields, or the first component field of the element starts on a new line.
- 10) If the number of characters of the component exceeds the remaining number of characters on this line, the component must be divided into two or more lines by "NL" function. However, it is not allowed to divide the component within a parameter.

TABLE 3/T.330

The structure of TAPDUs

```

TAPDU ::= SEQUENCE {
    ControlInfo,
    SEQUENCE OF MessageBodyInfo OPTIONAL },
ControlInfo ::= SEQUENCE {
    TAPDUId,
    Elements OPTIONAL }
TAPDUId ::= SEQUENCE {
    TAPDUNumber OPTIONAL,
    TAPDUName OPTIONAL }

    -- One of these must be present.
Elements ::= SEQUENCE {
    ElementId,
    ElementValues }
ElementId ::= SEQUENCE {
    ElementNumber OPTIONAL,
    ElementName OPTIONAL }

    -- One of these must be present.
ElementValues ::= SET OF Component
    -- See Note
Component ::= CHOICE {
    PrimitiveComponent,
    ConstructorComponent }
PrimitiveComponent ::= Parameter
ConstructorComponent ::= SET OF Parameter
Parameter ::= SEQUENCE {
    ParameterId OPTIONAL,
    ParameterValue }

```

Note – Order of components as prescribed by the TAPDU descriptions in § 10.

11.3 Coding rule

11.3.1 TAPDU ID

- 1) The TAPDU number assigned to TAPDU shall consist of two parts separated by a "period" (.). The first part identifies the application, for example, "3" is assigned to this application. The second part identifies the procedures specified in the application.
- 2) Where national requirements dictate the use of non-standardized TAPDU numbers. Administrations may choose any values in the range 1000-1999 for the first part of non-standardized application identifiers.
- 3) Other rules applied to TAPDU number and name are same as those of the element number and name, described below.

11.3.2 Element ID

- 1) The element number shall be sequentially assigned a different number.
- 2) An element number is always closed by the character "colon" (:).
- 3) There shall be no restriction of the number of digits for element numbers and any leading zeros are ignored.
- 4) Where national requirements dictate the use of non-standardized element numbers Administrations may choose any values in the range 1000-1999 for non-standardized elements.

- 5) The element number and the element name shall be separated by the character "space".
- 6) An element name shall be represented by a text string, that is a sequence of graphic characters. Capital and small characters have the same effect.
- 7) An element name is always closed by the character "colon" (:).

11.3.3 *Element value fields*

For unregistered TLM-users with international access, the pre-defined values as defined in the following tables have to be applied. For all other cases, these values can be replaced by nationally defined values.

11.3.4 *Separators and common rules*

- 1) TAPDU-Ids and elements shall be preceded by the following delimiters:
 - "CR LF" sequence, or
 - "CR LF BS +" sequence.
- 2) The Element-Id and the First component shall be separated by the character "space" or "New Line" functions ("NL" = "CR LF" or "LF CR").
- 3) Components shall be separated by "comma" (",") and optionally "NL".
- 4) When components with pre-defined and not pre-defined values are contained in an element, they shall be separated by a "NL" and the line with the pre-defined values should start with the character "=".
- 5) Parameters within one component field shall be separated by the character "slash" ("/") or "semi-colon" (";"). "CR LF" within a parameter is not allowed, except if the parameter is longer than 1 line.
- 6) The actual value of a parameter value is encoded by a sequence of graphic characters. Capital and small characters have the same effect.
- 7) If some pre-defined values are absent but required, then their default value shall apply.
- 8) The element ID and the first element value field shall be separated by the character "space" or the "NL" function.
- 9) Contiguous "NL" and "LF" are considered as one "NL".
- 10) Contiguous embedded spaces are considered as one space. Leading spaces in a line are ignored.
- 11) The character sequence "Space //" indicates that the following of the line shall be considered as a comment.

11.4 *Format of TAPDU*

The format of each TAPDU according to the above coding rules is shown in Annex C of this Recommendation.

11.5 *Reference between TAPDU components and its coding format*

This section provides the tables necessary for the encoding of TAPDU components.

11.5.1 *TAPDUId and elementId* (see Table 4/T.330)

Table 4/T.330 comprises four columns:

- 1) The first column contains the TAPDUId or Element-Id name as used in the ASN.1 description of § 10.
- 2) The second column contains the type of this element:
 - a) primitive: the element contains only one elementValue field;
 - b) constructor: the element may contain more than one elementValue field.
- 3) The third column contains the actual coding format of the TAPDUId or element-Id.
- 4) The last column contains remarks.

11.5.2 *ElementValues* (see Table 5/T.330)

Table 5/T.330 comprises five columns:

- 1) The first column contains the ElementValue name (component name) as used in the ASN.1 description of § 10.
- 2) The second column contains the type of ElementValue field:
 - a) primitive: the component contains only one parameter.
 - b) constructor: the component may contain more than one parameter.

- 3) The third column contains the type of the value:
 - a) predefined;
 - b) general, as defined in this section.
- 4) The fourth column contains the actual coding format, or, in case of general value, a reference name which points to the actual coding format in Table 6/T.330.
- 5) The last column contains remarks.

11.5.3 *General values* (see Table 6/T.330)

Table 6/T.330 comprises five columns:

- 1) The first column contains the reference name (general value name) used in Table 5/T.330.
- 2) The second column contains the name of the parameter.
- 3) The third column contains the code of the value.
- 4) The fourth column contains the keyword and format of this parameter,
- 5) The last column contains remarks.

TABLE 4/T.330

Format encoding of TAPDU and ElementId

TAPDUId and ElementId name	Type	T.61 Character coding format	Remarks
authorizing	Constructor	<u>21</u> : □ AUTHORIZING:	
autoFWDComment	Primitive	79 : □ AUTO-FWD-COMMENT:	
autoFWDHeading		78 : □ AUTO-FWD-HEADING:	For further study
autoFWDIPMs	Primitive	76 : □ AUTO-FWD-IPMS:	
autoFWDRecipients	Constructor	77 : □ AUTO-FWD-RECIPIENTS:	
autoOutput	Constructor	<u>60</u> : □ AUTO-OUTPUT:	
bcc	Constructor	<u>24</u> : □ BCC:	
bodyType	Constructor	<u>31</u> : □ BODY-TYPE:	
cancel	–	<u>3.13</u> : □ CANCEL:	
cc	Constructor	<u>23</u> : □ CC:	
comment	Primitive	<u>50</u> : □ COMMENT:	
contentIndicator	Constructor	<u>18</u> : □ CONTENT-INDICATOR:	
contentInfo	Primitive	<u>17</u> : □ CONTENT-INFO:	
contentReturned	–	<u>72</u> : □ CONTENT-RETURNED-INDICATION:	
conversion	Primitive	<u>16</u> : □ CONVERSION:	
conversionIndication	Constructor	<u>42</u> : □ CONVERSION-INDICATION:	
convertedInfoTypes	Primitive	<u>44</u> : □ CONVERTED-INFORMATION-TYPES:	
correlationInfo	Primitive	<u>1</u> : □ CORRELATION-INFORMATION:	
deleteAfterOutput	Primitive	<u>80</u> : □ DELETE-AFTER-OUTPUT:	
deliver	–	<u>3.3</u> : □ DELIVER:	
deliveryId	Primitive	<u>35</u> : □ DELIVERY-ID:	
deliveryStatusNotice	–	<u>3.4</u> : □ DELIVERY-STATUS-NOTICE:	
dsMode	Primitive	<u>58</u> : □ DS-MODE:	
dsQuery	–	<u>3.7</u> : □ DS-QUERY:	
dsReport	–	<u>3.8</u> : □ DS-REPORT:	
errors	Primitive	<u>9</u> : □ ERRORS:	
exception	–	<u>3.12</u> : □ EXCEPTION:	
expiredDiscard	Primitive	<u>73</u> : □ EXPIRED-DISCARD:	
explicitRN	–	<u>3.6</u> : □ EXPLICIT-RN:	
explicitRNack	–	<u>3.16</u> : □ EXPLICIT-RN-ACK:	
forwardedInfo	Constructor	<u>32</u> : □ FORWARDED-INFO:	
from	Primitive	<u>20</u> : □ FROM:	

TABLE 4/T.330 (cont.)

TAPDUId and ElementId name	Type	T.61 Character coding format	Remarks
orgIntendedRecipient	Primitive	<u>40</u> : <input type="checkbox"/> INTENDED-RECIPIENT:	
iPNOriginator	Primitive	<u>69</u> : <input type="checkbox"/> IPN-ORIGINATOR:	
language	Primitive	<u>53</u> : <input type="checkbox"/> LANGUAGE:	
latestDelivery	Primitive	<u>34</u> : <input type="checkbox"/> LATEST-DELIVERY:	
messageDelete	–	<u>3.18</u> : <input type="checkbox"/> MESSAGE-DELETE:	
messageDeleteMode	Primitive	<u>81</u> : <input type="checkbox"/> MESSAGE-DELETE-MODE:	
messageLength	Primitive	<u>37</u> : <input type="checkbox"/> MESSAGE-LENGTH:	
messageReturnedInd	–	<u>51</u> : <input type="checkbox"/> MESSAGE-RETURNED-INDICATION:	
messageSelector	Primitive	<u>82</u> : <input type="checkbox"/> MESSAGE-SELECTOR:	
messageStatus	Primitive	<u>83</u> : <input type="checkbox"/> MESSAGE-STATUS:	
messageType	Primitive	<u>52</u> : <input type="checkbox"/> MESSAGE-TYPE:	
msgIncomplete	–	<u>67</u> : <input type="checkbox"/> MSG-INCOMPLETE:	This element has not value
nonDeliveryReason	Primitive	<u>46</u> : <input type="checkbox"/> NON-DELIVERY-REASON:	
nonReceiptInfo	Primitive	<u>49</u> : <input type="checkbox"/> NON-RECEIPT-INFO:	
		<u>43</u> : <input type="checkbox"/> NOTIFICATION-TYPE:	
obsoletedDiscard	Primitive	<u>74</u> : <input type="checkbox"/> OBSOLETED-DISCARD:	
obsoletedIPMs	Constructor	<u>29</u> : <input type="checkbox"/> OBSOLETED:	
otherRecipients	Constructor	<u>41</u> : <input type="checkbox"/> OTHER-RECIPIENTS:	
outputMessage	–	<u>3.10</u> : <input type="checkbox"/> OUTPUT-MESSAGE:	
outputRequest	–	<u>3.9</u> : <input type="checkbox"/> OUTPUT-REQUEST:	
perMessageIndicators	Constructor	<u>19</u> : <input type="checkbox"/> FLAGS:	
preferredRecipient	Primitive	<u>70</u> : <input type="checkbox"/> PREFERRED-RECIPIENT:	
priority	Primitive	<u>13</u> : <input type="checkbox"/> PRIORITY:	
probe	–	<u>3.2</u> : <input type="checkbox"/> PROBE:	
probeAck	–	<u>3.15</u> : <input type="checkbox"/> PROBE-ACK:	
probeId	Primitive	<u>66</u> : <input type="checkbox"/> PROBE-ID:	
quantityOfDocs	Primitive	<u>62</u> : <input type="checkbox"/> QUANTITY-OF-DOCS:	
recipients	Constructor	<u>15</u> : <input type="checkbox"/> RECIPIENTS:	
receiptStatusNotice	–	<u>3.5</u> : <input type="checkbox"/> RECEIPT-STATUS-NOTICE:	
redirectedFrom	Constructor	<u>54</u> : <input type="checkbox"/> REDIRECTED-FROM:	
register	–	<u>3.11</u> : <input type="checkbox"/> REGISTER:	
registerAck	–	<u>3.17</u> : <input type="checkbox"/> REGISTER-ACK:	

TABLE 4/T.330 (end)

TAPDUId and ElementId name	Type	T.61 Character coding format	Remarks
related IPMs	Constructor	<u>28</u> : □ RELATED-IPMS:	
repliedToIPM	Primitive	<u>30</u> : □ REPLIED-TO-IPM:	
reply	Constructor	<u>25</u> : □ REPLY:	
reportedMessageId	Primitive	<u>75</u> : □ REPORTED-MESSAGE-ID:	
reportedRecipient	Primitive	<u>3</u> : □ REPORTED-RECIPIENT:	
retrievalId	Primitive	<u>38</u> : □ RETRIEVAL-ID:	
returnAddress	Primitive	<u>36</u> : □ RETURN-ADDRESS:	
send	–	<u>3.1</u> : □ SEND:	
sendAck	–	<u>3.14</u> : □ SEND-ACK:	
statusQuery	–	<u>3.19</u> : □ STATUS-QUERY:	
status Report	–	<u>3.20</u> : □ STATUS-REPORT:	
subject	Primitive	<u>26</u> : □ SUBJECT:	
subjectIPM	Primitive	<u>71</u> : □ SUBJECT-IPM:	
submissionId	Primitive	<u>65</u> : □ SUBMISSION-ID:	
submissionTime	Primitive	<u>33</u> : □ SUBMISSION-TIME:	
supplInfo	Primitive	<u>68</u> : □ SUPPLEMENTARY-INFORMATION:	
supplReceiptInfo	Primitive	<u>68</u> : □ SUPPLEMENTARY-INFORMATION:	
thisIPM	Primitive	<u>27</u> : □ THIS-IPM:	
thisRecipient	Primitive	<u>39</u> : □ THIS-RECIPIENT:	
timeOfDelivery	Primitive	<u>4</u> : □ TIME-OF-DELIVERY:	
timeOfReceipt	Primitive	<u>47</u> : □ TIME-OF-RECEIPT:	
timeOfReport	Primitive	<u>84</u> : □ TIME-OF-REPORT:	
tLMAUOperation	Constructor	<u>59</u> : □ TLMAU-OPERATION:	
to	Constructor	<u>22</u> : □ TO:	
typeOfReceipt	Primitive	<u>48</u> : □ TYPE-OF-RECEIPT:	
typeOfUA	Primitive	<u>45</u> : □ TYPE-OF-UA:	

Conventions:

- 1) Primitive: element contains only one element value field.
- 2) Constructor: element may contain more than one element value field.
- 3) abcde: underlined characters, i.e. "abc" are mandatory in case of international access (see § 11.2).
- 4) □: space character.

TABLE 5/T330

Format encoding of elements values

Element value name	Type of element value field	Type of value	T.61 character coding format	Remarks
alternate-recipient-allowed	Primitive	Predefined	<u>Allowed</u>	
authorizing	Constructor	–	<u>21</u> ; <input type="checkbox"/> AUTHORIZING:	
authorizing-user	Constructor	General	<i>OR Descriptor</i>	
auto-acknowledgment	Primitive	Predefined	<u>Auto-Receipt</u> , <u>Manual-Receipt</u> (default)	
auto-forwarded	Primitive	Predefined	<u>Auto-forwarded</u> , <u>Not-Auto-forwarded</u> (default)	
auto-fwd-Comment	Primitive	General	<i>any Text</i>	
auto-fwd-ipms	Primitive	Predefined	<u>Auto-forwarded</u> , <u>Not-Auto-forwarded</u> (default)	
auto-fwd-recipient-name	Constructor	General	<i>OR Name</i>	
blind-copy-recipient	Constructor	General	<i>OR Descriptor</i>	
body-part	Primitive	Predefined	<u>IA5Text</u> , <u>TLX</u> , <u>Voice</u> , <u>G3Fax</u> , <u>G4Fax-Class1</u> , <u>TTX</u> , <u>Videotex</u> , <u>Message</u> , <u>Mixed-Mode</u> , <u>Encrypted</u>	
call-id	Primitive	General	<i>Call Identification</i>	
comments	Primitive	General	<i>Comments</i>	
content-return-request	Primitive	Predefined	<u>Content-Return-Request</u>	
conversion-info			<u>NO</u> , <u>Yes</u> ; <u>WLOSS</u>	
copy-recipient	Constructor	General	<i>OR Descriptor</i>	
deferred-delivery-time	Primitive	General	<i>Date and Time</i>	
delete-after-output	Primitive	Predefined	<u>Keep</u> , <u>Delete</u>	
delivery-msg-id	Primitive	General	<i>Message Identifier</i>	
	Primitive	General	<i>Date and Time</i>	
diagnostic-code ^{a)}	Primitive	Predefined	<u>Unrecognized-OR-Name</u> , <u>Ambiguous-OR-Name</u> , <u>MTS-Congestion</u> , <u>Loop-Detected</u> , <u>Recipient-Unavailable</u> , <u>Maximum-Time-Expired</u> , <u>Content-Too-Long</u> , <u>Conversion-Impractical</u> , <u>Encoded-Information-Type-Unsupported</u> , <u>Conversion-Prohibited</u> , <u>Invalid-Arguments</u> , <u>Implicit-Conversion-Not-Subscribed</u> , <u>Content-Syntax-Error</u> , <u>Pragmatic-Constraint-Violation</u> , <u>Protocol-Violation</u> , <u>Content-Not-Supported</u> , <u>Too-Many-Recipient</u> , <u>No-Bilateral-Agreement</u>	

TABLE 5/T.330 (cont.)

Element value name	Type of element value field	Type of value	T.61 character coding format	Remarks
discard-ipm	Primitive	Predefined	<u>Discard</u> (default), <u>Not-Discard</u>	
discard-reason	Primitive	Predefined	<u>IPM-Expired</u> , <u>IPM-Obsoleted</u> , <u>User-Subscription-Terminated</u>	
disclose-recipients	Primitive	Predefined	<u>No-Disclosure</u>	
dsMode	Primitive	–	58:☐ DS-MODE:	
ds-mode	Primitive	Predefined	<u>Auto-Output</u> , <u>Retrieval</u>	
eIT	Primitive	Predefined	<u>IA5Text</u> , <u>TLX</u> , <u>Voice</u> , <u>G3Fax</u> , <u>G4Fax-Class1</u> , <u>TTX</u> , <u>Videotex</u> , <u>Undefined</u> , <u>Mixed-Mode</u>	
error-cause	Primitive	Predefined	<u>IPMS-Element-of-Service-Not-Subscribed</u> ,* <u>MTS-Element-of-Service-Not-Subscribed</u> ,* <u>Name-Malformed</u> ,* <u>IPM-Not-Submitted</u> , <u>IPM-Transferred</u> , <u>IPM-Delivered</u> , <u>Element-of-Service-Not-Subscribed</u> ,* <u>Message-Delivered</u> , <u>Message-Transferred</u> , <u>Originator-Invalid</u> ,* <u>Query-Identifier-Invalid</u> ,* <u>Recipient-Improperly-Specified</u> ,* <u>Submission-Identifier-Invalid</u> ,* <u>NO-Message-in-DS</u> , <u>DS-Not-Supported</u> , <u>DS-Not-Subscribed</u> , <u>Retrieval-Identifier-Invalid</u> ,* <u>Parameter-Invalid</u> ,* <u>Not-Changed</u>	* optionally followed by the name, service, parameter, etc. concerned in “ ... ”
error-recovery-mode	Primitive	Predefined	<u>Recovery-1</u> , <u>Recovery-2</u> , <u>Recovery-3</u>	
expiry-time	Primitive	General	<i>Date and Time</i>	
explicit-conversion	Primitive	Predefined	<u>TLX</u> , <u>IA5</u> , <u>G3</u> , <u>G4</u> , <u>VTX</u> , <u>TTX</u>	
forwarded-time	Primitive	General	<i>Date and Time</i>	
frequency	Primitive	General	<i>Frequency</i>	
importance	Primitive	Predefined	<u>Low</u> , <u>Normal</u> (default), <u>High</u>	
intended-recipient-name	Constructor	General	<i>OR Name</i>	
ipn-originating-user	Constructor	General	<i>OR Descriptor</i>	
language-ind	Primitive	Predefined		ffs
latest-delivery-time	Primitive	General	<i>Date and Time</i>	
message-delete-mode	Primitive	Predefined	<u>Auto-Delete</u> (default), <u>Manual-Delete</u>	
message-length	Primitive	General	<i>Message Length</i>	
messageType	Primitive	–	52:☐ MESSAGE-TYPE:	
non-receipt-reason	Primitive	Predefined	<u>IPM-DISCARD</u> , <u>IPM-Auto-forwarded</u>	
nrn-request	Primitive	Predefined	<u>NRN-Request</u>	
number-of-docs	Primitive	General	<i>Number Of Associated Documents</i>	

TABLE 5/T.330 (cont.)

Element value name	Type of element value field	Type of value	T.61 character coding format	Remarks
obsoleted-ipm-id	Constructor	General	<i>IPM Identifier</i>	
originating-user	Constructor	General	<i>OR Descriptor</i>	
originator-name	Constructor	General	<i>OR Name</i>	
originator-requested-alternate-recipient	Constructor	General	<i>OR Name</i>	
other-recipient-name	Constructor	General	<i>OR Name</i>	
output-time	Primitive	General	<i>Date and Time</i>	
Physical-delivery-mode	Primitive	Predefined	"PDM= "OM (default), EMS, SPEC, COL, TLXA, TTXA, PHA, BFA	
Physical-delivery-report-request	Primitive	Predefined	"REP= "UND (default), PDS, MHS, PDMHS	
Physical-forwarding-address-request	Primitive	Predefined	PFAR	
Physical-forwarding-prohibited	Primitive	Predefined	PF	
postal-address	Primitive	General	<i>Or Name</i>	
preferred-recipient	Constructor	General	<i>OR Descriptor</i>	
primary-recipient	Constructor	General	<i>OR Descriptor</i>	
priority-ind	Primitive	Predefined	<u>Urgent</u> , <u>Non-Urgent</u> , <u>Normal</u> (default)	
probe-msg-id	Primitive	General	<i>Message Identifier</i>	
reason-code	Primitive	Predefined	<u>Transfer-Failure</u> , <u>Unable-To-Transfer</u> , <u>Conversion-Not-Performed</u>	
receipt-time	Primitive	General	<i>Date and Time</i>	
recipient-name	Constructor	General	<i>OR Name</i>	
recipient-number-for-advice	Primitive	General	"CALL= "Number	
recipient-reassignment-prohibited	Primitive	Predefined	RRP	
redirected-from	Constructor	General	<i>OR Name</i>	
registered-mail-type	Primitive	Predefined	NRM (default), RM; RMA	
related-ipm-id	Constructor	General	<i>IPM Identifier</i>	
replied-to-ipm-id	Constructor	General	<i>IPM Identifier</i>	
reply-recipient	Constructor	General	<i>OR Descriptor</i>	
reply-request	Primitive	Predefined	<u>Reply</u> , <u>No-Reply</u> (default)	
reply-time	Primitive	General	<i>Date and Time</i>	
reported-message-id	Primitive	General	<i>Message Identifier</i>	

TABLE 5/T.330 (end)

Element value name	Type of element value field	Type of value	T.61 character coding format	Remarks
reported-recipient-name	Constructor	General	<i>OR Name</i>	
report-time	Primitive	General	<i>Date and Time</i>	
report-type	Primitive	Predefined	<u>Receipt</u> , <u>Non-Receipt</u> , <u>Delivery</u> , <u>Non-delivery</u>	
requested-delivery-method	Primitive	Predefined	"RDL= "ANY (default), MAS, PD, TLX, TTX, G3, G4 IA5, VTX	
retrieval-id	Primitive	General	<i>Retrieval Identifier</i>	
return-request	Primitive	Predefined	<u>Return-Request</u>	
rn-request	Primitive	Predefined	<u>RN-Request</u>	
sensitivity	Primitive	Predefinde	<u>Personal</u> , <u>Private</u> , <u>Company-Confidential</u>	
status	Primitive	Predefined	<u>In-Process</u> , <u>Delivered</u> , <u>Non-Delivered</u>	
subject-content	Primitive	General	<i>Subject</i>	
subject-ipm-id	Primitive	General	<i>Message Identifier</i>	
submission-msg-id	Primitive	General	<i>Message Identifier</i>	
submissionTime	Primitive	–	33:□ SUBMISSION-TIME:	
suppl-Info	Primitive	General	<i>Supplementary Information</i>	
suppl-receipt-Info	Primitive	General	<i>Supplementary Infomation</i>	
terminal-type	Primitive	Predefined	"TTyp=" TLX, TTX, G3, G4, IA5, vtx	
this-recipient-name	Constructor	General	<i>OR Name</i>	
this-ipm-id	Constructor	General	<i>IPM Identifier</i>	
type-of-receipt	Primitive	Predefined	<u>Manual</u> (default), <u>Automatic</u>	
type-of-ua	Primitive	Predefined	<u>Private</u> , <u>Public</u> (default)	
user-report-request	Primitive	Predefined	<u>No-Report</u> , <u>Non-Delivery-Report</u> , <u>Report</u>	

a) More diagnostic-codes can be found in Recommendation X.411 and should be translated into appropriate T.61 text.

Note – The character strings in italics in the fourth column are the entry name of the General value list, Table 6/T.330.

Conventions:

- 1) Primitive: elementary or value contains only one component.
- 2) Constructor: element or value contains more than one component.
- 3) abcde: underlined characters, i.e. "abc" are mandatory.

TABLE 6/T.330

General value list

General value		Value attribute		Remarks
Name	Parameter name	Code ^{a)}	Format/Keyword	
Call identification			Each component is separated by the character “ / ”	
	TLMAU TID	T.61	Defined in Rec. F.200	
	TLM TID	T.61	Defined in Rec. F.200	
	Date and time	P	YY-MM-DD-HH:mm	
	Document rel. No.	N	001 ~ 999	
	Additional session rel. No.	N	01 ~ 99	
Comments	—	P		
Date and time	—	P	YY-MM-DD-HH:mm	
Frequency	—	N	In minutes	
IPM identifier			<IPM Identifier> ::= <OR Name> “/” <Local Message ID> “>”	
	Or name		See OR name	
	Local message ID	AI5	“LID=” <Local Message ID>	
Message identifier	—	P		
Message length	—	N	In octets	
Number of associated documents	—	N		
OR descriptor	OR name		See OR name	
	Free form name	T.61	“Free Form Name=” <Free Form Name> “FM=” <Free Form Name>	
	Telephone number	P	“Telephone Number=” <Telephone Number> “TEL=” <Telephone Number>	

TABLE 6/T.330 (cont.)

General value	Value attribute			Remarks
Name	Parameter name	Code ^{a)}	Format/Keyword	
OR name			<OR Name> ::= <Standard Attribute Lists> <Domain Defined Attribute List>	
	Standard attribute list		<Standard Attribute List> ::= “<”<Keyword.Att> (“;” <Keyword.Att>)* “>”	
	Country name	N/P	“Country Name=” <Country Name> “CN=” <Country Name>	Default: Country of TLMAU
	Administration domain name	N/P	“Administration Domain Name=” <Administration Domain Name> “ADMD=” <Administration Domain Name>	Default: ADMD of TLMAU
	Network address	N	“X121 Address=” <Network Address> “X121 =” <Network Address> “Network Address=” <Network Address>	
	Terminal identifier	P	“Terminal ID=” <Terminal Identifier> “TID=” <Terminal Identifier>	
	Private domain name	N/P	“Private Domain Name=” <Private Domain Name> “PRMD=” <Private Domain Name>	
	Organisation name	P	“Organization Name=” <Organization Name> “ON=” <Organization Name>	
	Numeric user identifier	N	“User Agent ID=” <Numeric User Identifier> “UAID=” <Numeric User Identifier> “NUID=” <Numeric User Identifier>	
	Surname	P	“SurName=” <SurName> “SN=” <SurName>	
	Given name	P	“Given Name=” <Given Name> “GN=” <Given Name>	
	Initials	P	“Initials=” <Initial> “I=” <Initial>	
	Generation qualifier	P	“Generation Qualifier=” <Generation Qualifier> “GQ=” <Generation Qualifier>	
	Organization unit name	P	“Organization Unit Name=” <Organization Unit Name> “OU=” <Organization Unit Name>	
	Domain defined attribute list		<Domain Defined Attributes List> ::= “<DDA=” <Type> “;” <Value> (“;” <Type> “;” <Value>)* “>” “<Domain Defined Attributes=” <Type> “;” <Value> (“;” <Type> “;” <Value>)* “>”	
	Type	P		
Value	P			

TABLE 6/T.330 (cont.)

General value	Value attribute			
Name	Parameter name	Code ^{a)}	Format/Keyword	Remarks
OR name (continued)	Postal address			
	PDS-name	P	"PDSN=" <value>	
	Physical delivery country name	N/P	"PDCN=" <value>	
	Postal code	N/P	"PC=" <value>	
	Physical delivery office name	P	"PDNA=" <value>	
	Physical delivery office number	P	"PDNU=" <value>	
	Extension OR address components	P	"EOA=" <value>	
	Physical delivery personal name	P	"PNP=" <value>	
	Physical delivery organization name	P	"ONP=" <value>	
	Extension physical delivery address components	P	"EPD=" <value>	
Unformatted postal address	P	"UPA=" <value>	Max. 6 lines, max. 30 characters per line separated by "-"	
Street address	P	"STA=" <value>		
Post office box address	P	"POB=" <value>		
Poste restante address	P	"PRA=" <value>		
Unique postal name	P	"UN=" <value>		
Local postal attributes	P	"LPA=" <value>		

TABLE 6/T.330 (end)

General value	Value attribute			
Name	Parameter name	Code ^{a)}	Format/Keyword	Remarks
Retrieval identifier	–	N/P		
Subject	–	T.61		
Supplementary information	–	P		

^{a)} N = Numeric string in T.61 character set; P = Printable string in T.61 character set; AI5 = string in T.61 character set; T.61 = T.61 string.

^{b)} Descr.Att in OR Descriptor and Keyword Att in OR Name contains a Parameter-Id and a parameter-value. The Parameter-Ids appear with bold characters in this table.

Note 1 – Syntactical conventions are defined as follows and the character size does not make any sense:

<...> Represents a syntactical item, non-terminal

(...)* Iteration

| Exclusive or alternatives

"abc" Explicit characters abc.

Note 2 – Maximum length of parameter values can be found in X.400-Series Recommendations.

12 Error recovery

This section specifies the error recovery mechanisms for errors which may occur during the communication between a TLM terminal and a TLMAU. The type of recovery will depend upon whether the error occurred during the transmission of control information or a normal document. The recovery mechanism of Recommendation T.62 shall apply. In addition, the qualifications given in the following paragraphs shall apply.

For both directions of transfer, if an error is detected during the transmission of a document containing control information, the transmitted part shall be discarded and the complete TAPDU retransmitted.

If an error is detected during the transmission of a document containing message body, the recovery mechanism depends upon the direction of transfer. In the TLMAU-to-TLM terminal direction, two possibilities for recovery are available, the first one is to apply the rules of Recommendation T.62 using CDC, the second one is the complete retransmission of the TAPDU. In the TLM terminal-to-TLMAU direction, three possible recovery mechanisms are identified, the choice of which is a national matter:

- 1) No recovery is provided. The complete TAPDU (control information and message body) must be retransmitted.
- 2) The TLMAU shall retain the received part up to the last acknowledged checkpoint. The originating TLM terminal is required to resume the sending of the remainder of the TAPDU by using CDC protocol element as described in Recommendation T.62. The TLMAU shall automatically link the retained part with the resumed part prior to sending to the MTS. If the originating TLM terminal does not resume the sending within a predetermined time or if the TLMAU receives new TAPDU within that time, the TLMAU shall discard the received part of the TAPDU.
- 3) The last acknowledged normal document is considered the end of the TAPDU. The TLMAU will send this part to the MTS as a complete message. If the TLM terminal wishes to resume transmission of the interrupted TAPDU, it must first resend the control information. There is no correlation between the original message and the continued message at a recipient TLMAU or IPM-UA.

Note 1 – For failures occurring between document boundaries, the last acknowledged document shall be regarded as the end of the current TAPDU.

Note 2 – A TAPDU, which may be composed of more than one document, cannot be considered complete before at least two documents were received.

Note 3 – If there is a relationship between the sequence of normal documents (for example a multi-Part body message) being transferred, the recovery mechanism 2 should not be used.

13 Control procedures

13.1 *Session control procedure*

The session control procedures shall be in accordance with Recommendation T.612. However, the qualifications listed in the following subsections shall also apply.

13.1.1 *Session element of procedure*

The values of mandatory parameters used during session establishment shall be as given in Table 7/T.330. The format of the TLMAU terminal identifier shall be that specified in Recommendation F.200.

13.1.2 *Session rules*

Change control (CSCC/RSCCP exchange) may occur inside TAPDU boundaries but should occur outside document boundaries.

13.2 *Document control procedures*

The document control procedures shall be in accordance with Recommendation T.62.

Control information is conveyed in either a control document or a normal document. Where encoded information types of the control information is Teletex, the body of the IP-message is conveyed in a sequence (at least one) of normal documents.

Note – The use of the normal document for conveying control information should be restricted to Teletex access to IPMS.

TABLE 7/T.330

Mandatory parameter values during session establishment

T.62 parameters	Call Originator	
	TLM terminal	TLMAU
Terminal identifier of calling terminal (in CSS)	TLM TID	TLMAU's TID
Terminal identifier of called terminal (in RSSP)	TLMAU TID	TLM TID
Date and time	Date and time the TLM terminal originated call	Date and time the TLMAU terminal originated call
Service identifier	Telematic	Telematic
Window size	Note 1	Note 1
Session user date	Basic terminal capabilities and interchange format(s) which are available as receiving capabilities of the TLM terminal (Note 1)	Basic terminal capabilities and interchange format(s) which are available as receiving capabilities of the TLMAU (Note 1)

Note 1 – This parameter is mandatory for the Group 4 facsimile service.

Note 2 – TID means terminal identification.

Note 3 – Negotiation of the capabilities of handling control document encoded in T.61 characters is not needed.

In a multiple-TAPDU session, the TAPDUs are delimited by at least one of the following methods:

- 1) implicitly, that is, by the first document of the TAPDU being a control document;
- 2) explicitly, that is, by the first document of the TAPDU containing a parameter indicating the number of subsequence documents in the TAPDU.

Note – The second method is mandatory for multiple-action-unit session, where the first document of the TAPDU is a normal document.

When a single TAPDU is sent on a session connection, the TAPDU is delimited only by the end of the session.

13.3 *Log-on procedure*

It is conceivable that many subscribers will commonly use a single TLM terminal, some mechanism is required to identify a specific user by intermediate system, i.e. log-on procedure using user name and password. Detailed procedure requires further study.

ANNEX A

(to Recommendation T.330)

Formal definition of TLMA abstract service

This Annex is an integral part of this Recommendation.

This Annex, a supplement to § 7, formally defines for reference purposes the abstract service TLMA (telematic agent). It uses the PORT and ABSTRACT-BIND, -UNBIND, -OPERATION, and -ERROR macro of Recommendation X.407.

```

TLMAAbsService {ccitt-t330}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue
EXPORTS

    -- Primary port types
    miscellanea

IMPORTS

    -- Abstract service
    origination, reception, management
    FROM IPMSAbstractService {joint-iso-ccitt
        mhs(6) imps(1) modules(0) abstract-service(3) }
    import, export
    FROM MTSAbstractService {joint-iso-ccitt
        mhs(6) mts(3) modules(0) mTS-abstract-service(3) }

    -- Information objects and their aspects
    IPM, RN, NRN
    FROM IPMSInformationObjects {joint-iso-ccitt
        mhs(6) ipms(1) modules(0) information-objects(2) }

    -- Functional objects
    tlma
    FROM IPMSFunctionalObjects {joint iso-ccitt
        mhs(6) ipms(1) modules(0) functional-objects(1) }

    -- MTAS aspects
    MessageDeliveryEnvelope, ORName, MessageIdentifier, Priority, Time, ReportDelivery-Envelope,
    PerRecipientReportDeliveryFields
    FROM MTSAbstractService {joint-iso-ccitt
        mhs(6) mts(3) modules(0) mTS-abstract-service(3) }

    --Abstract services macros
    OBJECT, PORT, ABSTRACT-BIND, ABSTRACT-ERROR, ABSTRACT-OPERATION
    From AbstractServiceNotation {joint-iso-ccitt
        mhs(6) adsd(2) modules(0) notation(1) }

DatenandTime ::= Time

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

-- Ports
miscellanea PORT
    SUPPLIER PERFORMS {
        ChangeSubscriptionProfile,
        DSLList,
        DSDelete,
        DSFetch,
        MessageStatus }
    ::= id-pt-miscellanea

```

TLMAAbsService (continuation)

-- *Miscellanea port specific abstract error problems*

SubscriptionProfileProblem ::= ENUMERATED {
not-changed(0) }

DSProblem ::= ENUMERATED {
no-message-in-ds(0),
ds-not-supported(1),
ds-not-subscribed(2),
retrieval-identifier-invalid(3),
parameter-invalid(4) }

MessageStatusProblem ::= ENUMERATED {
query-identifier-invalid(0)

-- *RetrievalIdentifier*

RetrievalIdentifier ::= PrintableString

-- *CallIdentification*

CallIdentification ::= SEQUENCE {
TLMAU TID [0] TerminalIdentifier,
TLM TID [1] TerminalIdentifier,
date-and-time [2] DateAndTime,
document-ref-num [3] DocumentReferenceNumber,
additional-session-ref-num [4] AdditionalSessionReferenceNumber OPTIONAL }

TerminalIdentifier ::= PrintableString -- *defined in Recommendation F.200* --

DocumentReferenceNumber ::= NumericString

AdditionalSessionReferenceNumber ::= NumericString

-- *Change subscription profile abstract operation component types*

DSMode ::= SET {
[0] Mode,
frequency [1] Frequency OPTIONAL,
time [2] DateandTime OPTIONAL }

Mode ::= CHOICE {
[0] auto-output,
[1] retrieval }

Frequency ::= NumericString -- *in minutes* --

ErrorRecoveryMode ::= CHOICE { -- *see § 12 in this Recommendation* --
[0] recovery-1, -- *no-recovery* --
[1] recovery-2, -- *full-support-of-T.62-recovery-procedures* --
[2] recovery-3 -- *recovery-of-document-basis* -- }

MessageDeleteMode ::= CHOICE {
[0] auto-delete,
[1] manual-delete }

-- *DS List abstract operation component types*

ListReport ::= SET {
retrieval-id [0] RetrievalIdentifier,
message-type [1] MessageType,
priority [2] Priority OPTIONAL,
message-length [3] MessageLength OPTIONAL,
originator-name [4] ORName OPTIONAL }

MessageType ::= CHOICE {
[0] ipm,
[1] receipt-notice,
[2] non-receipt-notice,
[3] report }

MessageLength ::= NumericString -- *in octet* --

TLMAAbsService (continuation)

-- DS Fetch abstract operation component types

DeleteAfterOutput ::= CHOICE {
 [0] delete,
 [1] keep }

MessageReport ::= CHOICE {
 ipm-fetch [0] IPMFetch,
 rn-fetch [1] RNFetch,
 nrn-fetch [2] NRNFetch,
 report-fetch [3] ReportFetch }

IPMFetch ::= SEQUENCE {
 envelope[0] MessageDeliveryEnvelope,
 content [1] IPM }

RNFetch ::= SEQUENCE {
 envelope[0] MessageDeliveryEnvelope,
 -- not used intended-recipient-name and other recipient-name of deliver-envelope in RNFetch --
 content [1] RN }

NRNFetch ::= SEQUENCE {
 [0] number-of-docs [0] NumberOfAssociatedDocument OPTIONAL,
 [1] SET {
 envelope [1] MessageDeliveryEnvelope,
 -- not used intended-recipient-name and other recipient-name of deliver-envelope in NRNFetch --
 content [2] NRN }

ReportFetch ::= SEQUENCE {
 [0] SET {
 number-of-docs [0] NumberOfAssociatedDocuments OPTIONAL,
 call-id [1] CallIdentification },
 [1] SET {
 envelope [0] ReportDeliveryEnvelope,
 returned-content [1] IPM OPTIONAL } }

NumberOfAssociatedDocuments ::= NumericString

-- Message status abstract service

QueryIdentifier ::= CHOICE {
 submission-id [0] MessageIdentifier,
 call-id [1] CallIdentification }

StatusInfo ::= SET {
 [0] Status,
 per-recipient-info [1] PerRecipientReportDeliveryFields OPTIONAL }

Status ::= CHOICE {
 [0] in-process,
 [1] delivered,
 [2] non-delivered }

-- Miscellanea abstract-operations

ChangeSubscriptionProfile ::= ABSTRACT-OPERATION

 ARGUMENT SET {
 ds-mode [0] DSMode OPTIONAL,
 error-recovery-mode [1] ErrorRecoveryMode OPTIONAL,
 message-delete-mode [2] MessageDeleteMode OPTIONAL }

 RESULT { }

 ERRORS {
 name-error,
 ds-error,
 subscription-profile-error }

TLMAAbsService (continuation)

DSList ::= ABSTRACT-OPERATION

```
    ARGUMENT {}
    RESULT SET {
        list-reports [0] SET OF ListReport OPTIONAL }
    ERRORS {
        subscription-error,
        name-error,
        ds-error }
```

DSDelete ::= ABSTRACT-OPERATION

```
    ARGUMENT SET {
        selector [0] SET OF RetrievalIdentifier }
    RESULT
    ERRORS {
        subscription-error,
        name-error,
        ds-error }
```

DSFetch ::= ABSTRACT-OPERATION

```
    ARGUMENT SET {
        selector [0] SET OF RetrievalIdentifier,
        delete-after-output [1] DeleteAfterOutput OPTIONAL }
    RESULT SET {
        retrieval-id [0] RetrievalIdentifier,
        message-report [1] MessageReport }
    ERRORS {
        subscription-error,
        name-error,
        ds-error }
```

MessageStatus ::= ABSTRACT-OPERATION

```
    ARGUMENT SET {
        [0] QueryIdentifier }
    RESULT SET {
        report-time [0] DateandTime,
        reported-message-id [1] MessageIdentifier,
        [2] SET OF StatusInfo }
    ERRORS {
        subscription-error,
        name-error,
        message-status-error }
```

-- *Miscellanea port specific abstract errors*

subscription-profile-error ABSTRACT-ERROR

```
    PARAMETER SET {
        problem [0] SubscriptionProfileProblem }
    ::= 0
```

ds-error ABSTRACT-ERROR

```
    PARAMETER SET {
        problem [0] DSProblem }
    ::= 1
```

message-status-error ABSTRACT-ERROR

```
    PARAMETER SET {
        problem [0] MessageStatusProblem }
    ::= 2
```

END -- *of TLMAAbsService*

Format of TAPDU components

In this Annex the formats of the control information for different examples TAPDUS are shown. The principles of encoding are given in § 11. The formats of elements as defined in § 11 are illustrated by the use of the following four different syntax elements:

- 1) The elements number field is represented by a sequence of numeric graphic characters.
- 2) The element name field is represented by a text string giving the CCITT language reference name of the field. The actual value shall be a language-dependent representation of that reference name.
- 3) Separators are shown as they shall be represented in the actual control information.
- 4) Element value fields are shown in square brackets ("[" and "]"). The actual parameter values are described in § 11.

Note - The examples shown below are not exhaustive. Not all types (element value of the TAPDU description in § 10 (ASN.1 description) are included.

B.1 Conventions

Following symbols are used in this Annex:

- 1) NL new line function such as (CR LF or LF CR) or LF.
- 2) □ space character.
- 3) [] element value field. Actual values are defined in § 11 of this Recommendation.
- 4) [] iteration.

B.2 Example of TAPDU**B.2.1 Send-TAPDU**

- 3.1: □ SEND:
- 62: □ QUANTITY-OF-DOCS: □ [number-of-docs]
- 13: □ PRIORITY: □ [priority-ind]
- 19: □ FLAGS:
[deferred-delivery-time] 'NL'
= [disclose-recipients],[alternate-recipient-allowed]
- 16: □ CONVERSION: □ [conversion-prohibited]
- 17: □ CONTENT-INFO: □ [content-return-request]
- 27: □ THIS-IPM:
[this-ipm-id]
- 20: □ FROM:
[originating-user]
- 21: □ AUTHORIZING:
[[authorizing-user] 'NL']
- 22: □ TO:
[[primary-recipient] 'NL']
=[user-report-request],[rn-request],[nrn-request],[return-request],[reply-request] 'NL']

- 23: CC:
[[copy-recipient] 'NL'
=[rn-request],[lnrn-request],[return-request],[reply-request] 'NL']
- 24: BCC:
[[blind-copy-recipient] 'NL']
- 30: REPLIED-TO-IPM:
[replied-to-ipm-id]
- 29: OBSOLETES:
[[obsoleted-ipm-id] 'NL']
- 28: RELATED-IPMS:
[[related-ipm-id] 'NL']
- 26: SUBJECT:
[subject-content]
- 18: CONTENT-INDICATOR:
[expiry-time] 'NL'
= [importance],[sensitivity]
- 25: REPLY:
[reply-time] 'NL'
[[reply-recipient] 'NL']
- 31: BODY-TYPE: [[body-part],]

B.2.2 *SendAck-TAPDU*

- 3.14: SEND-ACK:
- 1: CORRELATION-INFORMATION:
[call-id]
- 65: SUBMISSION-ID:
[submission-msg-id]
- 33: SUBMISSION-TIME:
[submission-time]

B.2.3 *Exception-TAPDU*

- 3.12: EXCEPTION:
- 1: CORRELATION-INFORMATION:
[call-id]
- 9: ERRORS: [error-cause]

B.2.4 *Probe-TAPDU*

- 3.2: PROBE:

B.2.5 *ProbeAck-TAPDU*

- 3.15:□ PROBE-ACK:
- 1:□ CORRELATION-INFORMATION:
[call-id]
- 66:□ PROBE-ID:
[probe-msg-id]
- 33:□ SUBMISSION-TIME:
[submission-time]

B.2.6 *ExplicitRN-TAPDU*

- 3.6:□ EXPLICIT-RN:
- 15:□ RECIPIENTS:
[recipient-name]
- 13:□ PRIORITY: □ [priority-ind]
- 71:□ SUBJECT-IPM:
[subject-ipm-id]
- 69:□ IPN-ORIGINATOR:
[ipn-originating-user]
- 47:□ TIME-OF-RECEIPT:
[receipt-time]
- 44:□ CONVERTED-INFORMATION-TYPES:
[[eIT],]

B.2.7 *ExplicitRNAck-TAPDU*

- 3.16:□ EXPLICIT-RN-ACK:
- 1:□ CORRELATION-INFORMATION:
[call-id]
- 65:□ SUBMISSION-ID:
[submission-msg-id]
- 33:□ SUBMISSION-TIME:
[submission-time]

B.2.8 *Cancel-TAPDU*

- 3.13:□ CANCEL:
- 65:□ SUBMISSION-ID:
[submission-msg-id]
- 1:□ CORRELATION-INFORMATION:
[call-id]

B.2.9 *Deliver-TAPDU*

- 3.3:□ DELIVER:
- 62:□ QUANTITY-OF-DOCS:□[number-of-docs]
- 13:□ PRIORITY:□[priority-ind]
- 14:□ ORIGINATOR:
[originator-name]
- 39:□ THIS-RECIPIENT:
[this-recipient-name]
- 40:□ INTENDED-RECIPIENT:
[intended-recipient-name]
- 41:□ OTHER-RECIPIENTS:
[[other-recipient-name] 'NL']
- 33:□ SUBMISSION-TIME:
[submission-time]
- 35:□ DELIVERY-ID:
[delivery-msg-id]
- 42:□ CONVERSION-INDICATION:
[[eIT],] [conversion-prohibition]
- 44:□ CONVERTED-INFORMATION-TYPES:
[[eIT],]
- 27:□ THIS-IPM:
[this-ipm-id]
- 20:□ FROM:
[originating-user]
- 21:□ AUTHORIZING:
[[authorizing-user] 'NL']
- 22:□ TO:
[[primary-recipient] 'NL'
=[rn-request],[nrn-request],[return-request],[reply-request] 'NL']
- 23:□ CC:
[[copy-recipient] 'NL'
=[rn-request],[nrn-request],[return-request],[reply-request] 'NL']
- 24:□ BCC:
[[blind-copy-recipient] 'NL'
=[rn-request],[nrn-request],[return-request],[reply-request] 'NL']
- 30:□ REPLIED-TO-IPM:
[replied-to-ipm-id]
- 29:□ OBSOLETES:
[[obsoleted-ipm-id] 'NL']
- 28:□ RELATED-IPMS:
[[related-ipm-id], 'NL']
- 26:□ SUBJECT:
[subject-content]

- 18:☐ CONTENT-INDICATOR:
[expiry-time] 'NL'
=[importance],[sensitivity],[autoforwarded]
- 25:☐ REPLY:
[reply-time], 'NL'
[[reply-recipient] 'NL']
- 67:☐ EXTENSIONS:☐
[[extension-type] 'NL'
[extension-value] 'NL']
- 31:☐ BODY-TYPE:☐[body-part]
- 32:☐ FORWARDED-INFO:
[forwarded-time] 'NL'

//The delivery-envelope follows.

B.2.10 *ReceiptStatusNotice-TAPDU*

- 3.5:☐ RECEIPT-STATUS-NOTICE:
- 62:☐ QUANTITY-OF-DOCS:☐[number-of-docs]
- 13:☐ PRIORITY:☐[priority-ind]
- 35:☐ DELIVERY-ID:
[delivery-id]
- 14:☐ ORIGINATOR:
[originator-name]
- 39:☐ THIS-RECIPIENT:
[this-recipient-name]
- 33:☐ SUBMISSION-TIME:
[submission-time]
- 4:☐ TIME-OF-DELIVERY:
[delivery-time]
- 42:☐ CONVERSION-INDICATION:
[[eIT],] [conversion-prohibition]
- 44:☐ CONVERTED-INFORMATION-TYPES:
[[eIT],]
- 43:☐ NOTIFICATION-TYPE:☐[report-type]
- 71:☐ SUBJECT-IPM:
[subject-ipm-id]
- 69:☐ IPN-ORIGINATOR:
[ipn-originating-user]
- 70:☐ PREFERRED-RECIPIENT:
[preferred-recipient]

//The following three elements appear in case of RN.
- 47:☐ TIME-OF-RECEIPT:
[receipt-time]

- 48:□ TYPE-OF-RECEIPT:
[type-of-receipt]
- 68:□ SUPPLEMENTARY-INFORMATION:
[suppl-receipt-info]
//The following three elements appear in case of NRN.
- 49:□ NON-RECEIPT-INFO:
[non-receipt-reason],[discard-reason]
- 50:□ COMMENTS:
[comments]
- 51:□ MESSAGE-RETURNED-INDICATION:

B.2.11 *DeliveryStatusNotice-TAPDU*

- 3.4:□ DELIVERY-STATUS-NOTICE:
- 62:□ QUANTITY-OF-DOCS:□[number-of-docs]
- 1:□ CORRELATION-INFORMATION:
[call-id]
- 65:□ SUBMISSION-ID:
[submission-msg-id]
- 66:□ PROBE-ID:
[probe-msg-id]
[//repeated for each recipient
- 3:□ REPORTED-RECIPIENT:
[reported-recipient-name]
- 43:□ NOTIFICATION-TYPE:□[report-type]
- 40:□ INTENDED-RECIPIENT:
[intended-recipient-name]
- 44:□ CONVERTED-INFORMATION-TYPES:
[[eIT],]
//The following three elements appear in case of DN.
- 4:□ TIME-OF-DELIVERY:
[delivery-time]
- 45:□ TYPE-OF-UA:
[type-of-ua]
- 68:□ SUPPLEMENTARY-INFORMATION:
[suppl-info]
//The following two elements appear in case of NDN.
- 46:□ NON-DELIVERY-REASON:
[reason-code],[diagnostic-code]
]
- 72:□ CONTENT-RETURNED-INDICATION:

B.2.12 *Register-TAPDU*

- 3.11:□ REGISTER:
- 73:□ EXPIRED-DISCARD:□[discard-ipm]
- 74:□ OBSOLETE-DISCARD:□[discard-ipm]
- 76:□ AUTO-FWD-IPMS:□
[auto-fwd-ipms]
- 77:□ AUTO-FWD-RECIPIENTS:□
[[auto-fwd-recipient],]
- 78:□ AUTO-FWD-HEADING:□
[auto-fwd-heading]
- 79:□ AUTO-FWD-COMMENT:□
[auto-fwd-comment]
- 58:□ DS-MODE:□[ds-mode]
- 59:□ TLMAU-OPERATION:□[error-recovery-mode],[auto-acknowledgement]
- 60:□ AUTO-OUTPUT:
[frequency] 'NL'
[output-time]
- 81:□ MESSAGE-DELETE-MODE:□[message-delete-mode]

//and other components require further study.

B.2.13 *RegisterAck-TAPDU*

- 3.11:□ REGISTER-ACK:

B.2.14 *DSQuery-TAPDU*

- 3.7:□ DS-QUERY:

B.2.15 *DSReport-TAPDU*

- 3.8:□ DS-REPORT:

[//repeated for each message
- 38:□ RETRIEVAL-ID:
[retrieval-id]
- 52:□ MESSAGE-TYPE:□[message-type]
- 14:□ ORIGINATOR:
[originator-name]
- 13:□ PRIORITY:□[priority-ind]
- 37:□ MESSAGE-LENGTH:□[message-length]

]

B.2.16 *MessageDelete-TAPDU*

3.18:□ MESSAGE-DELETE:
82:□ MESSAGE-SELECTOR:
[[retrieval-id] 'NL']

B.2.17 *OutputRequest-TAPDU*

3.9:□ OUTPUT-REQUEST:
[//repeated for each retrieval identifier
38:□ RETRIEVAL-ID:
[retrieval-id]
80:□ DELETE-AFTER-OUTPUT:□[delete-after-output]
]

B.2.18 *OutputMessage-TAPDU*

3.10:□ OUTPUT-MESSAGE:
[//repeated for each message
62:□ QUANTITY-OF-DOCS:□[number-of-docs]
38:□ RETRIEVAL-ID:
[retrieval-id]
52:□ MESSAGE-TYPE:□[message-type]
4:□ TIME-OF-DELIVERY:
[delivery-time]
]

The remaining Components of this TAPDU are identical to the components in the Delivery. DeliveryStatusNotice and ReceiptStatusNotice-TAPDU. The actual components to be used depend upon the Message Type parameter value specified in the Message Type component.

B.2.19 *StatusQuery-TAPDU*

3.19:□ STATUS-QUERY:
14:□ ORIGINATOR:
[originator-name]
65:□ SUBMISSION-ID:
[submission-msg-id]
1:□ CORRELATION-INFORMATION:
[call-id]

B.2.20 *StatusReport-TAPDU*

3.20:□ STATUS-REPORT:

1:□ CORRELATION-INFORMATION:
[call-id]

84:□ TIME-OF-REPORT:
[report-time]

75:□ REPORTED-MESSAGE-ID:
[reported-message-id]
[//repeated for each message

83:□ MESSAGE-STATUS:
[status]

4:□ TIME-OF-DELIVERY:
[delivery-time]

45:□ TYPE-OF-UA:
[type-of-ua]

46:□ NON-DELIVERY-REASON:
[reason-code],[diagnostic-code]
]

ANNEXE C

(to Recommendation T.330)

Element ID list

This Annex provides a table which maps the coding of TAPDU and the corresponding element ID. The element ID list Table C-1/T.330 comprises:

- 1) The first column contains the element number and element name, as defined in § 11 of this Recommendation.
- 2) The second column contains the TAPDUId or ElementId name defined in § 10 of this Recommendation.
- 3) The third column contains the type of the TAPDU element, defined in § 10 of this Recommendation.
- 4) The fourth column contains the Remarks.

TABLE C-1/T.330

Element ID list

TAPDUId/element ID	Name	Type	Remarks
3.1: <input type="checkbox"/> SEND:	send	–	
3.2: <input type="checkbox"/> PROBE:	probe	–	
3.3: <input type="checkbox"/> DELIVER:	deliver	–	
3.4: <input type="checkbox"/> DELIVERY-STATUS-NOTICE:	deliveryStatusNotice	–	
3.5: <input type="checkbox"/> RECEIPT-STATUS-NOTICE:	receiptStatusNotice	–	
3.6: <input type="checkbox"/> EXPLICIT-RN:	explicitRN	–	
3.7: <input type="checkbox"/> DS-QUERY:	dsQuery	–	
3.8: <input type="checkbox"/> DS-REPORT:	dsReport	–	
3.9: <input type="checkbox"/> OUTPUT-REQUEST:	outputRequest	–	
3.10: <input type="checkbox"/> OUTPUT-MESSAGE:	outputMessage	–	
3.11: <input type="checkbox"/> REGISTER:	register	–	
3.12: <input type="checkbox"/> EXCEPTION:	exception	–	
3.13: <input type="checkbox"/> CANCEL:	cancel	–	
3.14: <input type="checkbox"/> SEND-ACK:	sendAck	–	
3.15: <input type="checkbox"/> PROBE-ACK:	probeAck	–	
3.16: <input type="checkbox"/> EXPLICIT-RN-ACK:	explicitRNAck	–	
3.17: <input type="checkbox"/> REGISTER-ACK:	registerAck	–	
3.18: <input type="checkbox"/> MESSAGE-DELETE:	messageDelete	–	
3.19: <input type="checkbox"/> STATUS-QUERY:	statusQuery	–	
3.20: <input type="checkbox"/> STATUS-REPORT:	statusReport	–	
1: <input type="checkbox"/> CORRELATION-INFORMATION:	correlationInfo	Primitive	
2: <input type="checkbox"/>			Reserved for CF
3: <input type="checkbox"/> REPORTED-RECIPIENT:	reportedRecipient	Primitive	
4: <input type="checkbox"/> TIME-OF-DELIVERY:	timeOfDelivery	Primitive	
5: <input type="checkbox"/>			Reserved for CF
6: <input type="checkbox"/>			Reserved for CF
7: <input type="checkbox"/>			Reserved for CF
8:			Reserved for CF
9: <input type="checkbox"/> ERRORS:	errors	Primitive	
10:			Reserved for CF
11:			Reserved for CF
12: <input type="checkbox"/> RECEIVED-TID			Reserved for CF
13: <input type="checkbox"/> PRIORITY:	priority	Primitive	
14: <input type="checkbox"/> ORIGINATOR:	originator	Constructor	
15: <input type="checkbox"/> RECIPIENTS:	recipients	Constructor	
16: <input type="checkbox"/> CONVERSION:	conversion	Primitive	
17: <input type="checkbox"/> CONTENT-INFO:	contentInfo	Constructor	
18: <input type="checkbox"/> CONTENT-INDICATOR:	contentIndicator	Constructor	
19: <input type="checkbox"/> FLAG:	per Message Indicator	Constructor	

TABLE C-1/T.330 (cont.)

TAPDUId/element ID	Name	Type	Remarks
20:☐ FROM:	from	Primitive	
21:☐ AUTHORIZING:	authorizing	Constructor	
22:☐ TO:	to	Constructor	
23:☐ CC:	cc	Constructor	
24:☐ BCC:	bcc	Constructor	
25:☐ REPLY:	reply	Constructor	
26:☐ SUBJECT:	subject	Primitive	
27:☐ THIS-IPM:	thisIPM	Primitive	
28:☐ RELATED-IPMS:	relatedIPMS	Constructor	
29:☐ OBSOLETES:	obsoletedIPMS	Constructor	
30:☐ REPLIED-TO-IPM:	repliedToIPM	Primitive	
31:☐ BODY-TYPE:	bodyType	Constructor	
32:☐ FORWARDED-INFO:	forwardedInfo	Constructor	For further study
33:☐ SUBMISSION-TIME:	submissionTime	Primitive	
34:☐ LATEST-DELIVERY:	date and time	Primitive	
35:☐ DELIVERY-ID:	deliveryId	Primitive	
36:☐ RETURN-ADDRESS:	return address	Primitive	
37:☐ MESSAGE-LENGTH:	messageLength	Primitive	
38:☐ RETRIEVAL-ID:	retrievalId	Primitive	
39:☐ THIS-RECIPIENT:	thisRecipient	Primitive	
40:☐ INTENDED-RECIPIENT:	intendedRecipient	Primitive	
41:☐ OTHER-RECIPIENTS:	otherRecipients	Constructor	
42:☐ CONVERSION-INDICATION:	conversionIndication	Constructor	
43:☐ NOTIFICATION-TYPE:	notificationType	Primitive	
44:☐ CONVERTED-INFORMATION-TYPES:	convertedInfoTypes	Primitive	
45:☐ TYPE-OF-UA:	typeOfUA	Primitive	
46:☐ NON-DELIVERY-REASON:	nonDeliveryReason	Primitive	
47:☐ TIME-OF-RECEIPT:	timeOfReceipt	Primitive	
48:☐ TYPE-OF-RECEIPT:	typeOfReceipt	Primitive	
49:☐ NON-RECEIPT-INFO:	nonReceiptInfo	Primitive	
50:☐ COMMENT:	comment	Primitive	
51:☐ MESSAGE-RETURNED-INDICATION:	messageReturnedInd	–	
52:☐ MESSAGE-TYPE:	messageType	Primitive	
53:☐ LANGUAGE:	languageInd	Primitive	
54:☐ REDIRECTED-FROM:	redirected from	Constructor	

TABLE C-1/T.330 (end)

TAPDUId/element ID	Name	Type	Remarks
55:□	–	–	Not used
56:□	–	–	Not used
57:□	–	–	Not used
58:□ DS-MODE:	dsMode	Primitive	
59:□ TLMAU-OPERATION:	tLMAUOperation	Constructor	
60:□ AUTO-OUTPUT:	autoOutput	Constructor	
61:□	–	–	not used
62:□ QUANTITY-OF-DOCS:	quantityOfDocs	Primitive	
63:			Not assigned
64:			Not assigned
65:□ SUBMISSION-ID:	submissionId	Primitive	
66:□ PROBE-ID:	probeId	Primitive	
67:□ MSG-INCOMPLETE:			
68:□ SUPPLEMENTARY-INFORMATION:	supplReceiptInfo SupplInfo	Primitive	
69:□ IPN-ORIGINATOR:	IPNOriginator	Primitive	
70:□ PREFERRED-RECIPIENT:	preferredRecipient	Primitive	
71:□ SUBJECT-IPM:	subjectIPM	Primitive	
72:□ CONTENT-RETURNED-INDICATION:	contentReturned	–	
73:□ EXPIRED-DISCARD:	expiredDiscard	Primitive	
74:□ OBSOLETE-DISCARD:	obsoletedDiscard	Primitive	
75:□ REPORTED-MESSAGE-ID:	reportedMessageId	Primitive	
76:□ AUTO-FWD-IPMS:	autoFWDIPMs	Primitive	
77:□ AUTO-FWD-RECIPIENTS:	autoFWDRecipient	Primitive	
78:□ AUTO-FWD-HEADING:	autoFWDHeading	Constructor	For further study
79:□ AUTO-FWD-COMMENT:	autoFWDComment	Primitive	
80:□ DELETE-AFTER-OUTPUT:	deleteAfterOutput	Primitive	
81:□ MESSAGE-DELETE-MODE:	messageDeleteMode	Primitive	
82:□ MESSAGE-SELECTOR:	messageSelector	Constructor	
83:□ MESSAGE-STATUS:	messageStatus	Primitive	
84:□ TIME-OF-REPORT:	timeOfReport	Primitive	

Conventions:

- 1) Primitive: element contains only one component
- 2) Constructor: element contains more than one component
- 3) □: space character

ANNEX D

(to Recommendation T.330)

Element of service for TTX/IPM service intercommunication

This PTXAU provides only the services listed in Table D-1/T.330 to telematic users. These services may be implemented using only the mhs-doc-xfer abstract operations: message send, message delivery, receipt status notice and delivery status notice.

When a non-registered user attempts to access a service not within the PTTXAU set of services, then a subscription error will be returned.

TABLE D-1/T.330

Element of service

Reference Rec. F.400 Annex B	F.400 Elements of service	Message submission from TTX to PTTXAU	Message delivery to TTX from PTTXAU	Information generated by PTTXAU
B.5	Authorizing users indication		X	
B.6	Auto-forwarded indication		X	
B.8	Blind copy recipient indication		X	
B.9	Body part encryption indication		X	
B.12	Content type indication		X	X
B.13	Conversion prohibition	X	X	
B.15	Converted indication		X	
B.18	Cross-referencing indication		X	
B.21	Delivery notification	X	NA	X
B.22	Delivery time stamp indication		X	X
B.25	Disclosure of other recipients	X	X	
B.26	DL expansion history indication		X	X
B.29	Expiry date indication		X	
B.31	Forwarded IP-message indication		X	
B.32	Grade of delivery selection	X	X	
B.34	Implicit conversion		NA	X
B.35	Importance indication		X	
B.37	IP-message identification		X	X
B.38	Language indication		X	
B.39	Latest delivery designation		NA	X
B.41	Message identification		X	
B.45	Multi-destination delivery	X	NA	
B.46	Multi-part body		X	
B.47	Non-delivery notification		NA	
B.48	Non-receipt notification request indication	X	NA	
B.52	Obsoleting indication		X	
B.54	Original encoded information types indication		X	X
B.55	Originator indication		X	
B.56	Originator requested alternate recipient		X	
B.62	Primary and copy recipients indication	X	X	
B.72	Reply request indication		X	
B.73	Replying IP-message indication		X	
B.80	Sensitivity indication		X	
B.88	Subject indication	X	X	
B.89	Submission time stamp indication		X	

NA Not applicable

X Applicable