INTERNATIONAL TELECOMMUNICATION UNION

CCITT

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

BLUE BOOK

VOLUME VII – FASCICLE VII.5

TERMINAL EQUIPMENT AND PROTOCOLS FOR TELEMATIC SERVICES

RECOMMENDATIONS T.65-T.101, T.150-T.390

IXTH PLENARY ASSEMBLY
MELBOURNE, 14-25 NOVEMBER 1988

Geneva 1989

INTERNATIONAL TELECOMMUNICATION UNION

CCITT

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

BLUE BOOK

VOLUME VII – FASCICLE VII.5

TERMINAL EQUIPMENT AND PROTOCOLS FOR TELEMATIC SERVICES

RECOMMENDATIONS T.65-T.101, T.150-T.390

IXTH PLENARY ASSEMBLY
MELBOURNE, 14-25 NOVEMBER 1988

Geneva 1989

ISBN 92-61-03631-7

CONTENTS OF THE CCITT BOOK
APPLICABLE AFTER THE NINTH PLENARY ASSEMBLY (1988)

BLUE BOOK

FASCICLE III.6 – Line transmission of non-telephone signals. Transmission of sound-programme and television signals. Series H and J Recommendations (Study Group XV).

FASCICLE III.7 – Integrated Services Digital Network (ISDN) – General structure and service capabilities. Recommendations I.110-I.257 (Study Group XVIII).

FASCICLE III.8 – Integrated Services Digital Network (ISDN) – Overall network aspects and functions, ISDN user-network interfaces. Recommendations I.310-I.470 (Study Group XVIII).

FASCICLE III.9 – Integrated Services Digital Network (ISDN) – Internetwork interfaces and maintenance principles. Recommendations I.500-I.605 (Study Group XVIII).

Volume IV

FASCICLE IV.1 – General maintenance principles: maintenance of international transmission systems and telephone circuits. Recommendations M.10-M.782 (Study Group IV).

FASCICLE IV.2 – Maintenance of international telegraph, phototelegraph and leased circuits. Maintenance of the international public telephone network. Maintenance of maritime satellite and data transmission systems. Recommendations M.800-M.1375 (Study Group IV).

FASCICLE IV.3 – Maintenance of international sound-programme and television transmission circuits. Series N Recommendations (Study Group IV).

FASCICLE IV.4 – Specifications for measuring equipment. Series O Recommendations (Study Group IV).

Volume V – Telephone transmission quality. Series P Recommendations (Study Group XII).

Volume VI

FASCICLE VI.1 – General Recommendations on telephone switching and signalling. Functions and information flows for services in the ISDN. Supplements. Recommendations Q.1-Q.118 *bis* (Study Group XI).

FASCICLE VI.2 – Specifications of Signalling Systems Nos. 4 and 5. Recommendations Q.120-Q.180 (Study Group XI).

FASCICLE VI.3 – Specifications of Signalling System No. 6. Recommendations Q.251-Q.300 (Study Group XI).

FASCICLE VI.4 – Specifications of Signalling Systems R1 and R2. Recommendations Q.310-Q.490 (Study Group XI).

FASCICLE VI.5 – Digital local, transit, combined and international exchanges in integrated digital networks and mixed analogue-digital networks. Supplements. Recommendations Q.500-Q.554 (Study Group XI).

FASCICLE VI.6 – Interworking of signalling systems. Recommendations Q.601-Q.699 (Study Group XI).

FASCICLE VI.7 – Specifications of Signalling System No. 7. Recommendations Q.700-Q.716 (Study Group XI).

FASCICLE VI.8 – Specifications of Signalling System No. 7. Recommendations Q.721-Q.766 (Study Group XI).

FASCICLE VI.9 – Specifications of Signalling System No. 7. Recommendations Q.771-Q.795 (Study Group XI).

FASCICLE VI.10 – Digital subscriber signalling system No. 1 (DSS 1), data link layer. Recommendations Q.920-Q.921 (Study Group XI).

FASCICLE VI.11 – Digital subscriber signalling system No. 1 (DSS 1), network layer, user-network management. Recommendations Q.930-Q.940 (Study Group XI).

FASCICLE VI.12 – Public land mobile network. Interworking with ISDN and PSTN. Recommendations Q.1000-Q.1032 (Study Group XI).

FASCICLE VI.13 – Public land mobile network. Mobile application part and interfaces. Recommendations Q.1051-Q.1063 (Study Group XI).

FASCICLE VI.14 – Interworking with satellite mobile systems. Recommendations Q.1100-Q.1152 (Study Group XI).

Volume VII

FASCICLE VII.1 – Telegraph transmission. Series R Recommendations. Telegraph services terminal equipment. Series S Recommendations (Study Group IX).

FASCICLE VII.2 – Telegraph switching. Series U Recommendations (Study Group IX).

FASCICLE VII.3 – Terminal equipment and protocols for telematic services. Recommendations T.0-T.63 (Study Group VIII).

FASCICLE VII.4 – Conformance testing procedures for the Teletex Recommendations. Recommendation T.64 (Study Group VIII).

FASCICLE VII.5 – Terminal equipment and protocols for telematic services. Recommendations T.65-T.101, T.150-T.390 (Study Group VIII).

FASCICLE VII.6 – Terminal equipment and protocols for telematic services. Recommendations T.400-T.418 (Study Group VIII).

FASCICLE VII.7 – Terminal equipment and protocols for telematic services. Recommendations T.431-T.564 (Study Group VIII).

Volume VIII

FASCICLE VIII.1 – Data communication over the telephone network. Series V Recommendations (Study Group XVII).

FASCICLE VIII.2 – Data communication networks: services and facilities, interfaces. Recommendations X.1-X.32 (Study Group VII).

FASCICLE VIII.3 – Data communication networks: transmission, signalling and switching, network aspects, maintenance and administrative arrangements. Recommendations X.40-X.181 (Study Group VII).

FASCICLE VIII.4 – Data communication networks: Open Systems Interconnection (OSI) – Model and notation, service definition. Recommendations X.200-X.219 (Study Group VII).

FASCICLE VIII.5 – Data communication networks: Open Systems Interconnection (OSI) – Protocol specifications, conformance testing. Recommendations X.220-X.290 (Study Group VII).

FASCICLE VIII.6 – Data communication networks: interworking between networks, mobile data transmission systems, internetwork management. Recommendations X.300-X.370 (Study Group VII).

FASCICLE VIII.7 – Data communication networks: message handling systems. Recommendations X.400-X.420 (Study Group VII).

FASCICLE VIII.8 – Data communication networks: directory. Recommendations X.500-X.521 (Study Group VII).

Volume IX – Protection against interference. Series K Recommendations (Study Group V). Construction, installation and protection of cable and other elements of outside plant. Series L Recommendations (Study Group VI).

Volume X

CONTENTS OF FASCICLE VII.5 OF THE BLUE BOOK

Recommmendations T.65-T.101, T.150-T.390

**Terminal equipment and protocols
for telematic services**

PRELIMINARY NOTES

1 The Questions entrusted to each Study Group for the Study Period 1989-1992 can be found in Contribution No. 1 to that Study Group.

2 In this fascicle, the expression "Administration" is used for shortness to indicate both a telecommunication Administration and a recognized private operating agency.

3 The status of annexes and appendices attached to the Series T Recommendations should be interpreted as follows:

- an *annex* to a Recommendation forms an integral part of the Recommendation;
- an *appendix* to a Recommendation does not form part of the Recommendation and only provides some complementary explanation or information.

FASCICLE VII.5

Recommendations T.65-T.101, T.150-T.390

TERMINAL EQUIPMENT AND PROTOCOLS FOR TELEMATIC SERVICES

PAGE INTENTIONALLY LEFT BLANK

PAGE LAISSEE EN BLANC INTENTIONNELLEMENT

APPLICABILITY OF TELEMATIC PROTOCOLS AND TERMINAL CHARACTERISTICS TO COMPUTERIZED COMMUNICATION TERMINALS (CCTs)

(Melbourne, 1988)

The CCITT,

considering

(a) that there is an increasingly growing base of computerized communication terminals, such as communicating personal computers;

(b) that Administrations will require provisions to enable these devices to access CCITT-defined services, such as telematic services;

(c) that communication of these devices with each other may use provisions specified for communication within telematic services;

(d) that such devices may, due to their adaptive nature, require, in some areas, different protocols and terminal characteristics than existing telematic terminals;

(e) that the various telematic services are defined in the F-Series of Recommendations;

(f) that the reference model for open systems interconnection is defined in the X-200-Series of Recommendations;

(g) that the various telematic protocols and terminal characteristics are defined in the T-Series of Recommendations;

(h) that there is a requirement to assess the applicability of the protocols and terminal characteristics defined in the CCITT telematic recommendations to computerized communication terminals;

unanimously declares the view

that the following technical provisions determine the applicability of protocols and terminal characteristics specified in CCITT Telematic Recommendations to Computerized Communication Terminals.

CONTENTS

1 Scope

1.1 This Recommendation addresses the applicability of the protocol and terminal characteristics specified in CCITT-defined Recommendations to Computerized Communication Terminals (CCTs). It should be observed that the "adaptive" (as opposed to dedicated) nature of CCTs calls for, in certain areas, more flexibility, but without undue degradation of capabilities. The issues of flexibility versus degradation of capabilities strongly influenced the proposals made in this Recommendation.

1.2 This Recommendation specifies how the various telematic Recommendations may be used, and any additional requirements, to enable computerized communication terminals to access the various telematic services. It is noted that while this Recommendation is applicable to CCTs only when accessing telematic services, consideration may be given to the use of the technical aspects of this Recommendation if CCTs communicate with each other utilizing the telematic protocols.

1.3 Section 2 describes the characteristics of computerized communication terminals. The remaining sections define how the relevant telematic Recommendations may be used to enable CCTs to access the telematic services.

1.4 Figure 1/T.65 shows various methods for CCTs to access the telematic services which are described in §§ 3 to 9.

Three methods are proposed:

i) access to and from a telematic service via service access facility (SAF) (see § 3.4.2, for example);

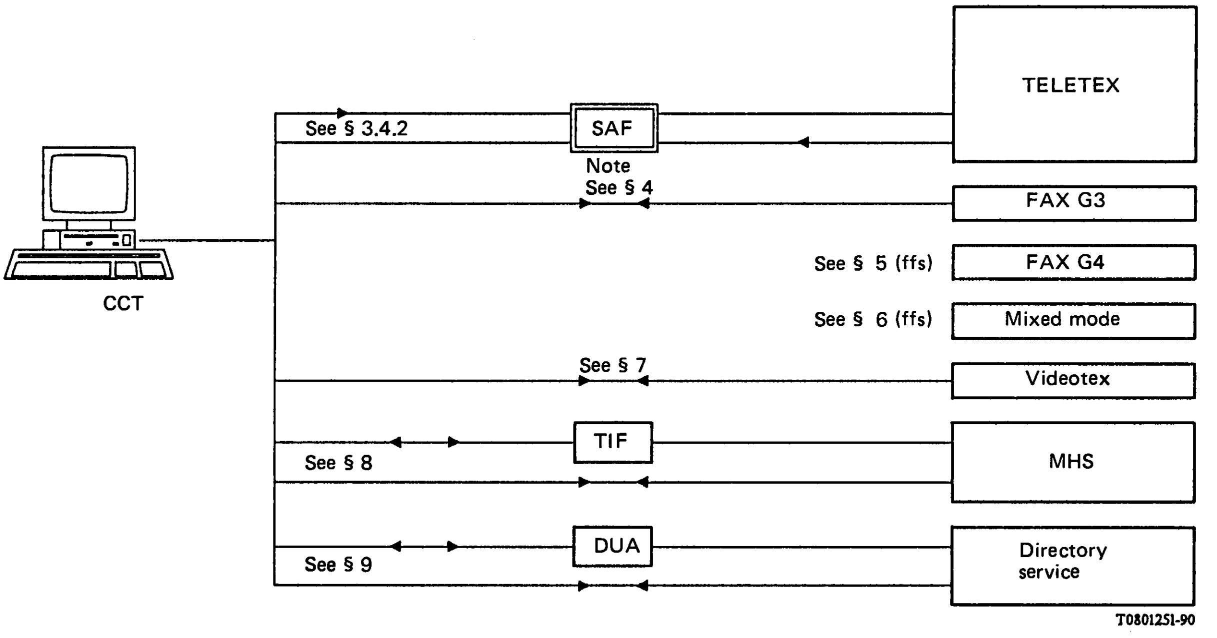
ii) direct access from and to a telematic service;

iii) direct access from a CCT to a telematic service, reverse access via SAF (see § 3.4.3, for example).

2 Characteristics and model

2.1 *Definition*

The term Computerized Communication Terminal (CCT) refers to a device or equipment, which may be portable, with a processor and communication facility, typically a user work station, which permits entry of various applications and which can access CCITT-defined services, such as telematics, as prescribed in this Recommendation.



CCT Computerized communication terminal
DUA Directory user agent
MHS Message handling system
SAF Service access facility
TIF Telematic interworking facility
ffs For further study

Note – As far as possible, the SAF functions should be realized using Message Handling Systems.

FIGURE 1/T.65

Various access methods

2.2 *Characteristics*

Computerized communication terminals differ in certain characteristics from telematic terminals. The following subsections identify the characteristics of CCTs. Characteristics specific to each case of the access to telematic services are given in §§ 3 to 9.

2.2.1 *Capability*

A CCT maybe used to access the telematic services. The provisions in this Recommendation provide a basic level of compatibility between CCTs and the telematic services.

2.2.2 *Protocols*

In general, CCTs will use OSI protocols defined in the X.200-Series of Recommendations, but configured to meet the requirements defined in the relevant T-Series of Recommendations. Exceptions include the cases of access to the non-OSI-based telematic services, where the relevant T-Series of Recommendations apply.

2.2.3 *Terminal requirements*

In general, the relevant T-Series Recommendations for terminal requirements apply. The details specified to each access to telematic services, and any additional (or relaxed) requirements are specified in §§ 3 to 9.

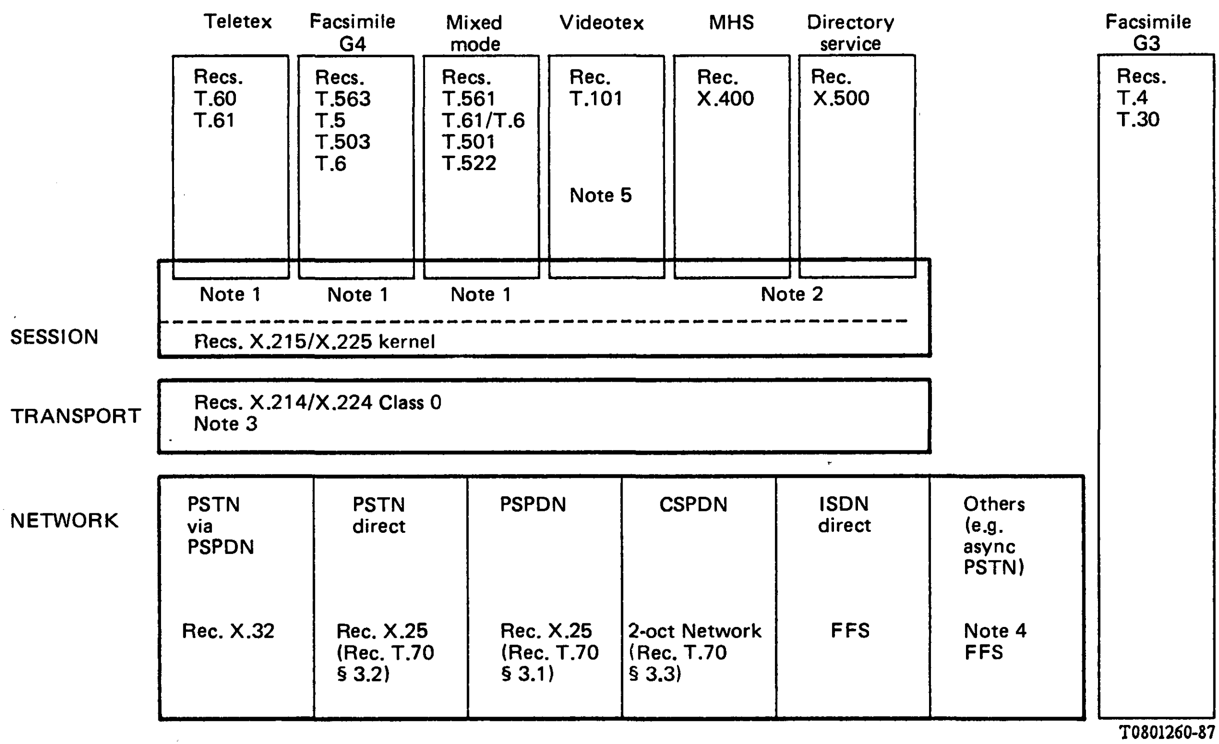
2.3 *General model*

A model for CCTs accessing telematic services based on OSI is given in Figure 2/T.65. The model identifies the relevant Recommendations applicable to each level in the OSI layers, for each case of access to telematic services. In particular, two sets of protocols are identified for access to OSI-based telematic services:

a) A set of OSI protocols common to most accesses to telematic services is identified for the lower layers up to and including the session kernel in the session layer. The corresponding CCITT Recommendations required are identified.

b) Above the common set of protocols, additional session layer functional units based on Recommendations X.215/X.225 are identified, together with any Recommendations required for each of the cases of the access to telematic services.

There are telematic services which require the use of non-OSI-based protocols. In these cases, the common set of protocols may not be applicable and the relevant T-Series Recommendations must be used.



FFS For further study

Note 1 – The functional units minor sync., half-duplex, capability data, activity management and exceptions; together with application rules provide the equivalent to Recommendation T.62.

Note 2 – For further study. But half-duplex and/or duplex functional units are required to conform with Recommendations X.215/225.

Note 3 – Class 0 is equivalent to § 5 of Recommendation T.70.

Note 4 – When ASYNC PAD access with error handling is defined by SG VII.

Note 5 – The full scenarios of network, transport and session layer capabilities for videotex access require further study.

FIGURE 2/T.65

A model for CCTs accessing telematic services

2.4 *Minimum capability*

For a CCT to access an OSI-based telematic service it must support all the following capabilities, and any additional capability required for each case of access to telematic service as prescribed in §§ 3, 5, 6, 7, 8 and 9:

a) The appropriate network capability as prescribed in § 3 of Recommendation T.70.

b) X.214/X.224 Class 0 Transport procedure.

c) X.215/X.225 Kernel; together with half-duplex, or full-duplex functional units.

Note – The applicability of the minimum capability to videotex access requires further study.

3 Access to the Teletex service

3.1 *General*

The access of CCTs to the Teletex service is a common case of communication with an OSI-based telematic service due to the well defined nature of Teletex. The following sections describe the characteristics of such an access and specify how the various Teletex-related Recommendations may be used.

3.2 *Characteristics*

3.2.1 From the technical point of view, CCTs will be able to establish communications directly with a Teletex device and exchange documents on a real-time, end-to-end basis without the use of conversion facilities.

3.2.2 As far as possible, CCT access to the Teletex service should be done via message handling systems. The technical implementation is a national matter.

3.2.3 CCTs may not necessarily be available continuously to receive incoming calls. However, when a CCT is available it will be technically able to receive calls directly from and exchange documents with other Teletex devices.

3.2.4 CCTs may technically use the Teletex protocol and terminal characteristics as prescribed in § 3.3 of this Recommendation to exchange Teletex documents with each other.

3.2.5 If a Teletex device communicates with a CCT, it must be made aware of that fact. How this information is conveyed within the Teletex terminal identification with a specific value for Part 3 is described in § 3.4.

3.3 *Applicability of the relevant CCITT Recommendations*

3.3.1 *Protocols*

a) The network capabilities are in accordance with § 3 of Recommendation T.70.

b) The transport procedure is in accordance with either:

   – Class 0 of the OSI transport protocol, as specified in Recommendations X.214/X.224, together with application rules to be compatible with and conform to the § 5 and Annexes of Recommendation T.70; or

   – Paragraph 5 and annexes of Recommendation T.70.

c) The session layer procedure is in accordance with either:

   – Kernel with the functional units minor sync, half-duplex, capability data, activity management, and exceptions specified in Recommendations X.215/X.225 together with application rules to be compatible with and conform to Recommendation T.62; or

   – Recommendation T.62.

d) The applicability of higher-layer Recommendations, such as T.300 and T.400, requires further study.

3.3.2 *Terminal requirements and character repertoire*

The terminal requirements and character repertoire specified in Recommendations T.60 and T.61 will apply except for the following:

a) A CCT may or may not support full automatic operation.

b) A CCT must be able to receive and store all characters belonging to the basic Teletex character repertoire. However, only those graphic characters which form the primary character set of the basic Teletex character set as defined in Recommendation T.61 need to be presented.

c) A CCT may require a different terminal identification from that of a Teletex terminal. The format of this identification is defined in § 3.4.3.1.

d) Other items require further study.

3.4 *Access methods*

3.4.1 *Introduction*

This paragraph describes a technical method for CCT access to and from the Teletex service. This access method is based on the assumption that CCTs should enjoy a maximum flexibility and that the service characteristics of Teletex should not be degraded.

These prerequisites imply that the CCT must be supported by a service access facility (SAF) which emulates the Teletex service characteristics and provide for the handling of messages.

3.4.2 *Description of the access method*

A CCT may establish a connection to the SAF at any time, from any network and from any access point within these networks. If a CCT wants to transmit a message but does not wish to receive a message, it need not be identified. The message will be received by the SAF and forwarded immediately to the Teletex destination. The SAF must add information which will indicate to the Teletex destination that this message was originated by an unidentified CCT.

If a CCT is to receive an answer to its previously transmitted message, it should be able to register itself temporarily using a password. The password will be provided by the CCT user. The message from the CCT will be forwarded immediately to the Teletex destination including information that the answer may be placed in the SAF under the given password. Provisions to allow positive or negative acknowledgements to the Teletex source and to allow control of the status of messages sent by the Teletex source are technically feasible.

In the following, the functions of the SAF are described which are needed to support CCTs for access to/from the Teletex service.

3.4.3 *Model* (see Figure 3/T.65)

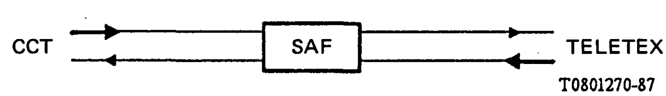


FIGURE 3/T.65

Access to/from Teletex via SAF

3.4.3.1 *CCT to Teletex*

The following functions will be provided by the SAF in order to enable a CCT to access the Teletex service:

a) insertion of an appropriate information from which the Teletex subscribes can identify that the message is being sent from a CCT (e.g., the letters "CCT" into Part 3 of the Teletex-TID);

b) temporary registration on an optional basis (to allow messages to be sent back to the CCT by a Teletex terminal, see § 3.4.3.2).

3.4.3.2 *Teletex to CCT*

The following functions will be provided by the SAF in order to enable a Teletex terminal to send documents to a CCT:

a) memory for storing messages sent by the Teletex terminal;

b) allocation of stored messages to registration numbers to allow their retrieval by the CCT;

c) means for a delivery notification call to the Teletex terminal to indicate that the CCT has retrieved the message;

d) a time-out mechanism for deleting a message if not retrieved within a certain period of time;

e) additional notification calls (e.g., status of stored messages) are for further study.

4 Access to the Group 3 facsimile service

4.1 *General*

A CCT may be used to access the Group 3 facsimile service.

4.2 *Characteristics*

A CCT accessing the Group 3 facsimile service will operate in accordance with the CCITT Recommendations T.4 and T.30.

4.3 *Applicability of the relevant CCITT Recommendations*

4.3.1 *Protocols*

The requirements defined in the CCITT Recommendation T.30 apply.

4.3.2 *Modulation systems*

The requirements defined in the CCITT Recommendation T.4 apply.

5 Access to the Group 4 facsimile service

(For further study.)

6 Access to the mixed-mode option of the Teletex service

(For further study.)

7 Access to the videotex service

7.1 *General*

A CCT may be used to access the videotex service. Since a videotex service will not distinguish between what type of terminal is connected to it, there are no special requirements for CCTs above those which apply to dedicated videotex terminals.

7.2 *Characteristics*

7.2.1 CCTs accessing the videotex service should emulate videotex terminal characteristics. In the emulation, attention should be given to the profiles, ranks or service reference modes of the videotex terminals concerned as used in the various videotex services. Where insufficient display capabilities are available, a CCT should provide fall-back by graceful degradation of capabilities so that the integrity of the information content is preserved. For example, a wide range of colours may fall back to fewer related colours, or to grey scales, or an accented character may fall back to a character without accent.

7.2.2 Videotex services are interactive and CCTs should be able to transmit and receive data interactively.

7.3 *Applicability of the relevant CCITT Recommendations*

7.3.1 *Protocols*

To be defined.

7.3.2 *Data syntax and terminal requirements*

The requirements defined in CCITT Recommendation T.101 (Annexes B, C and D) apply.

8 Access to MHS

8.1 *General*

This paragraph describes the characteristics of CCTs to access MHS and specifies how the various related Recommendations may be used.

8.2 *Characteristics*

In its present form, the message handling system has as its fundamental component the message transfer system (MTS), which comprises a number of message transfer agents (MTAs). A CCT can then access MHS in two ways as described in Figure 4/T.65 and the text below.

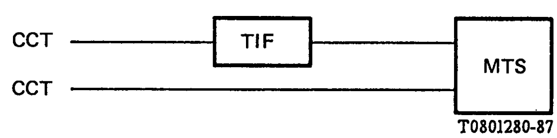


FIGURE 4/T.65

Access paths to MHS

i) The CCT can access MHS through a telematic interworking facility (TIF) as defined in Recommendations T.300-Series.

ii) The CCT can support the MHS user agent functions to access the MTS directly.

8.3 *Applicability of the relevant CCITT Recommendations*

When a CCT does not support the MHS user agent functions it shall access MHS through a TIF, which provides for interworking between Telematic services and MHS. In this case the relevant sections of Recommendations T.300-Series and T.65 apply, depending on the choice of protocols and terminal characteristics.

When a CCT supports the MHS user agent functions in addition to the Telematic protocols and terminal characteristics, it will use the relevant sections in the series of Recommendations X.400.

9 Access to the directory service

9.1 *General*

The access of CCTs to the directory service will often precede the other CCITT-defined services such as MHS, Teletex, or telephony, in order to determine or ascertain the address of a user or service. This section describes the characteristics of such an access and specifies how the various related Recommendations may be used.

9.2 *Characteristics*

In its present form, the directory system has two fundamental components: the directory user agent (DUA) and the directory (see Figure 5/T.65).

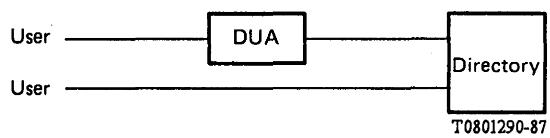


FIGURE 5/T.65

Access paths to the directory service

In terms of this model two ways of CCT access are possible:

i) The CCT can access the DUA using suitable telematic protocols and terminal characteristics defined in the T-Series of Recommendations.

ii) The CCT can support DUA functions to access the directory directly.

It should be noted that directory access is essentially an interactive application. Therefore, this interactive nature influences the protocol and terminal requirements.

9.3 *Applicability of the relevant CCITT Recommendations*

When a CCT does not support DUA functions, it shall access the directory through a DUA. In this case the relevant sections of the Recommendations X.500 and T.65 apply, depending on the choice of protocols and terminal characteristics.

When a CCT supports DUA functions in addition to the Telematic protocols and terminal characteristics it will use the relevant sections in the series of Recommendations X.500.

Recommendation T.70

NETWORK-INDEPENDENT BASIC TRANSPORT SERVICE FOR THE TELEMATIC SERVICES

(Geneva, 1980, amended at Malaga-Torremolinos, 1984, and Melbourne, 1988)

The CCITT,

considering

(a) that the Teletex service will be introduced in different types of network, i.e. circuit-switched public data networks (CSPDN), packet-switched public data networks (PSPDN) and the public switched telephone network (PSTN);

(b) that there is a need for international interworking between terminals belonging to the same or different types of Telematic services;

unanimously declares the following view

1 Scope

1.1 This Recommendation defines the *network-independent basic transport service* applicable to Teletex and Group 4 facsimile terminals connected to the types of network mentioned in (a) above in terms of:

a) the transport services provided to the higher layer [the transport services are provided by the transport layer (layer 4) in association with the underlying services provided by the supporting layers 1 to 3];

b) the transport layer procedure (see § 5 below).

1.2 Paragraph 2 describes the transport service. Paragraph 3 describes the transport service implementation for different types of networks. Paragraph 4 outlines the guidelines for interworking between networks. Paragraph 5 specifies the transport layer procedure, and Annexes A and B provide associated state transition diagrams and tables respectively.

2 Transport service

2.1 *Transport service objectives*

2.1.1 The purpose of the transport service is to provide two communicating session entities within two terminals with transport services, i.e. the means for transparent and reliable end-to-end transfer of data between them irrespective of the particular type of network used.

2.1.2 The main requirements of the transport service to be provided by a transport entity to the local transport user, i.e. the session entity, are:

a) *Network independence.* The transport service shall be homogeneous, while allowing a suitable wide variety of underlying communications media, protocols and mechanisms.

b) *End-to-end significance.* The transport service shall have end-to-end significance, connecting the end users irrespective of the number of individual communication links used.

c) *Transparency.* The transport service shall be octet transparent, i.e. not restrict the content, format or coding of the information (data or control) received from or delivered to the transport user.

d) *Error-free delivery.* The transport service shall assure error-free delivery. Non-recoverable errors are to be visible to the transport service user.

e) *Cost efficiency.* The transport service shall optimize the use of the available communication resources to provide the performance required by each communicating transport user at maximum efficiency.

2.2 *General structure of the transport service*

2.2.1 The general structure of the transport service is shown in Figure 1/T.70.

3 Transport service implementation for different types of networks

Note – The transport layer procedure on all types of networks is defined in § 5. The network dependent control procedures of the underlying layers are described in the following.

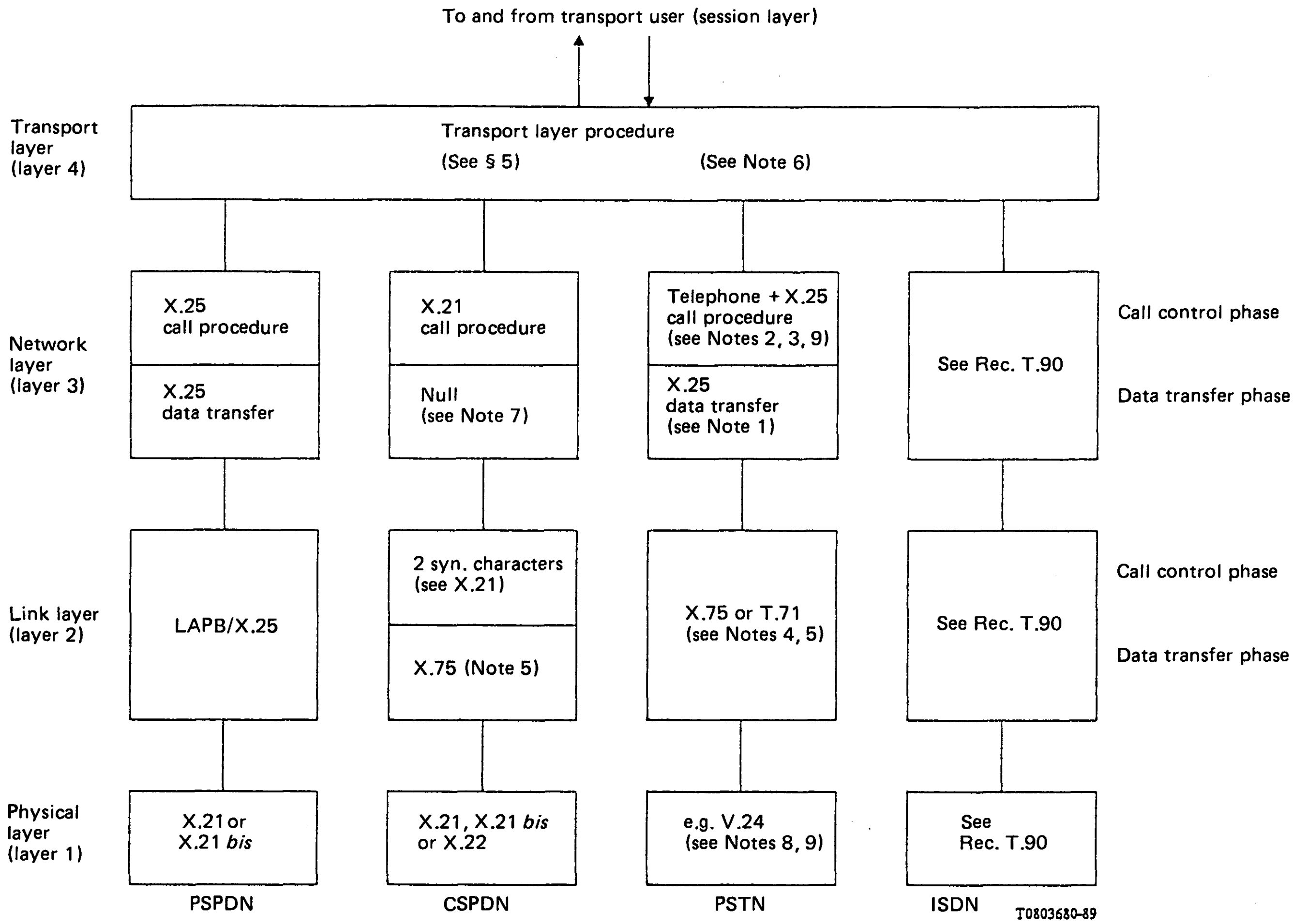
3.1 *Terminals connected to a PSPDN*

3.1.1 *Physical layer DTE/DCE interface characteristics*

The physical layer of Recommendation X.25 applies.

3.1.2 *Link layer procedure*

The link layer procedure shall, unless otherwise specified, be the symmetrical procedures as specified in Recommendation X.25, LAPB (Link Access Procedure B).



Note 1 – The X.25 network layer procedure is introduced to ease interworking with PSPDNs.

Note 2 – The establishing of the network connection is performed by two-stage selection: the first using normal telephone procedures and the second using X.25 call control procedures.

Note 3 – For terminals connected to PSTN accessing PSPDN, the procedures in Note 2 apply. See also Recommendation X.32.

Note 4 – T.71 defines a half-duplex link access procedure, based on Recommendation X.75 for single link operation (see § 3.2.2).

Note 5 – The link layer procedures are in accordance with Recommendation X.75 for single link operation. Additional application rules can be found in §§ 3.2.2 and 3.3.2.

Note 6 – In all cases of interworking including interworking between terminals connected to the same type of network or to different types of networks (i.e. CSPDN, PSPDN, PSTN), this transport layer procedure is executed peer-to-peer between the communicating terminals.

Note 7 – For terminals connected to CSPDNs, no function is needed in the network layer in the data transfer phase as indicated in this figure. However, in order to facilitate interworking with PSPDNs a minimum network layer is introduced (see § 3.3.3).

Note 8 – The modem may also be integrated within the terminal and in such cases Recommendation V.24 need not apply (see § 3.2.1).

Note 9 – For automatic calling and/or answering. Recommendation V.25 may be applicable.

FIGURE 1/T.70

Transport service general structure

3.1.3 *Network layer procedure*

Recommendation X.25 Virtual Call procedures apply. However the following points should be noted when using this transport protocol:

a) The qualifier bit in data packets should always be set to 0.

b) The delivery confirmation bits in all packets should be set to 0.

c) The terminal should not send an *interrupt request* packet.

d) Normal X.25 reset procedures will apply.

e) Each control block or data block of the transport layer shall be carried in a complete data packet sequence.

f) The terminal should not send a *DTE REJ packet.*

g) Terminals shall use a specific protocol identifier within call request/incoming call packets for the Teletex service and Group 4 facsimile apparatus. This identifier is represented by the first octet of the call user data field (remaining octets, if any, should be ignored) as shown below:

   bit 87654321

   octet 1 00000010

   In the case of CSPDN/PSPDN interworking the functional mapping of this protocol identifier requires further study.

h) Terminals shall not use the fast select facility.

3.2 *Terminals connected to the PSTN*

3.2.1 *Physical layer DTE/DCE interface characteristics*

The DTE/DCE physical layer element shall be in accordance with existing Series V Recommendations. The physical layer may provide for half-duplex or full-duplex transmission depending on the modem standard.

Note – The PSTN modem standards are discussed in Study Group XVII. Furthermore, in the case of a modem integrated in the terminal, the interface may only be functionally equivalent to a Series V Recommendation. This is also for further consideration in Study Group XVII.

3.2.2 *Link layer procedure*

3.2.2.1 Depending on the service provided by the physical layer, the link layer procedures over a single physical circuit between two terminals have to cater for a half-duplex or full-duplex transmission facility to provide a full-duplex service to the network layer. For full-duplex physical layer service, the link layer procedure shall conform to the Link Access Procedure described in Recommendation X.75, for single link operation. For addressing assignments and the system parameters see §§ 3.2.2.2 and 3.2.2.3 respectively. For half-duplex physical layer service the link layer procedure is as defined in Recommendation T.71. This is a half-duplex Link Access Procedure, based on Recommendation X.75 for single link operation.

3.2.2.2 The following describes the application of the link addressing procedure of Recommendation X.75. Link addresses (A and B) shall be assigned dynamically or on a per-call basis according to the following rules:

a) the calling terminal shall take Address A;

b) the called terminal shall take Address B;

c) commands and responses shall be transferred as shown in Figure 2/T.70;

d) A and B addresses are coded as follows:

Address	12345678
A	11000000
B	10000000

Note – The terminal will discard all frames received with an address other than A and B.

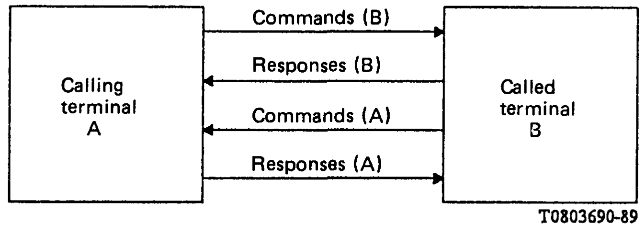


FIGURE 2/T.70

3.2.2.3 System parameters are:

a) timer, T1;

b) maximum number of retransmissions, N2;

c) maximum number of bits in an I frame, N1;

d) maximum number of outstanding I frames, *k*.

The above system parameters are to be specified by the Administration. However, the possible range of values that may be attributed to each parameter is to be standardized. Such values are for further study.

3.2.3 *Network layer procedure*

3.2.3.1 See § 3.1.3. In addition, for all calls (PSTN only, PSTN-PSPDN, PSTN-PSPDN-PSTN) second stage addressing will apply using X.25 virtual call procedures. The calling terminal should include the called address and the calling address (see Note 2) in call request packets. The format of the called address shall conform to:

a) the telephone network addressing scheme for PSTN only calls;

b) the telephone network addressing scheme with an X.121 DNIC for PSTN-PSPDN calls (see Note 3);

c) the X.121 addressing scheme for PSTN-PSPDN calls (see Note 1).

Note 1 – For other cases of internetworking the above rule shall apply.

Note 2 – In the case of PSTN-PSPDN calls the verification of the calling address by the network requires further study. The format of the calling address is for further study.

Note 3 – The feasibility of such connections is for further study.

3.3 *Terminals connected to a CSPDN*

3.3.1 *Physical layer DTE/DCE interface characteristics*

The DTE/DCE physical interface characteristics shall be in accordance with Recommendation X.21, or as an option, Recommendation X.22 for multi-call operation.

3.3.2 *Link layer procedure*

3.3.2.1 *General*

The link layer procedure shall be used during the data phase of Recommendation X.21 (or X.22) for data interchange over a single physical circuit between two terminals operating in User Classes of Services 3 to 7 and 30 as defined in Recommendation X.1. The link layer procedure shall consist of a fully symmetrical HDLC procedure as defined in Recommendation X.75 for single link operation.

3.3.2.2 *Link layer address procedure*

The following describes the application of the link addressing procedures of Recommendation X.75. Link addresses (A and B) shall be assigned dynamically on a per-call basis according to the following rules:

a) the calling terminal shall take address A;

b) the called terminal shall take address B;

c) commands and responses shall be transferred as shown in Figure 3/T.70;

d) A and B addresses are coded as follows:

Address	12345678
A	11000000
B	10000000

Note – The terminal will discard all frames received with an address other than A and B.

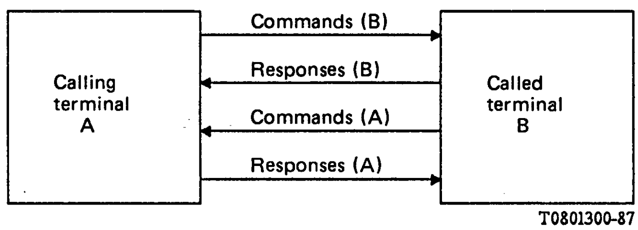


FIGURE 3/T.70

3.3.2.3 *Link layer implementation rules*

In order to achieve full compatibility between different implementations, the rules below for the implementation of Recommendation X.75 shall be followed.

3.3.2.3.1 *General rules*

a) The 1984 version (*Red Book*) of CCITT Recommendation X.75, § 2, shall be used as the reference specification.

b) The term "STE" shall be read as "DTE".

c) The Non-Extended mode of operation (modulo 8) shall be used.

d) Only the single link procedure (SLP) shall be used.

3.3.2.3.2 *Specific rules*

The following rules refer to the indicated sections and tables of Recommendation X.75.

a) *Table 1/X.75* (see Note 1)

I-frame must not be sent with an empty I-field.
$N \geqslant 0$
$N \leqslant N1 - 32$
A received empty I-frame shall be treated as a valid I-frame.

b) *§ 2.3.4.9*

Subparagraphs 5), 6) and 7) are not valid (shall not result in the sending of a FRMR). Instead the following actions shall be implemented:
– Not expected supervisory frames with the F bit set to 1 shall be ignored.
– Not expected UA or DM response shall be ignored.
– Frames with an invalid N(S) shall be responded to by sending REJECT.

Frames with a FRMR cntrol field shall not be responded by sending a FRMR.

c) *Table 7/X.75*

Bits W, X, Y and Z set to 0 indicate that no reason for frame rejection is given.

d) *§ 2.3.5.3*

The DTE and the CSPDN are not octet aligned and the last paragraph is therefore not valid.

e) *§ 2.3.5.5*

Higher layers should be notified when timer T3 expires (excessive idle state).

f) *§ 2.4.3*

Related to the first paragraph, read instead of "next response" "corresponding response".

g) *§ 2.4.4.1*

In the active channel state, the DTE shall transmit contiguous flags independent of the other DTE.

The calling DTE shall initialize the link by sending a SABM command with the P bit set to 1.

h) *§ 2.4.4.4.1*

A condition for entering the disconnected phase is also that no unacknowledged DISC command exists, because of collision cases (Ref. X.75, § 2.4.4.5).

In the disconnected phase, it is the calling DTE which may initiate link set up.

i) *§ 2.4.5.9, fourth paragraph*

If a RNR is received, the DTE shall remain in the timer recovery condition (because the other DTE is still in the busy condition).

j) *§ 2.4.5.9, fifth paragraph*

If a RNR is received, the DTE shall not resume I-frame transmission or retransmission.

k) *§ 2.4.5.9, last paragraph*

If the transmission attempt variable is equal to N2, the DTE shall enter the disconnected phase.

l) *§ 2.4.7.3*

In the frame rejection condition, the DTE shall only check the commands and react with a FRMR according to the P bit.

The frame rejection condition is cleared when the DTE receives a SABM, or, receives or transmits a DISC command.

m) *§ 2.4.7.3, second paragraph* (see Note 2)

Only the DTE which caused the FRMR condition may try to reset the link.

n) *§ 2.4.7.3, third paragraph* (see Note 3)

After N2 attempts to get the other DTE to reset the link, the DTE shall enter the disconnected phase.

o) *§ 2.4.8.1* (see Note 4)

The timer T1 shall be started at the end of frame transmission. The value of T1 depends on the data signalling rate, the frame length, the value of N2 and a fixed time of 1.5 s representing both T2 and the transmission delay [see § 3.3.2.3.2 r)]. The recommended value range is: 1.5-15 s.

p) *§ 2.4.8.2* (see Note 4)

$T1 > T2$
$T2 \leqslant 1$ s
Depending on the acknowledgement strategy used, the DTE designer may regard T2 as a decision parameter only, in which case the DTE is not obliged to implement a corresponding timer.

q) *§ 2.4.8.3, second paragraph*

$30 \text{ s} \leqslant T3 \leqslant 60 \text{ s}$

r) *§ 2.4.8.4*

$N2 \times T1 \geqslant 60$ s

s) *§ 2.4.8.5*

$N1 = 1080 + (n \times 1024)$ bits; $n = 0$ or 1 or 3 or 7 or 15.

t) *§ 2.4.8.6* (see Note 4)

$k = 2 - 7$ (modulo 8)

Note 1 – Terminals complying with the *Red Book* version of Recommendation T.70 may react by DL Reset ind. (FRMR).

Note 2 – Terminals complying with the *Red Book* version of Recommendation T.70 may react differently.

Note 3 – It is not meaningful to reset the link if the other DTE is not responding for N2 × T1.

Note 4 – The acknowledgement strategy used by the receiving DTE should be independent of any knowledge about the value of *k* used by the sending DTE. This can be achieved by either acknowledging every correctly received I-frame as soon as possible or by implementing an acknowledgement timer, i.e., a T2 timer as defined above [see § 3.3.2.3.2 p)].

3.3.3 *Network layer procedure*

3.3.3.1 *Call control phase*

The call control procedure conforms to Recommendation X.21, or as an option, Recommendation X.22 for multi-call operation.

3.3.3.2 *Data transfer phase*

A minimal network layer is present during the data transfer phase and accommodated through the use of a two-octet network block header. The header comprises a one-octet length indicator followed by a network block type code specified below. The only network block currently defined is a network protocol data block as shown in Figure 4/T.70.

Octets \ Bits	8	7	6	5	4	3	2	1
1	0	0	0	0	0	0	0	1[a)]
2	M[b)]	Q[c)]	0[d)]	0[e)]	0[f)]	0[f)]	0[f)]	0[f)]
3 to n	Network user data field							

a) The length indicator expresses in octets the length of the network data block header. This length does not include octet 1.

b) The more data mark (M) is used to preserve the integrity of transport layer control and transport data blocks. When M is set to 1 it indicates that more data is to follow. A terminal has to accept as much bytes as the terminal can support by block size negotiation. Receiving more bytes the terminal may react by N DISC indication.

c) The qualifier bit (Q) is introduced to provide a functional mapping with the X.25 qualifier bit for CSPDN/PSPDN interworking. If the Q bit is not used it shall be set to zero.

d) and e) are spare bits for possible new single bit functions.

f) The bits 1 to 4 with the code four zeros (0000) are used to identify the network data block. Other network layer protocol unit types (i.e. control blocks used in the data transfer phase) may be defined in the future.
The network user data field is delimited by the HDLC Closing Flag at the link layer. It must contain at least one octet.

FIGURE 4/T.70

Network data block

3.3.3.3 *Data transfer procedure*

3.3.3.3.1 *Handling of the M-bit*

The calling DTE shall negotiate the TPDU size with the called DTE at the transport layer, based on either the maximum TPDU size supported or the optimum TPDU size for the specific call, unless the default value of 128 octets is used. The agreed value will allow the sending DTE to transfer TPDUs without the need for segmenting at the Network layer and consequently the M-bit is set to zero.

However, receiving DTEs must always be capable of reassembling segmented TPDUs by using the M-bit, since segmenting may take place in the network in some interworking situations, e.g., when the composite network connection comprises a PSDN.

3.3.3.3.2 *Error procedures*

A Data PDU with a length indicator different from hexadecimal "01" and/or with less than three octets shall be discarded and the physical network connection shall be cleared.

3.4 *Terminals connected to an ISDN*

See Recommendation T.90.

4 Interworking between networks

4.1 It is the responsibility of Administrations to decide in which network(s) the telematic services are to be provided.

4.2 Four possibilities are considered below:

a) Terminals connected to a circuit switched public data network (CSPDN);

b) Terminals connected to a packet switched public data network (PSPDN);

c) Terminals connected to a public switched telephone network (PSTN);

d) Terminals connected to an integrated services digital network (ISDN).

4.3 Interworking between telematic terminals connected to any network must be possible.

4.4 International interworking between telematic terminals shall preferably take place between networks of the same type when these networks are provided by both countries involved.

4.5 In the case of international interworking between telematic terminals connected to dissimilar networks, Recommendation X.300 shall apply.

The interworking between CSPDNs and PSPDNs is described in Recommendation X.82 (Detailed arrangements for interworking between CSPDNs and PSPDNs based on Recommendation T.70).

5 Transport layer procedure

5.1 *Transport functions*

5.1.1 *General*

5.1.1.1 The transport layer will perform all those functions that are necessary to bridge the gap between the services provided by the network layer and the services needed by the session layer. Therefore, the functions performed are dependent on two criteria: the services provided by the underlying network layer and the services required by the session layer.

5.1.1.2 It is the responsibility of the transport service user to select a given quality of service, which may imply the use of certain transport layer functions such as:

a) establishment of a transport connection
   - transport connection identification
   - transport connection multiplexing;

b) data transfer
   - sequence control
   - error detection
   - error recovery
   - segmenting and reassembling
   - flow control
   - purge;

c) termination of a transport connection.

Note – Not all of the above functions will be available in the basic transport service (see § 5.1.3).

5.1.2 *Transport protocol classes*

5.1.2.1 Transport layer functions are grouped (for ease of negotiation) into a hierarchical system of transport protocol classes whereby classes occupying superior positions in the hierarchy implement functions of the lower classes together with the optional functions identified for their own class.

5.1.2.2 During transport connection establishment the use of a given transport protocol and optional functions should be negotiated according to the following rules:

- the calling terminal indicates the transport protocol class and (if applicable) the optional functions required;
- the called terminal indicates the transport protocol class and (if applicable) the optional functions that it is willing to support;
- all parameters to be used in the transport connection must be explicity indicated, otherwise default values will apply.

5.1.2.3 The basic transport service described here is fulfilled by a protocol denoted in Recommendation X.224 as transport protocol class 0. That protocol class is compatible with Recommendation T.70. In the event of a discrepancy between transport protocol class 0 as described in Recommendation X.224 and Recommendation T.70, the latter takes precedence.

5.1.3 *The basic transport service (TS)*

5.1.3.1 A limited set of transport layer functions is defined for a basic transport service. The basic transport service is provided by transport layer functions which are performed by *transport layer protocol elements.*

5.1.3.2 Transport protocol data units (TPDUs) carrying transport service (TS) user information or control information are called *blocks.*

5.1.3.3 Transport layer block types are as follows:

a) transport connection request (TCR) block;

b) transport connection accept (TCA) block;

c) transport connection clear (TCC) block;

d) transport data (TDT) block;

e) transport block reject (TBR) block.

5.1.3.4 The TCR and TCA blocks are used to indicate the protocol class, and optional functions, applying to a transport connection. The TCC block is used to indicate the reason for refusing a connection establishment. The TDT block carries information of the transport service user. The TBR block is used to report procedure errors to the remote terminal.

5.1.4 *Transport layer functions*

5.1.4.1 Basic class functions and associated transport layer protocol elements, i.e. blocks, include:

a) transport connection establishment, transport connection identification, optional extended addressing and optional transport data block size negotiation (TCR, TCA and TCC blocks);

b) data delimitation, segmentation/reassembling of arbitrarily long transport service data units (TSDU). These are contained within TDT blocks. The end of a TSDU is indicated by a TSDU end mark in the last data block;

c) detection and indication of procedural errors (TBR block).

5.1.4.2 Other characteristics of the basic transport service are:

a) maintenance of TSDU integrity;

b) overflow: if the user cannot absorb new data and if the appropriate buffers are not available, flow control is performed at the network/link layer as appropriate;

c) error: no mechanism is provided within the transport layer to facilitate recovery from detected errors. Where such errors are detected the user of the transport service should be informed so that appropriate recovery action may be taken.

5.2 *Description of connection establishment and termination procedures*

5.2.1 *General*

5.2.1.1 The transport layer connection establishment and termination procedures shall also be used for negotiating transport protocol class and, if applicable, optional transport connection functions.

5.2.1.2 For the basic transport service, means are provided to establish a transport connection using a TCR block and a TCA block. This exchange provides:

a) a way to negotiate options;

b) a transport connection identification. The transport connection is identified by use of cross-references. Each end of the connection is responsible for selecting a suitable transport connection identifier.

5.2.1.3 This mechanism also provides an identification of the transport connection independent of any network connection identification and therefore provides independence from the life of the network connection. The binary value 0 should not be used as an identifier. The use of such references for reconnection requires further definition.

5.2.2 *Transport connection request (TCR) block*

5.2.2.1 The calling terminal shall indicate a transport connection request by transferring a TCR block to the remote terminal. The TCR block includes the transport functions (e.g. source reference, class, and optional functions) for negotiation of the characteristics of the transport connection being established.

5.2.3 *Transport connection accept (TCA) block*

5.2.3.1 The called terminal shall indicate its acceptance of the transport connection by transferring a TCA block to the remote terminal. The TCA block includes the transport parameters applying to the connection and to be used by the calling terminal.

5.2.3.2 If a terminal receives the request for an optional TDT block size it may either:

- indicate its support by reproducing the requested value in the TCA block;
- request in the TCA block the use of a shorter allowable TDT block. The calling side either accepts this size by sending the first TDT block or disconnects the network connection;
- not accept the requested TDT block size parameter value by sending a TCA block without a TDT block size parameter. Therefore, the standardized TDT block size will apply.

A TCR requesting an optional TDT block size not supported by the called side should not be answered with TBR.

5.2.4 *Transport connection clear (TCC) block*

5.2.4.1 If a transport connection cannot be established, the called terminal shall respond to the TCR block with a TCC block. The clearing cause shall indicate why the connection was not accepted.

It is up to the calling side whether the receipt of a TCC will cause complete disconnection or whether a new TCR with a parameter different from the first one will be sent (e.g. another extended transport layer address). In order to allow for subsequent TCRs, the sender of TCC may provide in the optional parameter field an appropriate parameter and associated value to indicate that another TCR is invited. The new optional parameter and its associated value(s) are for further study.

Note – There is no explicit transport connection termination procedure in this Recommendation. Therefore, the lifetime of the transport connection is directly correlated to the lifetime of the supporting network connection.

5.2.5 *Transport connection collision*

5.2.5.1 If the calling terminal receives a TCR block, it shall transfer a TBR block to notify the called terminal of the procedure error (see Annex B).

5.2.6 *Extended addressing*

5.2.6.1 The extended addressing capability may be used to address terminals in a multiterminal configuration.

The extension addresses for called and calling terminals are optional parameters to TCR and TCA. The use of the calling extension address is for further study.

5.2.6.2 The receiving terminal shall respond with a TCA according to Table 1/T.70.

TABLEAU 1/T.70

Received TCR	Receiver reaction	
	Multi-terminal with extended addressing [a]	Stand-alone terminal
Without extended addressing	Send TCA with extended addressing	Send TCA without extended addressing
With extended addressing	Send TCA with extended addressing [b]	Send TCA without extended addressing

[a] Multi-terminal configuration, with capability for extended addressing.

[b] If the called terminal is occupied or out of order, the call should be routed to a default terminal or mailbox. The sender will then be informed of the routing by the extension address of the connected terminal. The receiver of TCR may also in this case react by sending TCC.

5.2.6.3 The calling terminal may, when receiving a called terminal address in the TCA, act as specified in Table 2/T.70.

TABLE 2/T.70

Sent TCR	Calling terminal reaction		
	TCA received with:		
	No extended addressing	Correct extended addressing	Incorrect extended addressing
Without extended addressing	OK	Neglect extension (See Note)	
With extended addressing	[a]	OK	[a]

[a] Reaction left to the discretion of the calling terminal.

Note – Terminal complying with the 1980-1984 version of Recommendation T.70 may react by releasing the network connection.

5.3 *Description ofdata transfer procedures*

5.3.1 *General*

5.3.1.1 The data transfer procedure described in the following subsections applies only when the transport layer is in the data transfer phase, i.e. after completion of transport connection establishment and prior to clearing.

Note – When a connection is cleared, transport data blocks may be discarded. Hence it is left to the transport service user to define protocols able to cope with the various possible situations that may occur.

5.3.2 *Transport data block (TDT) length*

5.3.2.1 The standard maximum TDT block length to be supported by all terminals is 128 octets including the data block header octets. However, the TDT block length may be restricted to a lower value when the TDT block is concatenated with other TDT blocks (see § 5.5.3).

5.3.2.2 Other maximum data field lengths may be supported in conjunction with an optional TDT block size negotiation connection function (see §§ 5.5.4.3 and 5.5.5.3). Optional maximum data field lengths shall be chosen from the following: 256, 512, 1024 and 2048 octets. If the requested optional TDT block size cannot be supported, a shorter allowable TDT block size must be selected (see § 5.2.3.2).

The agreed maximum TDT block size should be aimed at for TDT blocks having the TSDU end mark set to 0 and a number of octets less than the agreed maximum shall not cause the receiving transport entity to reject this TDT block.

5.3.3 *Transport service data unit (TSDU) end*

5.3.3.1 The *TSDU end mark* is used to preserve the integrity of the TSDU. The TSDU end mark is set to binary 1 in the last TDT data block carrying information related to a certain TSDU. Exceptionally, this TDT block may be sent without carrying user information in order to allow for an immediate termination of a TSDU in certain error conditions.

In case of a TSDU that comprises a single TDT block the TSDU end mark is also set to 1. In all other cases the TSDU end mark is set to zero.

5.4 *Treatment of procedure errors*

5.4.1 A terminal shall send a TBR block to the remote terminal to report the receipt of an invalid or not implemented block (if not explicitly specified otherwise in this Recommendation). During the establishment of a transport connection, terminals shall not send a TBR block upon the receipt of a TCR block whose parameters or parameter values are invalid or not implemented. In this case, terminals shall act as if no errors have occurred and send the appropriate response (if any).

A terminal receiving a TBR block shall take appropriate recovery action.

Note 1 – A TBR whether invalid or valid shall not be answered by sending a TBR block.

Note 2 – Terminals complying with the Recommendation T.70 version of the 1981-1984 study period may react to all of the above indicated conditions by sending TBR.

Note 3 – The definition of invalid block/parameter, etc. is provided by the state transition tables (see Annex B).

Note 4 – A TCR of which the PV of the TPDU size parameter is less than 07 (which is the basic length of the transport block size) shall be considered as an invalid TPDU.

Note 5 – In the states 1.1 for the calling side and 2.1 for the calling and called side the terminal may react either by sending TBR or by releasing the network connection.

Attention: The state tables and state transition diagrams have to be read according to Notes 4 and 5 above.

5.5 *Formats*

5.5.1 *General*

5.5.1.1 Transport protocol data units (TPDUs) carrying transport service (TS) user information or control information are called *blocks* (see § 5.1.3). All blocks contain an integral number of octets.

5.5.1.2 Bits of an octet are numbered 8 to 1 where bit 1 is the low order bit and is transmitted first. Octets of a block are consecutively numbered starting from 1 and are transmitted in this order.

When consecutive octets are used to represent a binary number, the lower octet has the most significant value.

5.5.1.3 *TDT block(s)* are used to transfer a transport service data unit (TSDU) transparently whilst maintaining the structure of the latter by means of the TSDU end mark.

5.5.1.4 *Control blocks* (TCR, TCA, TCC, TBR) are used to control the transport protocol functions, including optional functions.

5.5.1.5 A parameter field is present in all control blocks within the basic transport service to indicate optional functions. The parameter field contains one or more parameter elements. The first octet of each parameter element contains a parameter code to indicate the function(s) requested.

The general coding structure is shown in Figure 5/T.70.

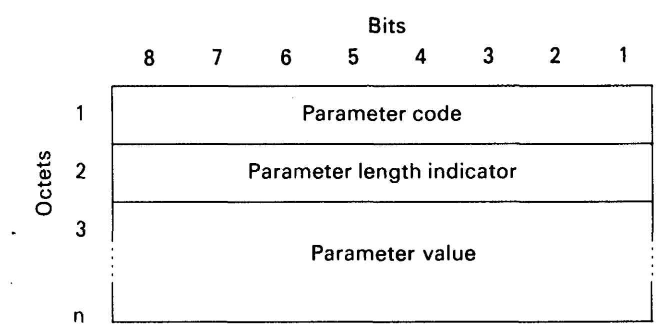


FIGURE 5/T.70
Parameter element coding structure

5.5.1.6 The parameter code field is binary coded and, without extension, provides for a maximum of 255 parameters. Parameter code 11111111 is reserved for extension of the parameter code. The extension mechanism is for further study.

Octet 2 indicates the length, in octets, of the parameter value field. The parameter field length is binary coded and bit 1 is the low order bit of this indicator.

Octet 3 and subsequent octets contain the value of the parameter identified in the parameter code field. The coding of the parameter value field is dependent on the function being requested.

5.5.2 *Structure of transport control and transport data blocks*

5.5.2.1 Figure 6/T.70 illustrates the general structure of transport layer blocks. A summary of transport layer blocks is given in Figure 7/T.70.

Bits 8 1	Octet 1	Octet 2	Octet 3 ... Octet N	Octet N+1 ... Octet M
	Length indicator	Block type	Functional code field (fixed format)	Parameter or data field (variable format)

CCITT-43961

FIGURE 6/T.70

General block structure

	Octet 1	Octet 2	Octet 3	Octet 4	Octet 5	Octet 6	Octet 7	
TCR	Length	11100000	00000000	00000000	Source	reference	00000000	Parameters
TCA	Length	11010000	Destination	reference	Source	reference	00000000	Parameters
TCC	Length	10000000	Destination	reference	Source	reference	Clearing cause	Parameters
TBR	Length	01110000	Destination	reference	Reject cause	Parameters		
TDT	Length	11110000	0000000	Data				

TSDU end mark

CCITT-43971

Note — The terms "source" and "destination" refer to the initiator and the recipient of the transport protocol data unit (TPDU), respectively. The value of the "source reference" is a local system parameter. The source reference of a received transport block is to be used as destination reference in the response to that transport block.

FIGURE 7/T.70

Transport layer block types

5.5.2.2 *Length indicator (LI) field*

5.5.2.2.1 Octet 1 contains the length indicator (LI). The value of this indicator is a binary number that represents the length in octets of the control block (including parameters) and the header length in octets of data blocks (excluding any subsequent user information). In both cases this length does not include octet 1.

5.5.2.2.2 The basic LI value shall be restricted to 127 (i.e., a binary value of 01111111). The use of higher LI values and the use of the binary value 11111111 for extension purposes is for further study.

5.5.2.3 *Block type field*

5.5.2.3.1 Octet 2 contains the block type code. Bits 1 to 4 of octet 2 are set to 0 for all transport layer blocks currently defined. It is for further study to determine whether or not bits 1 to 4 are required for future extension to the range of transport layer blocks currently defined or are used for other functions.

5.5.2.4 *Functional code field*

5.5.2.4.1 Octet 3 and subsequent octets contain functional codes in a fixed format as per the block type (see Figure 7/T.70).

5.5.2.5 *Parameter or TSDU field*

5.5.2.5.1 A parameter field or a data field containing transport service (TS) user data may optionally follow the functional code field.

5.5.3 *Concatenation*

5.5.3.1 Concatenation of transport control and/or transport data blocks is currently not applicable to this Recommendation. However, where concatenation is used in the future, the arrangement shown in Figure 8/T.70 would apply.

Bits 8 7 6 5 4 3 2 1	
HDLC flag	Link level
HDLC address field	
HDLC control field	
Network header (e.g. Rec. X.25 or as per § 3.3.3)	Network level
Transport control block	Transport level
Transport control block	
. . .	
Transport data block header	
Transport user data . . .	Session level and above
HDLC frame check-sequence	Link level
HDLC flag	

Note — This figure does not imply that a transport data or control block will fit within a single network data block.

FIGURE 8/T.70

Information field structure of HDLC I-frame (example)

5.5.4 *Transport connection request (TCR) block format*

5.5.4.1 Figure 9/T.70 illustrates the format of the TCR block.

Octets	Bits 8	7	6	5	4	3	2	1
1	Length (octets)							
2	1	1	1	0	0	0	0	0[a)]
3	0	0	0	0	0	0	0	0[b)]
4	0	0	0	0	0	0	0	0[b)]
5	Source							
6	reference							
7	0	0	0	0	0	0	0	0[c)]
8 … n	Parameters[d)] (optional)							

a) Block type: TCR.

b) Octets 3 and 4 are not used and shall be set to zero.

c) Transport service extention field: Octet 7 is reserved for any future extention such as providing for a range of transport service classes. In the basic transport service this octet shall be set to zero.

d) The parameter field is present only when the terminal is requesting an optional transport connection function.

FIGURE 9/T.70

Transport connection request block

5.5.4.2 *Parameters for extended addressing*

Separate parameters are provided for the indication of called and calling extension addresses. The coding of these parameters is shown in Figure 10/T.70. The setting of bit 8 for extended addressing should be ignored by the transport layer.

The use of more than one called extension address is for further study.

Octets	Bits 8 7 6 5 4 3 2 1		
1	Extension address parameter type		11000001 for calling address type or 11000010 for called address type
2	Parameter length indicator		
3	IA5 digit	1	
	IA5 digit	2	
	⋮	⋮	
	IA5 digit	n	

FIGURE 10/T.70

Extended addressing

5.5.4.3 *Parameter for transport data block size negotiation*

This parameter defines the proposed maximum transport data block size (in octets including the transport data block header) to be used over the requested transport connection. The coding of this parameter is shown in Figure 11/T.70.

Octets \ Bits	8	7	6	5	4	3	2	1	
1	1	1	0	0	0	0	0	0	Parameter type code
2	0	0	0	0	0	0	0	1	Parameter length indicator
3	0	0	0	0	X	X	X	X	Transport data block size

XXXX		
1 0 1 1	=	2048 octets
1 0 1 0	=	1024 octets
1 0 0 1	=	512 octets
1 0 0 0	=	256 octets
0 1 1 1	=	128 octets

FIGURE 11/T.70

Transport data block size parameter

5.5.5 *Transport connection accept (TCA) block format*

5.5.5.1 Figure 12/T.70 illustrates the format of the TCA block.

Octets \ Bits	8	7	6	5	4	3	2	1
1	Length (octets)							
2	1	1	0	1	0	0	0	0[a)]
3	Destination							
4	reference							
5	Source							
6	reference							
7	0	0	0	0	0	0	0	0[b)]
8 … n	Parameters (optional)[c)]							

a) Block type: TCA.

b) Transport service extension field; Octet 7 is reserved for any future extension such as providing for a range of transport service classes. In the basic transport service this octet shall be set to zero irrespective of the setting in the TCR block.

c) The parameter field is present only when the terminal is requesting or confirming an optional transport connection function.

FIGURE 12/T.70

Transport connection accept block

5.5.5.2 *Parameters for extended addressing*

See § 5.5.4.2.

5.5.5.3 *Parameter for transport data block size negotiation*

See § 5.5.4.3. The parameter value shall be equal to or less than the value specified in the TCR block.

5.5.6 *Transport connection clear (TCC) block format*

5.5.6.1 Figure 13/T.70 illustrates the format of the TCC block.

Octets	Bits 8	7	6	5	4	3	2	1
1	Length (octets)							
2	1	0	0	0	0	0	0	0[a]
3	Destination							
4	reference							
5	Source							
6	reference							
7	Clearing cause[b]							
8 … n	Parameters (optional)							

[a] Block type: TCC

[b] Clearing cause:

		Bits 8 7 6 5 4 3 2 1
0 – Reason not specified	=	0 0 0 0 0 0 0 0
1 – Terminal occupied	=	0 0 0 0 0 0 0 1
2 – Terminal out of order	=	0 0 0 0 0 0 1 0
3 – Address unknown	=	0 0 0 0 0 0 1 1

FIGURE 13/T.70

Transport connection clear block

5.5.6.2 *Parameter for additional clearing information*

This parameter is provided to allow additional information relating to the clearing of the connection. The coding of this parameter is given in Figure 14/T.70 below.

Octets	Bits 8	7	6	5	4	3	2	1	
1	1	1	1	0	0	0	0	0	Parameter code
2	Parameter length indicator								
3 … n	Additional information								

FIGURE 14/T.70

Additional clearing information parameter

5.5.7 *Transport block reject (TBR) block format*

5.5.7.1 Figure 15/T.70 illustrates the format of the TBR block.

Octets	Bits 8	7	6	5	4	3	2	1
1	Length (octets)							
2	0	1	1	1	0	0	0	0[a]
3	Destination							
4	reference							
5	Reject cause[b]							
6 … n	Parameter field							

[a] Block type: TBR

[b] Reject cause:

		Bits 8 7 6 5 4 3 2 1
0 – Reason not specified	=	0 0 0 0 0 0 0 0
1 – Function not implemented	=	0 0 0 0 0 0 0 1
2 – Invalid block	=	0 0 0 0 0 0 1 0
3 – Invalid parameter	=	0 0 0 0 0 0 1 1

FIGURE 15/T.70

Transport block reject block

5.5.7.2 *Rejected block parameter* (mandatory)

This parameter is used to indicate the bit pattern of the rejected block up to and including the octet that caused the rejection. Only the first detected procedural error or parameter, which cannot be acted upon, shall be indicated by this method. The coding of this parameter is given in Figure 16/T.70.

Octets \ Bits	8	7	6	5	4	3	2	1
1	1	1	0	0	0	0	0	1
2	Parameter length indicator							
3 … n	Rejected information							

FIGURE 16/T.70

5.5.8 *Transport data block (TDT) format*

5.5.8.1 Figure 17/T.70 illustrates the format of the TDT block.

Octets \ Bits	8	7	6	5	4	3	2	1
1	Length (octets)							
2	1	1	1	1	0	0	0	0 [a]
3	TSDU end [b]	0	0	0	0	0	0	0
4 … n	Data							

[a] Block type: TDT

[b] TSDU end: indicates the end of TSDU when set to 1

FIGURE 17/T.70

Transport data block

ANNEX A

(to Recommendation T.70)

A.1 *Transport and network service*

The transport service (TS) is provided by the transport protocol (TP) making use of the services available from the network layer. This Annex also defines the TS characteristics which the TS users may exploit.

Interactions between TS users and the TS provider take place at the two TS access points (TSAP) (see Figures A-1/T.70 to A-6/T.70). Information is passed between a TS user and a TS provider by means of primitives, which may convey parameters.

Primitives are abstract representations of interactions. They are solely descriptive and do not represent a specification or implementation.

The occurrence of a primitive is a logically instantaneous and indivisible event. The event occurs at a logically separate instant, which cannot be interrupted by another event. Only primitives of global significance are mentioned (having an impact on the remote user).

The following types of primitives are defined:

a) request primitive

b) indication primitive

c) response primitive

d) confirm primitive.

The primitives a) and c) are directed from the service user to the service provider, b) and d) are going in the opposite direction.

"Transport" is designated by T, "Network" is designated by N. The terms CONNECT, DATA, DISCONNECT as part of a primitive name indicate that the primitive is used for establishment, data transfer, release of a transport connection (TC) or network connection (NC).

Examples:

T-CONNECT request	request to establish a TC
T-DATA request	request to transmit TS user data
N-DISCONNECT indication	indication that the NC has been released.

The relationship between valid sequences of TS primitives and the appropriate protocol elements is shown in Figures A-1/T.70 to A-6/T.70. The sequences of valid network service (NS) primitives are illustrated in Figures A-7/T.70 to A-12/T.70.

A.1.1 *Transport service*

The interactions shown in Figures A-1/T.70 to A-6/T.70 are not exhaustive.

A.1.1.1 *Transport connection establishment*

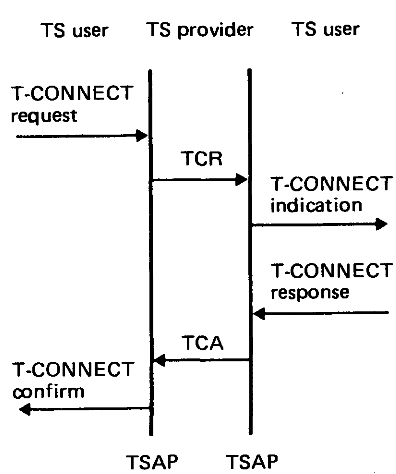


FIGURE A-1/T.70

Successful TC establishment

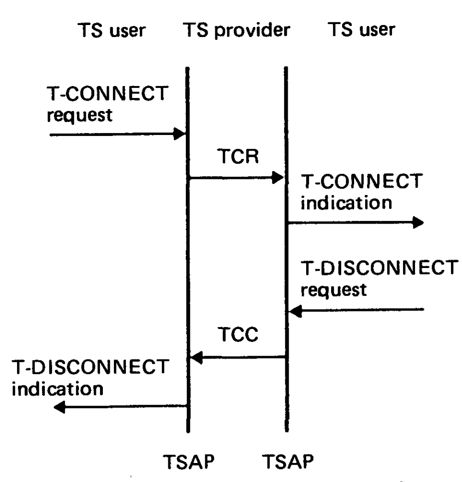
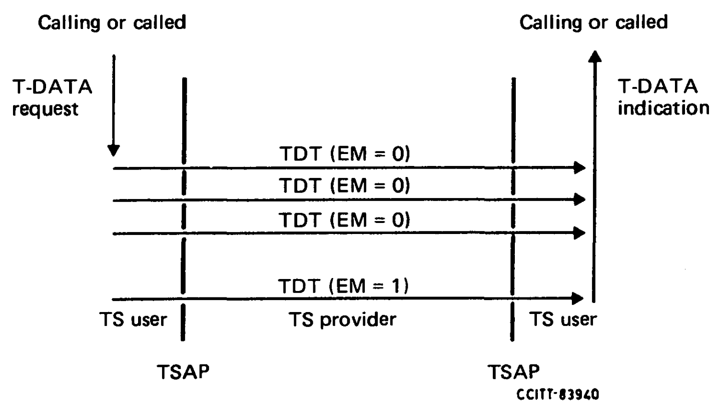


FIGURE A-2/T.70

Rejection of TC establishment by TS user

A.1.1.2 *Transfer phase*

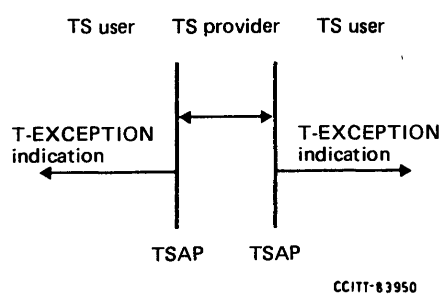


Note – This is one method of realizing segmenting/reassembling.

FIGURE A-3/T.70

T-DATA transfer

A.1.1.3 *Transport service error reporting*



Note – The use of this primitive is optional.

FIGURE A-4/T.70

Transport service error reporting

A.1.1.4 *TC release*

At present only the implicit release of TC is defined (see § 5.2.4.1 of this Recommendation).

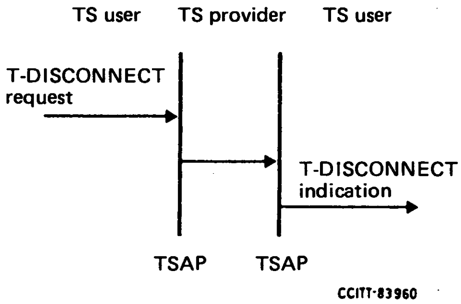


FIGURE A-5/T.70

TC release initiated by TS user

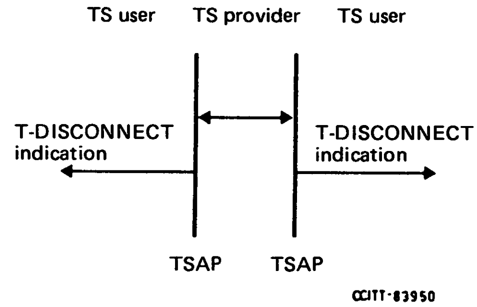


FIGURE A-6/T.70

TC release initiated by the TS provider

A.1.2 *Network service*

Figures A-7/T.70 to A-12/T.70 show the relationships of network service (NS) primitives at both sides of an NC.

A.1.2.1 *Network connection establishment*

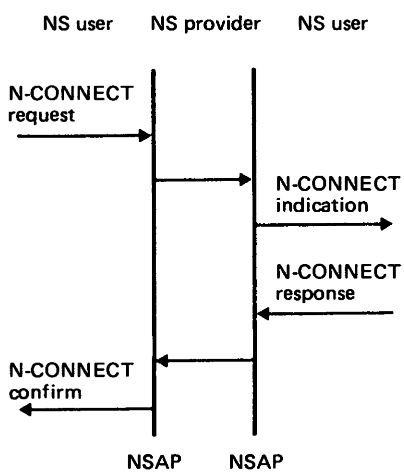


FIGURE A-7/T.70

Successful NC establishment

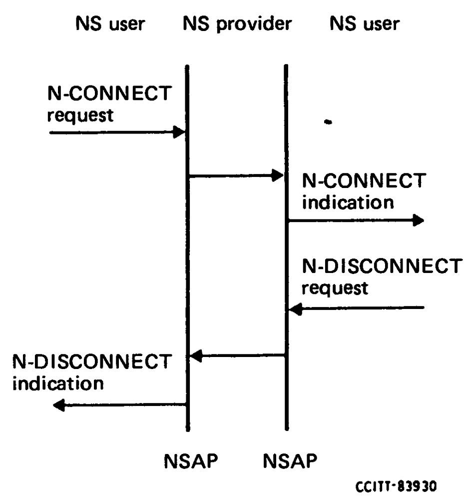


FIGURE A-8/T.70

Rejection of NC establishment by NS user

A.1.2.2 *Network data transfer*

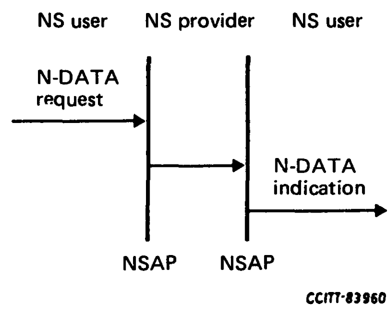


FIGURE A-9/T.70

N-DATA transfer

A.1.2.3 *Network service error reporting*

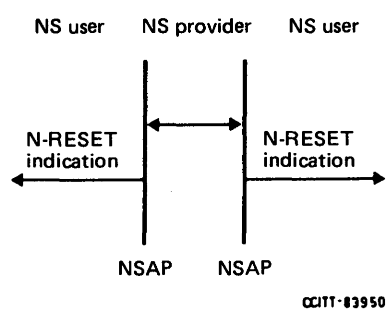


FIGURE A-10/T.70

Network service error reporting

A.1.2.4 *Network connection release*

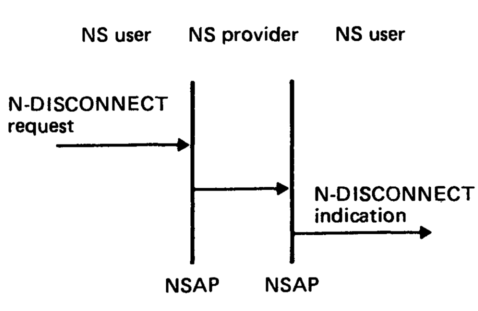


FIGURE A-11/T.70

NS release initiated by NS user

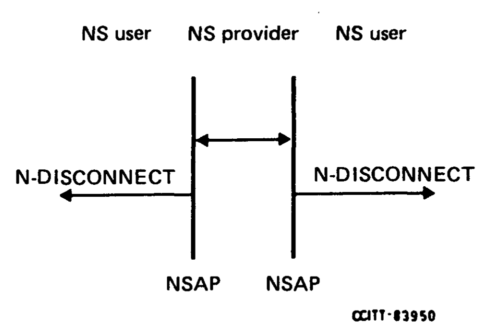


FIGURE A-12/T.70

NC release initiated by the NS provider

A.2 *State transition diagrams for the basic transport layer procedures*

This part represents detailed state transition diagrams for the basic transport procedures.

Two description levels are used:

a) *Protocol level*

This level addresses only the peer to peer protocol activities between two transport entities. It identifies the protocol state, events [receipt of transport protocol data units (TPDUs)] and actions (sending of TPDUs).

b) *Detailed level*

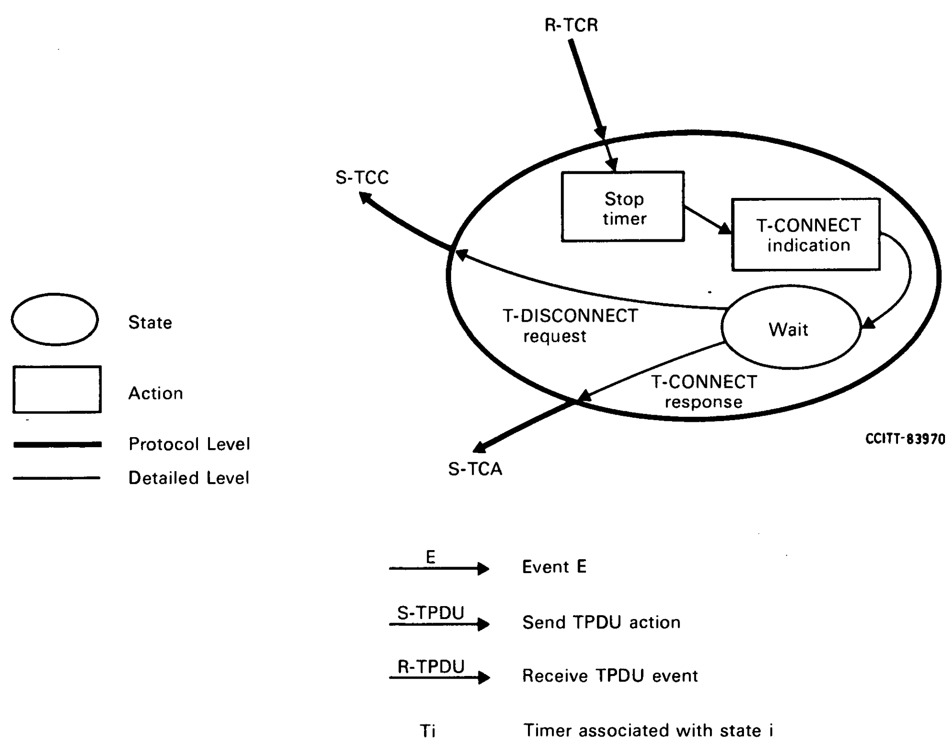
This level addresses the inter-layer and local activities. It identifies the events, actions, conditions and states within each of the protocol level states. The inter-layer activities are described using the transport service primitives defined in the first part of this Annex.

Example (see Figure A-13/T.70)

For pure illustrative reasons, the example shows a simplified description of state 1 (response pending, called side) of the state transition diagram of this Recommendation. The event R-TCR may be answered either by sending the action S-TCA or S-TCC.

The events and actions are not interruptable. They will complete their transfer irrespective of the occurrence of other events.

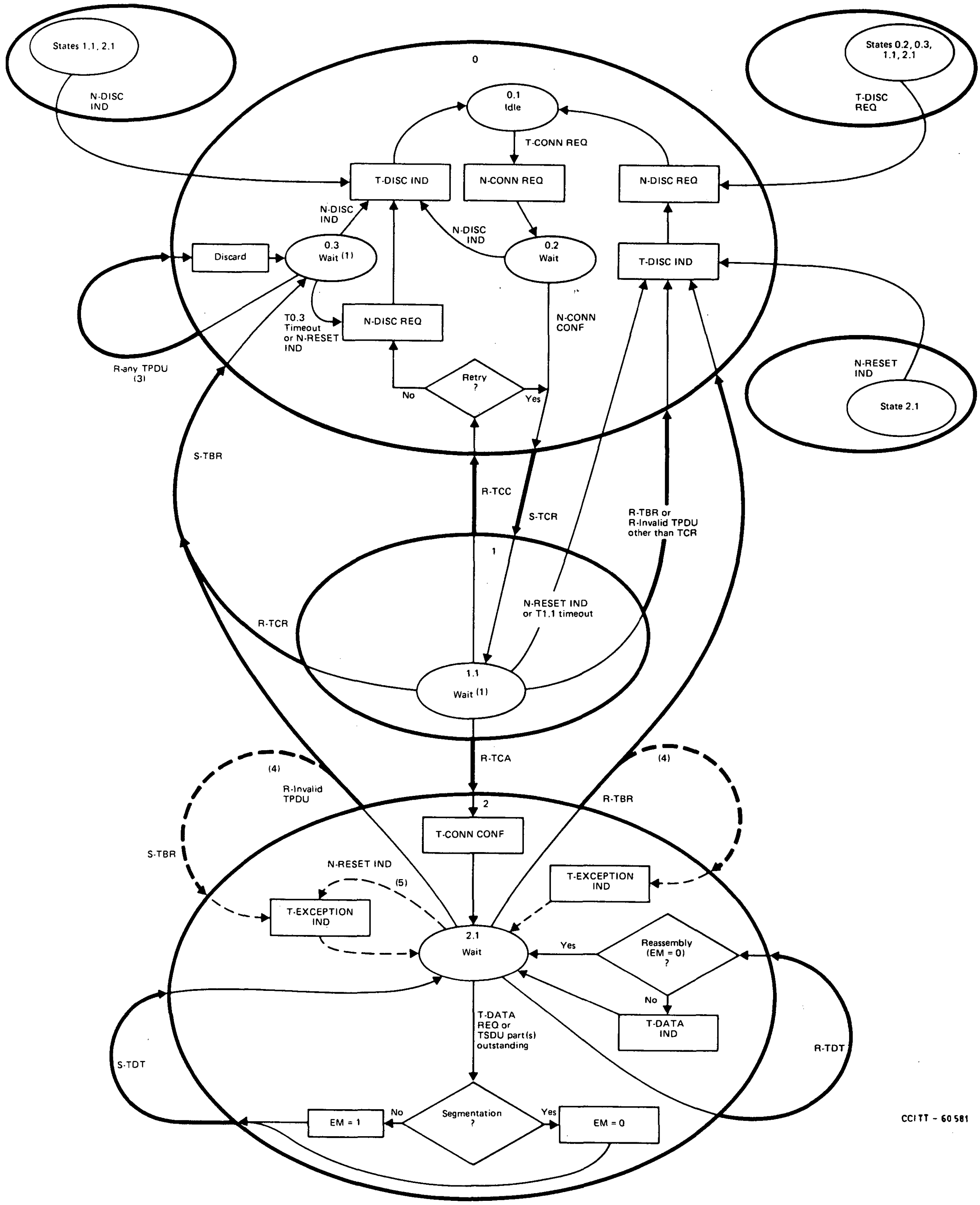
The detailed state transition diagrams are given in Figures A-14/ and A-15/T.70.



Note 1 — Each TPDU is transferred by N-DATA request. The NSDU will contain the TPDU.

Note 2 — Each TPDU is received by N-DATA indication. The NSDU will contain the TPDU.

FIGURE A-13/T.70



(1) States 0.3 and 1.1 have timers, T0.3 and T1.1 respectively. When entering any of these states the associated timer is started. The timer is stopped upon exit from the state [see (3)].

(2) There are other valid methods for describing segmentation.

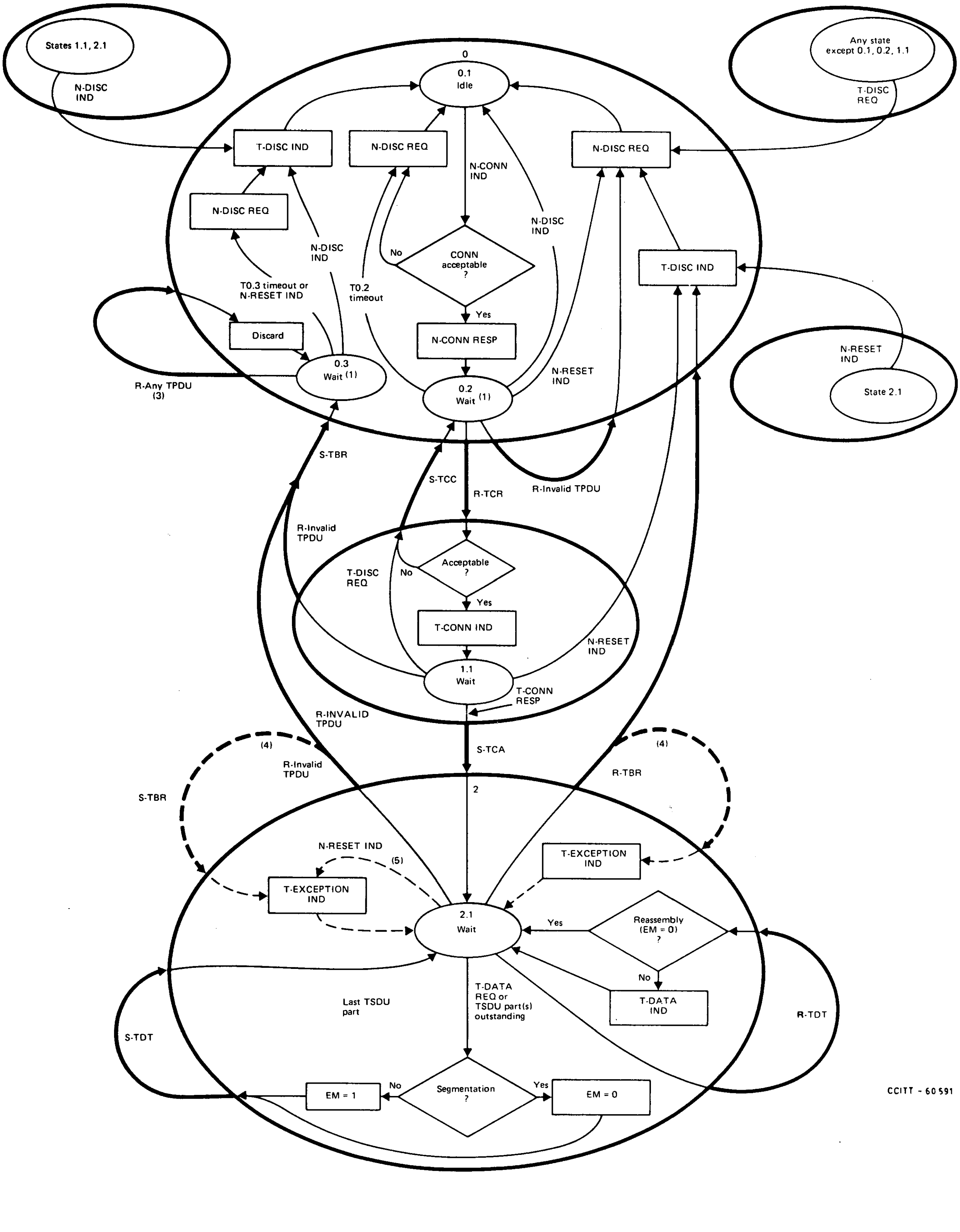
(3) This state transition does not start/stop T0.3 timer.

(4) Optional transition (shown in broken lines) if "T-EXCEPTION IND" is provided.

(5) Optional transition (shown in broken lines) if "T-EXCEPTION IND" is provided. The use of this option is independent of the use of option (4) above.

FIGURE A-14/T.70

Teletex transport state transition diagram (calling side)



(1) States 0.2 and 0.3 have timers, T0.2 and T0.3 respectively. When entering any of these states the associated timer is started. The timer is stopped upon exit from the state [see (3)].

(2) There are other valid methods for describing segmentation.

(3) This state transition does not start/stop T0.3 timer.

(4) Optional transition (shown in broken lines) if "T-EXCEPTION IND" is provided.

(5) Optional transition (shown in broken lines) if "T-EXCEPTION IND" is provided. The use of this option is independent of the use of option (4) above.

FIGURE A-15/T.70

Teletex transport state transition diagram (called side)

ANNEX B

(to Recommendation T.70)

B.1 *State tables*

The state tables:

B-1/T.70: Transport connection establishment, calling side

B-2/T.70: Transport connection establishment, called side

B-3/T.70: Data phase (symmetrical protocol)

present the transitions of the transport protocol in a table form in contrast to the diagram form to be seen in Annex A. While the diagrams are useful to overview the protocol mechanism the appropriate tables give clear information of which event is possible in which state and which actions are to be performed. Moreover each of the events and conditions is combined with a shortening in brackets (e.g.: E 5) which is a pointer to the 2nd part of this annex, so that the reader of these tables can easily come to know which meaning a certain event, action or condition has.

An impossible event related to a certain state can be recognized by an empty field in the crossing-point of the state and the event.

B.2 *Lists of events, actions and conditions*

The lists of events (Table B-4/T.70), actions (Table B-5/T.70) and conditions (Table B-6/T.70) intend to care for detailed explanations and clarification related to the protocol components (events, actions and conditions) found in the diagrams and tables.

All the components in the tables are accompanied by a list number (e.g. E 1, A 10, C 3, etc.) which can be interpreted as a pointer to the corresponding additional information in the lists. The letters E, A, C of the list numbers stand for Event, Action, Condition.

The following abbreviations are used:

EM	End Mark
LI	Length Indicator of the transport block (octet 1)
loc.	local
NC	Network Connection
NS	Network Service
NSDU	Network Service Data Unit
PLI	Parameter Length Indicator
TC	Transport Connection
TP	Transport Protocol
TPDU	Transport Protocol Data Unit
TS	Transport Service
TSDU	Transport Service Data Unit

AND, OR and NOT (used mainly in E 5) shall be considered as the known Boolean operators.

TABLE B-1/T.70

State table for calling side

Event			State	Idle												Waiting			
				0.1				0.2				0.3				1.1			
No.	Local	Protocol event	Service primitive	Local	Protocol action	Service primitive	Final state	Local	Protocol action	Service primitive	Final state	Local	Protocol action	Service primitive	Final state	Local	Protocol action	Service primitive	Final state
1.1		R-TCR (E 1)											Discard any R-TPDU (A 14)		0.3	STOP T1.1 (A 1) START T0.3 (A 2)	S-TBR (A 3)		0.3
1.2		R-TCC (E 2) Retry (C 1)													0.3	RESTART T1.1 (A 6)	S-TCR (A 7)		1.1
1.3		R-TCC (E 2) No retry (C 2)													0.3	STOPT T1.1 (A 1)		N-DISC req. (A 4) T-DISC ind. (A 5)	0.1
1.4		R-TCA (E 3)													0.3	STOPT T1.1 (A 1)		T-CONN. conf (A 8)	2.1
1.5		R-TBR (E 4)													0.3	STOP T1.1 (A 1)		T-DISC. ind. (A 5) N-DISC. req. (A 4)	0.1
1.6		R-invalid TPDU (E 5)													0.3	STOP T1.1 (A 1)		T-DISC. ind. (A 5) N-DISC. req. (A 4)	0.1
1.7			T-CONN. req.(E 6)			T-CONN. req.(A 10)	0.2												
1.8			N-CONN. conf.(E 7)					START T1.1 (A 12)	S-TCR (A 7)		1.1								
1.9			N-DISC ind. (E 8)							T-DISC. ind. (A 5)	0.1	STOP T0.3 (A 13)		T-DISC. ind. (A 5)	0.1	STOP T1.1 (A 1)		T-DISC. ind. (A 5)	0.1
1.10			N-RESET ind. (E 9)									STOP T0.3 (A 13)		N-DISC. rec.(A 4) T-DISC. ind. (A 5)	0.1	STOP T1.1 (A 1)		T-DISC. ind.(A 5) N-DISC. req. (A 4)	0.1
1.11			T-DISC req. (E 10)							N-DISC. req. (A 4)	0.1	STOP T0.3 (A 13)		N-DISC. req. (A 4)	0.1	STOP T1.1 (A 1)		N-DISC. req. (A 4)	0.1
1.12	TIME-OUT (E 11)											STOP T0.3 (A 13)		N-DISC. req. (A 4) T-DISC. ind. (A 5)	0.1	STOP T1.1 (A 1)		T-DISC. ind. (A 5) N-DISC. req. (A 4)	0.1

Req. Request
Ind. Indication
Conf. Confirmation

CONN. CONNECTION
DISC. DISCONNECTION

TABLE B-2/T.70

State table for called side

	Event		State	Idle												Waiting			
				0.1				0.2				0.3				1.1			
No.	Local	Protocol event	Service primitive	Local	Protocol action	Service primitive	Final state	Local	Protocol action	Service primitive	Final state	Local	Protocol action	Service primitive	Final state	Local	Protocol action	Service primitive	Final state
2.1		R-TCR (E 1) Acceptable (C 3)						STOP T0.2 (A 11)		T-CONN. ind. (A 15)	1.1		Discard any R-TPDU (A 14)		0.3				
2.2		R-TCR (E 1) Not acceptable (C 4)						RESTART T0.2 (A 16)	S-TCC (A 17)		0.2				0.3				
2.3		R-invalid TDPU (E5)						STOP T0.2 (A 11)	N-DISC. req. (A 4)		0.1				0.3	START T0.3 (A 2)	S-TBR (A 3)		0.3
2.4			N-CONN ind. (E 12) Acceptable (C 5)	START T0.2 (A 9)		N-CONN. resp. (A 22)	0.2												
2.5			N-CONN ind. (E 12) Not acceptable (C 6)			N-DISC. req. (A 4)	0.1												
2.6			T-CONN. resp. (E 13)														S-TCA (A 24)		2.1
2.7			N-DISC. ind.(E 8)					STOP T0.2 (A 11)			0.1	STOP T0.3 (A 13)		T-DISC. ind. (A 5)	0.1			T-DISC. ind. (A 5)	0.1
2.8			N-RESET ind.(E 9)					STOP T0.2 (A 11)		N-DISC. req. (A 4)	0.1	STOP T0.3 (A 13)		T-DISC. ind. (A 5) N-DISC. req. (A 4)	0.1			T-DISC. ind. (A 5) N-DISC. req. (A 4)	0.1
2.9			T-DISC. req. (E 10)									STOP T0.3 (A 13)		N-DISC. req. (A 4)	0.1	START T0.2 (A 9)	S-TCC (A 17)		0.2
2.10	TIME-OUT (E 11)							STOP T0.2 (A 11)		N-DISC. req. (A 4)	0.1	STOP T0.3 (A 13)		T-DISC. ind.(A 5) N-DISC. req. (A4)	0.1				

Req. Request
Ind. Indication
Resp. Response

CONN. CONNECTION
DISC. DISCONNECTION

TABLE B-3/T.70

Data phase (symmetrical protocol)

	Event			State: Data phase 2.1			
	Local	Protocol event	Service primitive	Local	Protocol action	Service primitive	Final state
3.1		R-TDT (E 14) EM = 0 (C 7)					2.1
3.2		R-TDT (E 14) EM = 1 (C 8)				T-DATA ind. (A 18)	2.1
3.3		R-TBR (E 4) Recovery (C 9)				T-EXCEPT. ind. (A 19)	2.1
3.4		R-TBR (E 4) No recovery (C 10)				T-DISC. ind. (A 5) N-DISC. req. (A 4)	0.1
3.5		R-inv. TPDU (E 5) Recovery (C 9)			S-TBR (A 3)	T-EXCEPT. ind. (A 19)	2.1
3.6		R-inv. TPDU (E 5) No recovery (C 10)		START T0.3 (A 2)	S-TBR (A 3)		0.3
3.7			T-DATA req. (E 15) Segm. (C 11)		S-TDT (EM = 0) (A 20)		2.1
3.8			T-DATA req. (E 15) No segm. (C 12)		S-TDT (EM = 1) (A 21)		2.1
3.9	TSDU part(s) outsand (E 16) Segm. (C 11)				S-TDT (EM = 0) (A 20)		2.1
3.10	TSDU part(s) outsand (E 16) No segm. (C 12)				S-TDT (EM = 1) (A 21)		2.1
3.11			N-RESET ind. (E 9) Recovery (C 9)			T-EXCEPT. ind. (A 19)	2.1
3.12			N-RESET ind. (E 9) No recovery (C 10)			T-DISC. ind. (A 5) N-DISC. req. (A 4)	0.1
3.13			N-DISC. ind. (E 8)			T-DISC. ind. (A 5)	0.1
3.14			T-DISC. ind. (E 10)			N-DISC. ind. (A 4)	0.1

Ind. Indication

DISCO. DISCONNECTION

EXCEP. EXCEPTION

TABLE B-4/T.70

List of events

No.	Name	Type	Description
E 1	R-TCR	TP	Layer 4 receives via the NS N-DATA indication a TPDU including the transport block TCR.
E 2	R-TCC	TP	Layer 4 receives via the NS N-DATA indication a TPDU including the transport block TCC.
E 3	R-TCA	TP	Layer 4 receives via the NS N-DATA indication a TPDU including the transport block TCA.
E 4	R-TBR	TP	Layer 4 receives via the NS N-DATA indication a TPDU including the transport block TBR.
E 5	R-invalid TPDU	TP	Layer 4 receives via the NS N-DATA indication a TPDU whose validity check fails due to following reasons: – syntactical errors – procedure errors 1. *Invalid TPDUs due to syntactical errors* 1.1 TCR: 1.1.1 The value of octet 1 (LI): 1.1.1.1 ≠ the number of the TCR block octets minus 1 OR 1.1.1.2 is greater than 127 OR 1.1.1.3 is smaller than 6 OR 1.1.2 see 1.6 1.2 TCA: 1.2.1 The value of octet 1 (LI): 1.2.1.1 ≠ the number of the TCA block octets minus 1 OR 1.2.1.2 is greater than 127 OR 1.2.1.3 is smaller than 6 OR 1.2.2 see 1.6 OR 1.2.3 The value of octet 3 (4 resp.) ≠ octet 5 (6 resp.) of the appropriate TCR block OR 1.2.4 The value of octet 7 ≠ 0 OR 1.2.5 The parameter "Transport Data Block Size" is present: 1.2.5.1 AND its value ≠ 07 (hexadecimal), in response to a TCR block without the transport data block size parameter OR 1.2.5.2 AND its value does not respond to the rules according to § 5.2.3.2 of Recommendation T.70 OR 1.2.5.3 AND its value is different from the values (hexadecimal): 07, 08, 09, 0A, 0B OR 1.2.5.4 AND the PLI > 1 OR 1.2.6 $LI \neq 6 + 2N + \sum_{i=1}^{N} PLI$ where N is the number of parameters 1.3 TCC: 1.3.1 The value of the LI (octet 1): 1.3.1.1 ≠ the number of the TCC block octets minus 1 OR 1.3.1.2 is greater than 127 OR 1.3.1.3 is smaller than 6 OR 1.3.2 see 1.6 OR 1.3.3 The value of octet 3 (4 resp.) ≠ octet 5 (6 resp.) of the appropriate TCR block OR 1.3.4 $LI \neq 6 + 2N + \sum_{i=1}^{N} PLI$ where N is the number of parameters 1.4 TBR: (also see § 5.4.1, Note 1) 1.4.1 The value of the LI: 1.4.1.1 ≠ the number of the TBR block octets minus 1 OR 1.4.1.2 is greater than 127 OR 1.4.1.3 is smaller than 7 OR 1.4.2 see 1.6 OR
	(continued)		

<table>
<tr><th>No.</th><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>E 5</td><td>R-invalid TPDU (cont.)</td><td>TP</td><td>1.4.3 The value of octet 3 (4 resp.) $\neq$ octet 5 (6 resp.) of the appropriate TC establishment block (TCR resp. TCA) received from the peer entity OR
1.4.4 The value of LI minus 6 $\neq$ value of the PLI OR
1.4.5 The Rejected block parameter is not present

1.5 TDT:
1.5.1 The value of the LI $\neq$ 2 OR
1.5.2 The TSDU end mark is 0 AND the information field is empty OR
1.5.3 The TDT block size is larger than negotiated in the establishment phase

1.6 No identified block:
The value of the TPDU octet 2 is not equal to one of the following values (hexadecimal): EX, DO, 80, 70, FO.
X may be in the range of $0 \leq X \leq F$.

2. Invalid TPDUs due to procedure errors

Failure cases:
2.1 After S-TCR:
2.1.1 NOT R-TCA OR
2.1.2 NOT R-TCC OR
2.1.3 NOT R-TBR OR
2.2 After S-TCA:
2.2.1 NOT R-TDT OR
2.2.2 NOT R-TBR OR
2.3 After S-TDT:
2.3.1 Not R-TDT OR
2.3.2 Not R-TBR OR
2.4 After S-TCC: NOT R-TCR OR
2.5 After S-TBR: NOT R-TDT (in state 2.1) OR
2.6 After R-TDT (EM = 1): R-empty TDT (EM = 1) OR
2.7 After R-empty (EM = 1): R-empty TDT (EM = 1) OR
2.8 After N-CONNECT response: NOT R-TCR</td></tr>
<tr><td>E 6</td><td>T-CONNECT request</td><td>TS</td><td>Layer 5 requests a TC from layer 4.</td></tr>
<tr><td>E 7</td><td>N-CONNECT confirm</td><td>NS</td><td>Affirmative answer to N-CONNECT request (A 10); a NC is existing now.</td></tr>
<tr><td>E 8</td><td>N-DISCONNECT indication</td><td>NS</td><td>Report from layer 3 to layer 4 that the NC is not existing (any more).</td></tr>
<tr><td>E 9</td><td>N-RESET indication</td><td>NS</td><td>Indication to layer 4 that an error has occurred in layer 1, 2 or 3, possibly with data loss. The NC is kept existing.</td></tr>
<tr><td>E 10</td><td>T-DISCONNECT request</td><td>TS</td><td>Layer 5 requests a TC clearing from layer 4.</td></tr>
<tr><td>E 11</td><td>TIMEOUT</td><td>loc.</td><td>The timer presently surveying a state reached its limit. Following value ranges are defined:
<table>
<tr><th></th><th colspan="2">Values</th></tr>
<tr><th>States</th><th>Calling side</th><th>Called side</th></tr>
<tr><td>0.2</td><td>not applicable</td><td>45 s ± 30 s</td></tr>
<tr><td>0.3</td><td>6 s ± 4 s</td><td>6 s ± 4 s</td></tr>
<tr><td>1.1</td><td>45 s ± 30 s</td><td>not applicable</td></tr>
</table></td></tr>
</table>

TABLE B-4/T.70 *(end)*

No.	Name	Type	Description
E 12	N-CONNECT indication	NS	Indication to layer 4 by the layer 3 that an NC is being established; the answer to this is N-CONNECT response (A 22) or N-DISCONNECT request (A 4).
E 13	T-CONNECT response	TS	Affirmative answer by the layer 5 to T-CONNECT indication (A 15).
E 14	R-TDT	TP	Layer 4 receives via the NS N-DATA indication, an NSDU including the transport block TDT.
E 15	T-DATA request	TS	Layer 5 requests the transmission of data. Whether this is a complete TSDU or not, is a local matter, and not subject of this definition.
E 16	TSDU part(s) outstanding	loc.	Layer 4 is ready to send the next TDT block.

TABLE B-5/T.70

List of actions

No.	Name	Type	Description
A 1	STOP Timer T1.1	loc.	Timer T1.1 surveying the state 1.1 is stopped.
A 2	START Timer T0.3	loc.	Timer T0.3 surveying the state 0.3 is started after having been reset.
A 3	S-TBR	TP	Via the NS N-DATA request a NSDU including the transport block TBR is sent to the peer entity.
A 4	N-DISCONNECT request	NS	Layer 4 requests the layer 3 to release the offered or existing NC.
A 5	T-DISCONNECT indication	TS	Layer 5 is informed by the layer 4 that the TC being established or existing is cleared.
A 6	RESTART T1.1	loc.	Timer T1.1 surveying the state 1.1 is reset and started again. Moreover, it is necessary either to limit the number of T1.1-restarts or to limit the sum of all the times of T1.1; otherwise, an infinite loop S-TCR – R-TCC – S-TCR – etc., would be allowed.
A 7	S-TCR	TP	Via the NS N-DATA request a NSDU including the transport block TCR is sent to the peer entity.
A 8	T-CONNECT confirm	TS	Affirmative answer to the event T-CONNECT request (E 6) indicating that the data phase of the TC has been entered.
A 9	START T0.2	loc.	Timer T0.2 surveying the state 0.2 is started after having been reset.
A 10	N-CONNECT request	NS	Layer 4 requests the layer 3 for an NC to be established.
A 11	STOP T0.2	loc.	Timer T0.2 surveying the state 0.2 is stopped.
A 12	START T1.1	loc.	Timer T1.1 surveying the state 1.1 is started after having been reset.
A 13	STOP T0.3	loc.	Timer T0.3 surveying the state 0.3 is stopped.
A 14	DISCARD any R-TPDU	TS	Any data received by N-DATA indication are discared. The transmission of further data is stopped.
A 15	T-CONNECT indication	TS	Layer 4 indicates a request for a TC-establishment to the layer 5.
A 16	RESTART T0.2	loc.	Timer T0.2 surveying the state 0.2 is reset and started again.
A 17	S-TCC	TP	Via the NS N-DATA request, an NSDU including the transport block TCC is sent to the peer entity.
A 18	T-DATA indication	TS	Layer 4 indicates the receipt of a complete TSDU to the layer 5. How and when the contents are transferred is a local matter, and therefore, not shown here.
A 19	T-EXCEPTION indication	TS	Layer 5 is informed of an error which occurred between the layer 1 and layer 4, possibly with data loss; the TC is kept existing. Due to this error it is possible that the following TSDU transferred to the layer 5 contains errors or deficiencies.
A 20	S-TDT (EM=0)	TP	A TPDU with TSDU end mark set to 0 is sent to the peer entity and further parts of the TSDU will follow (i.e., segmenting occurs).
A 21	S-TDT (EM=1)	TP	See A 20, but the TSDU end mark is set to 1 (i.e., this TPDU contains a complete TSDU or the last part of a TSDU).
A 22	N-CONNECT response	NS	Affirmative answer to N-CONNECT indication (E 12).
A 23	S-TBR	TP	The called side sends a TBR block to the calling side in order to point to a received failured TPDU. In this case the destination reference can be set to 0.
A 24	S-TCA	TP	Via the NS N-DATA request an NSDU including the transport block TCA is sent to the peer entity.

TABLE B-6/T.70

List of conditions

No.	Name	Description
C 1	Retry	The TC establishment is tried once more.
C 2	No retry	NOT C 1
C 3	TC acceptable	The TC offered by the peer entity is accepted by the layer 4 due to local circumstances.
C 4	TC not acceptable	NOT C 3
C 5	NC acceptable	The NC offered by the layer 3 is accepted by the layer 4 due to local circumstances.
C 6	NC not acceptable	NOT C 5
C 7	EM = 0	TSDU end mark of the TDT block is 0
C 8	EM = 1	TSDU end mark of the TDT block is 1
C 9	Recovery	The terminal provides the TS T-EXCEPTION indication
C 10	No recovery	NOT C 9
C 11	Segmentation	The TSDU received from layer 5 is longer than the negotiated TDT block size and has, therefore, to be segmented and consequently, to be reassembled on the receiver side.
C 12	No segmentation	NOT C 11

ANNEX C

(to Recommendation T.70)

Recommendations for implementation of Recommendation X.21

C.1 *General*

This Annex deals with recommended actions to be taken by a telematic DTE in relation to the receipt of call progress (CP) signals from the network and in relation to the handling of optional user facilities. The adherence to these recommendations is not mandatory in order to conform to Recommendation T.70 but may be of importance for the performance of the DTE.

Telematic terminals are in general assumed to make automatic repeated call attempts and sequential automatic calls to a number of addresses for which the following actions apply.

C.2 *Receipt of call progress signals 01 or 04*

When one of the CPS 01 or 04 is received the DTE should use the timer T3B and wait up to 60s for the completion of the call.

C.3 *Receipt of call progress signal 03*

The DTE should use either timer T3A or T3B in this case, depending on the time the DTE is prepared to wait for the completion of the call. Observe that the queuing time is charged as communication time in some networks.

C.4 *Receipt of call progress signals of the group 2 to 8*

See Table C-1/T.70.

TABLE C-1/T.70

Code group/Code	Delay for reattempts (s)	Number of reattempts	Delay between series of reattempts (s)
2, 6	⩾ 5	⩽ 7	⩾ 60
41, 42, 43, 48 5, 8	⩾ 5	⩽ 1	Reattempts are not recommended
44, 45, 46, 47, 49 7	⩾ 5	⩽ 1	⩾ 600

Note – Some networks charge for call attempts, when the call is unsuccessful due to the condition of the called DTE. Examples of such situations are the receipt of the call progress signals 21 (busy) and 45 (controlled not ready).

ANNEX D

(to Recommendation T.70)

Service definitions and state transition diagrams for the HDLC procedure and the network layer defined for CSPDN

D.1 *Service definitions*

D.1.1 *Physical service used by HDLC*

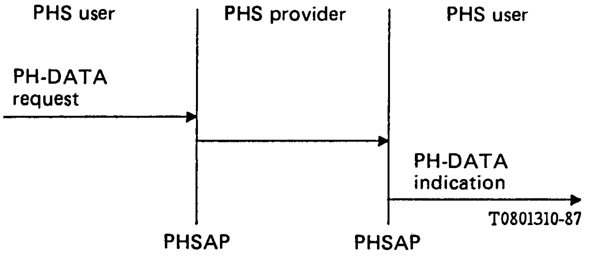


FIGURE D-1/T.70

PH-data-transfer

D.1.2 *Data link service (HDLC)*

D.1.2.1 *Data link connection establishment*

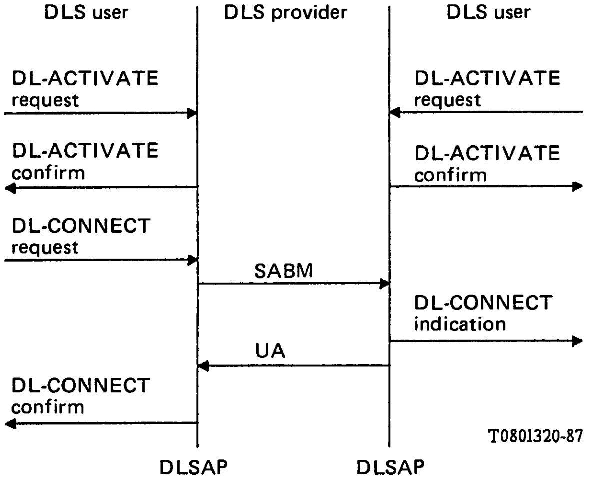


FIGURE D-2/T.70

Successful DLC-establishment

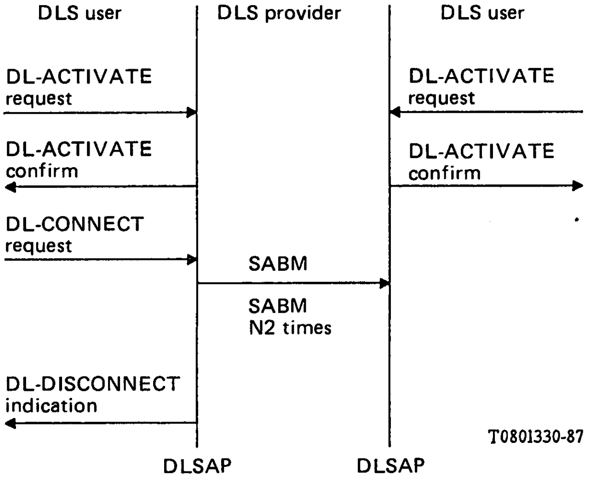


FIGURE D-3/T.70

Not successful DLC-establishment

D.1.2.2 *Data link transfer phase*

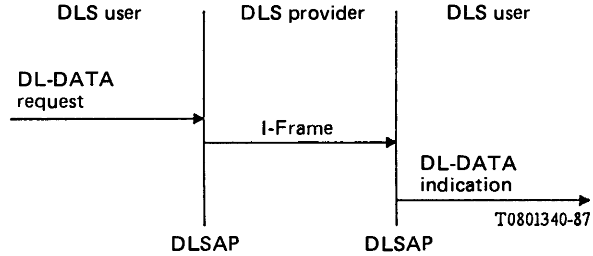


FIGURE D-4/T.70

DL-data transfer

D.1.2.3 *Data link release*

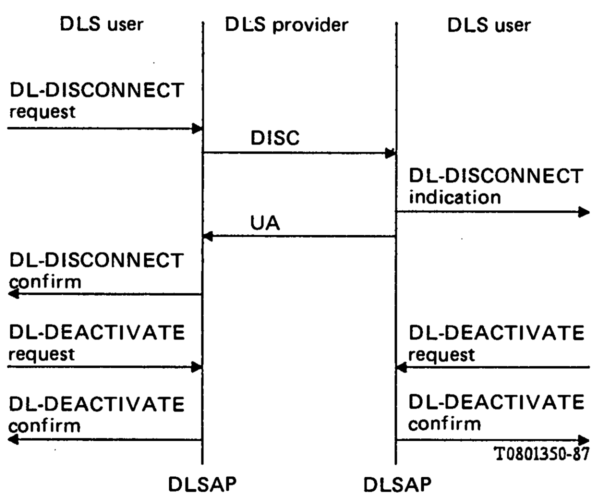


FIGURE D-5/T.70

DL-release initiated by DL-user

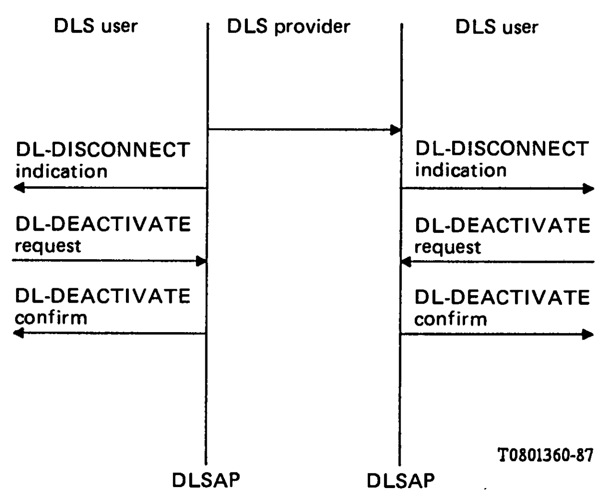


FIGURE D-6/T.70

DL-release initiated by DL-provider

D.1.2.4 *Data link resetting*

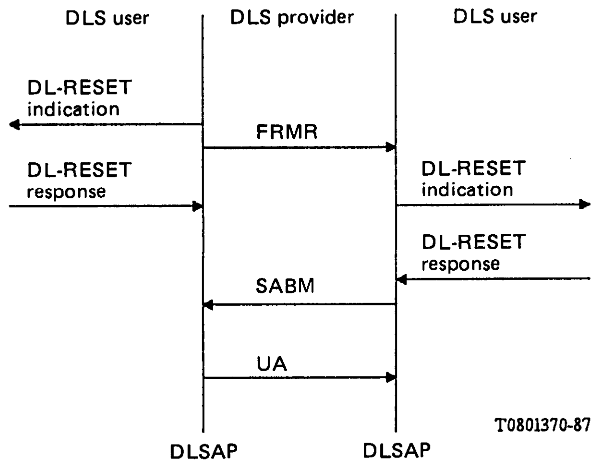


FIGURE D-7/T.70

Successful resetting

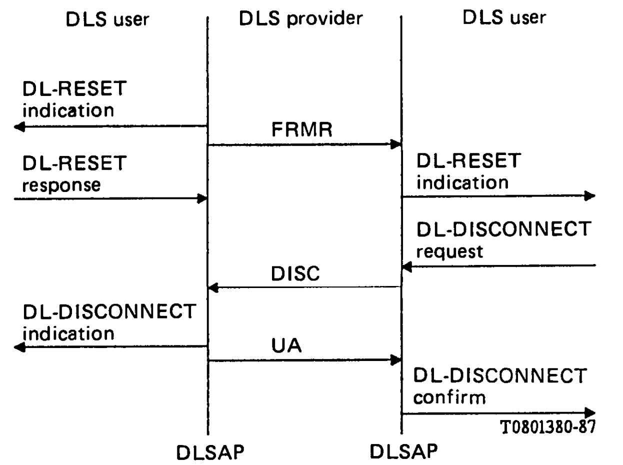


FIGURE D-8/T.70

Resetting not accepted by the receiver of FRMR

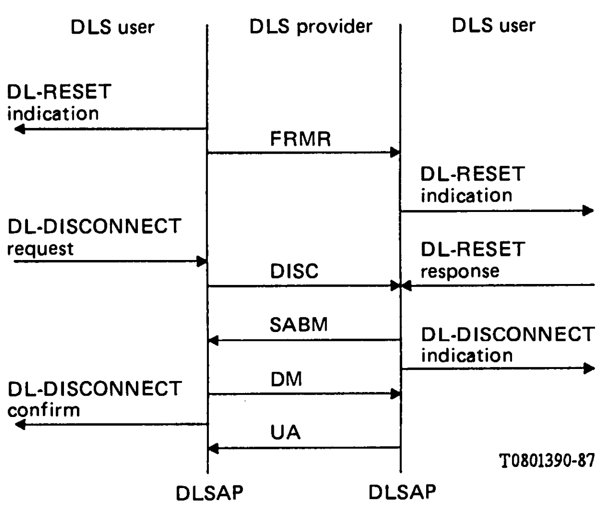


FIGURE D-9/T.70

Resetting not supported by the transmitter of FRMR

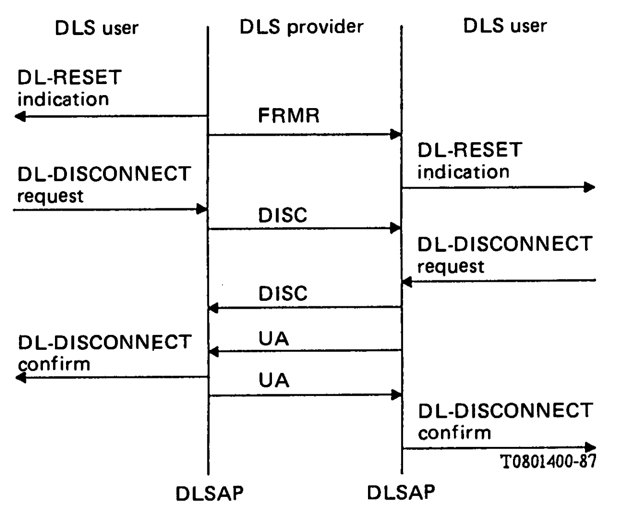


FIGURE D-10/T.70

Resetting not accepted by both

D.2 *State transition diagrams HDLC*

D.2.1 *The relation between the diagrams*

The following diagrams describe the HDLC procedure as one functional unit. The first page comprises the whole protocol and the following page gives the details to specific states.

D.2.2 *Abbreviations*

ABM	Asynchronous balanced mode
ADM	Asynchronous disconnected mode
R:xxx	Receive xxx (command or response)
R:Cxxx	Receive a command
R:Rxxx	Receive a response
S:xxx	Send xxx
F	Final bit
P	Poll bit
XXX	Not this condition
RC	Redrive counter
RCB	Redrive counter busy
IC	I-Frame counter
V_{uu}	Variable for sequence updating

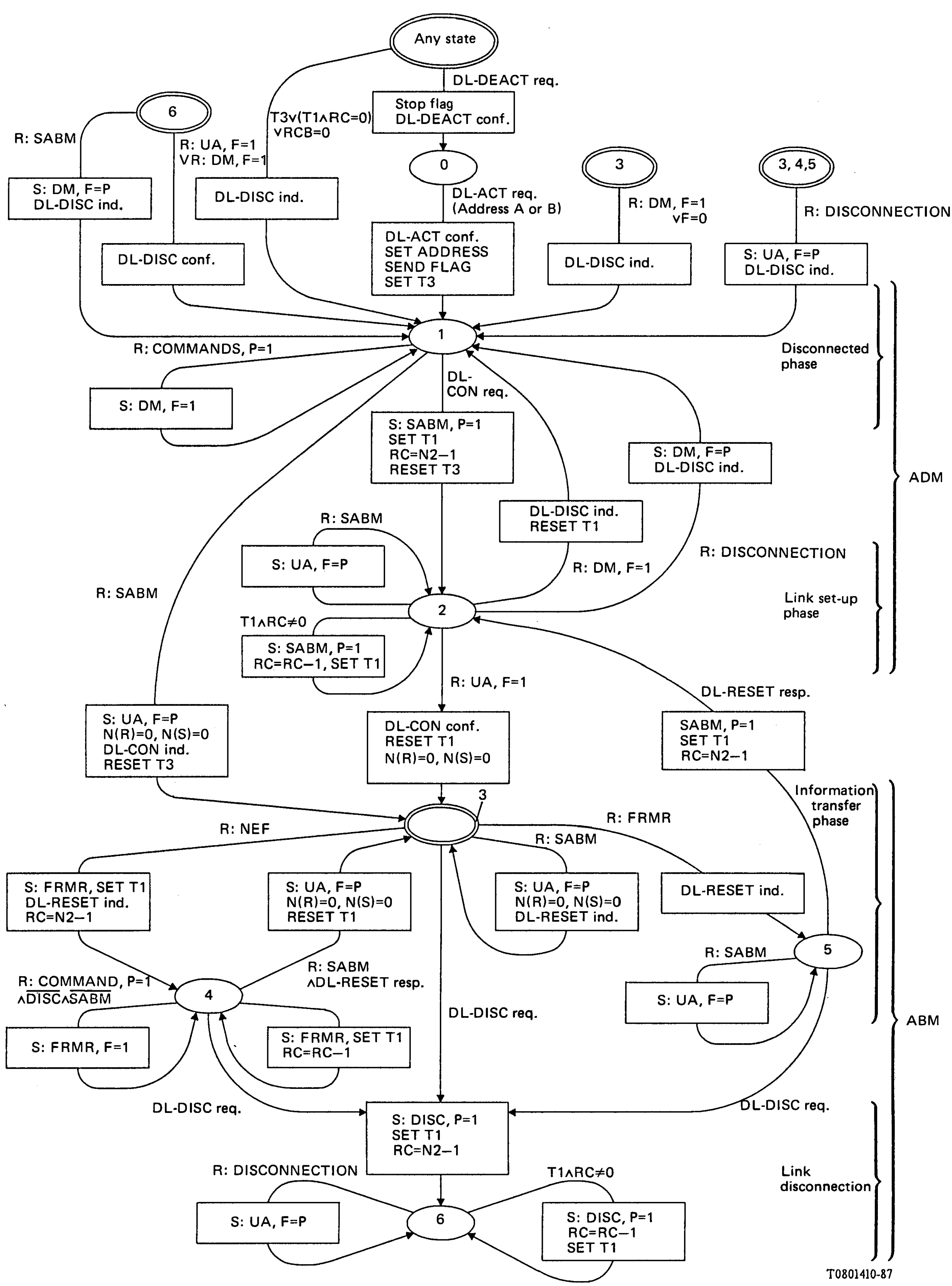
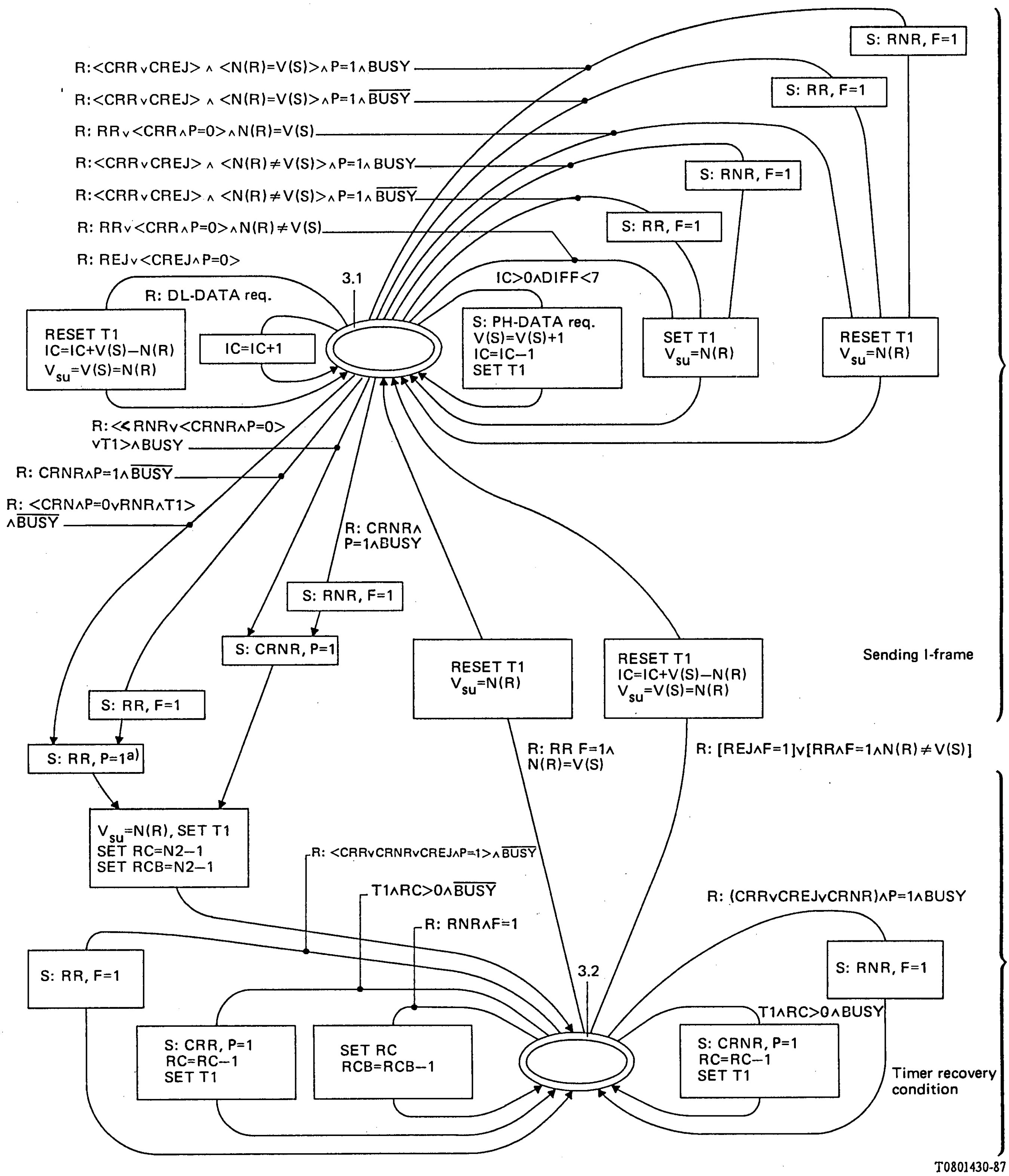


FIGURE D-11/T.70

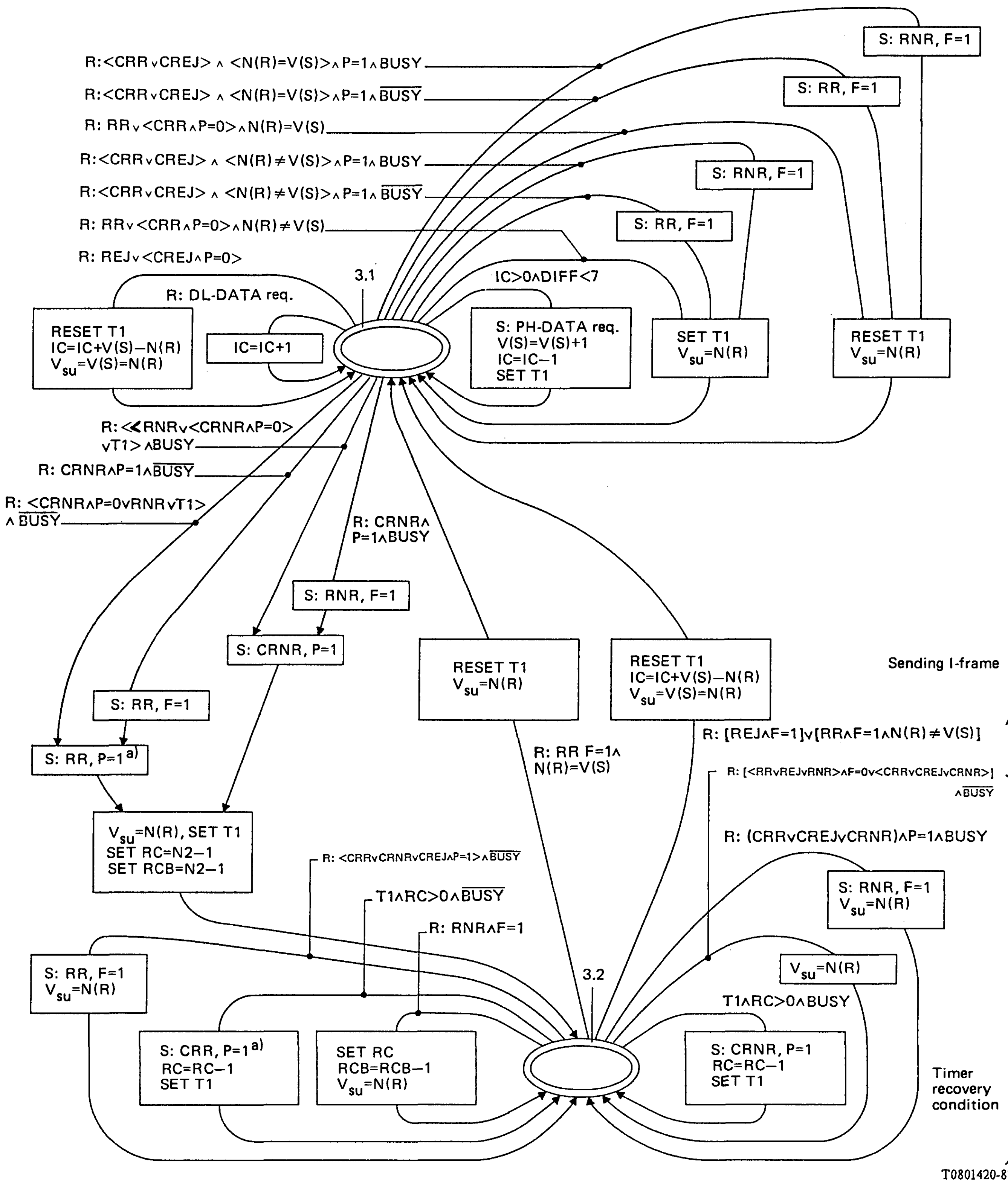
State transition diagram HDLC
(Data link control)



a) Alternatively to RR, P = 1 it is allowed to send PH-DATA req. P = 1 or CREJ, P = 1.

FIGURE D-12/T.70

State transition diagram HDLC
(3 information transfer phase, I-frame control)



a) Alternatively to RR, P = 1 it is allowed to send PH-DATA req. P = 1 or CREJ, P = 1.

FIGURE D-13/T.70

State transition diagram HDLC
(3 information transfer phase, I-frame control with update of N(R)
in timer recovery condition)

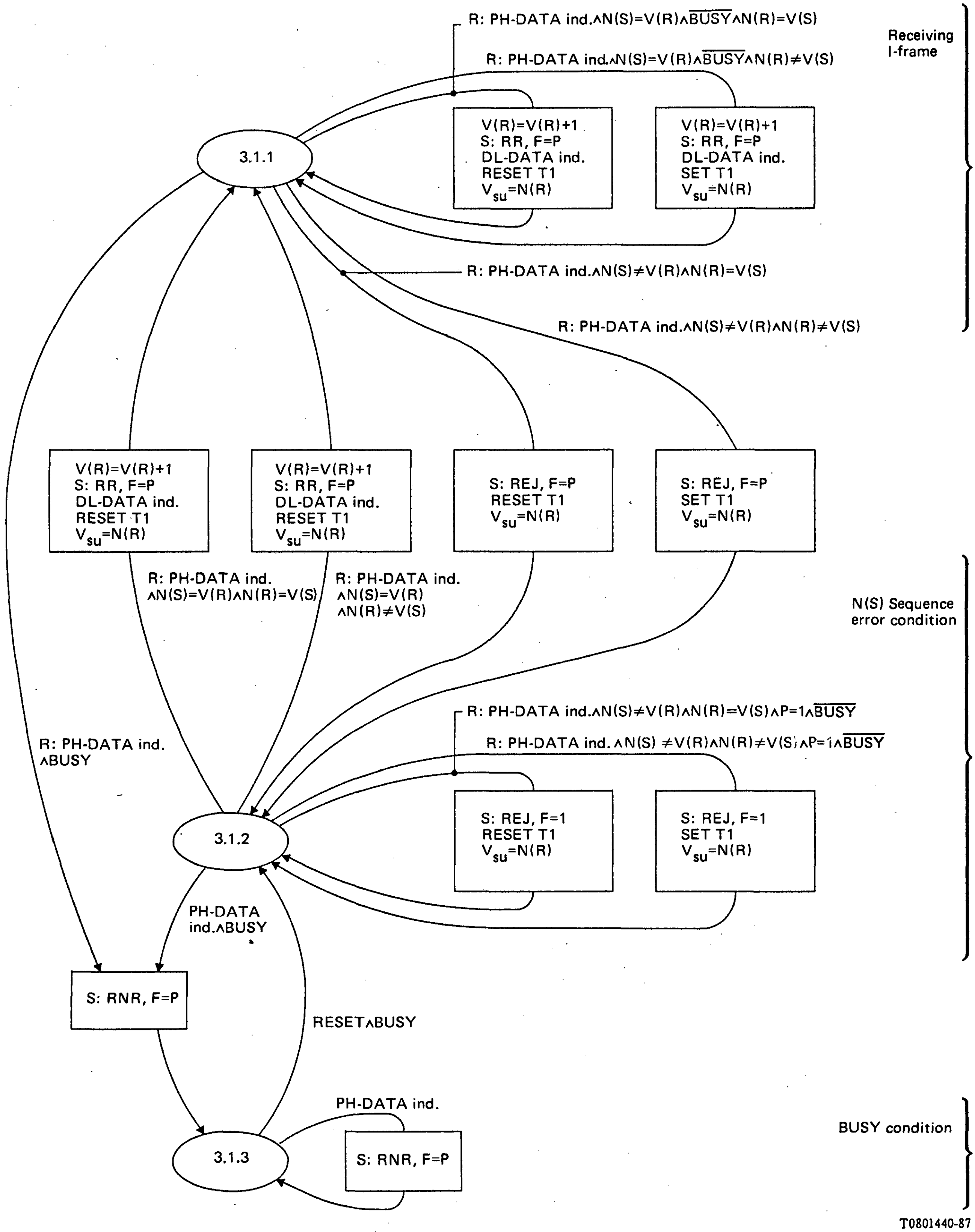


FIGURE D-14/T.70

State transition diagram HDLC
(3.1 information transfer phase, I-frame acknowledgement)

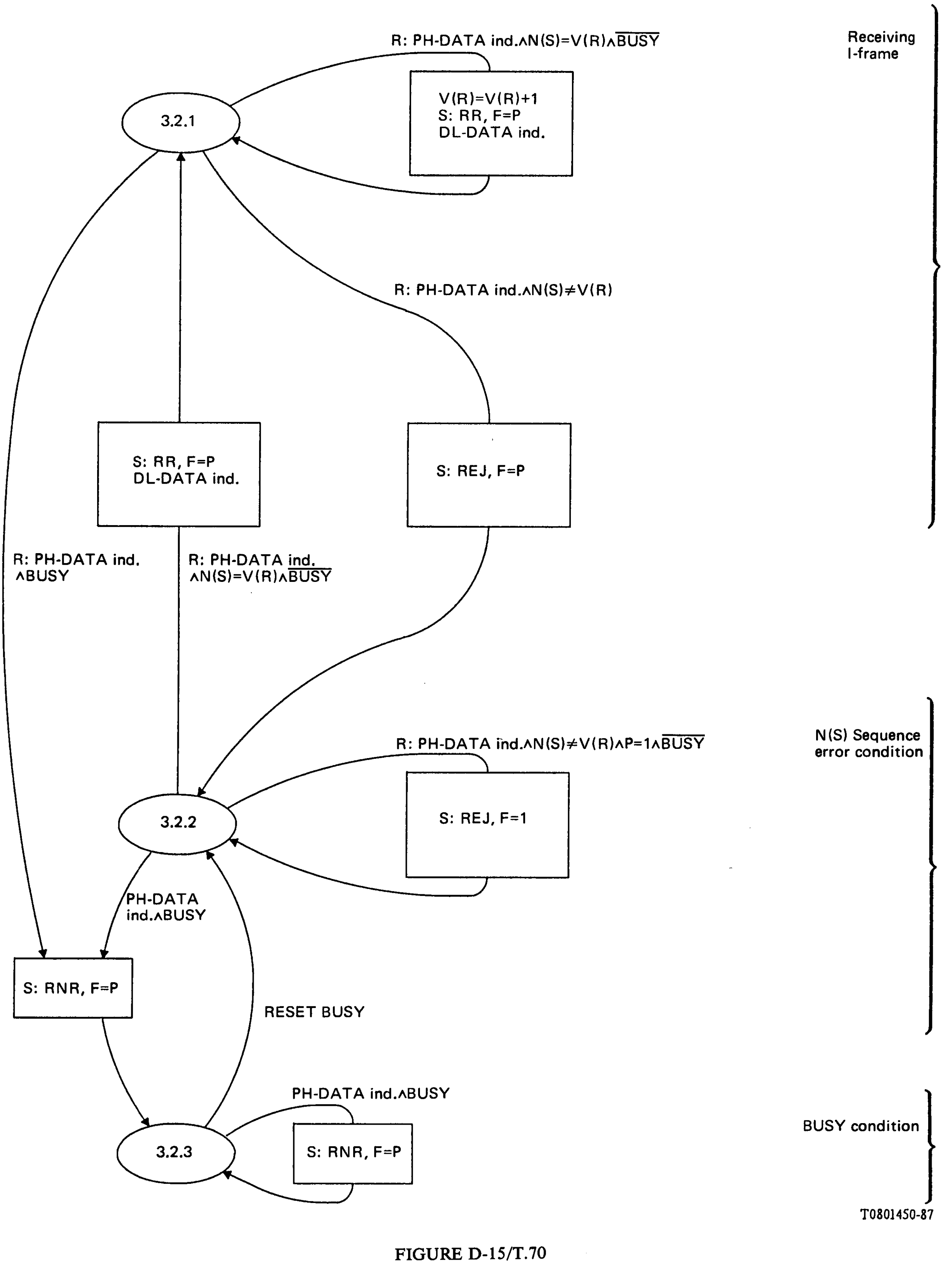


FIGURE D-15/T.70

State transition diagram HDLC
(3.2 information transfer phase, I-frame acknowledgement in exception conditions)

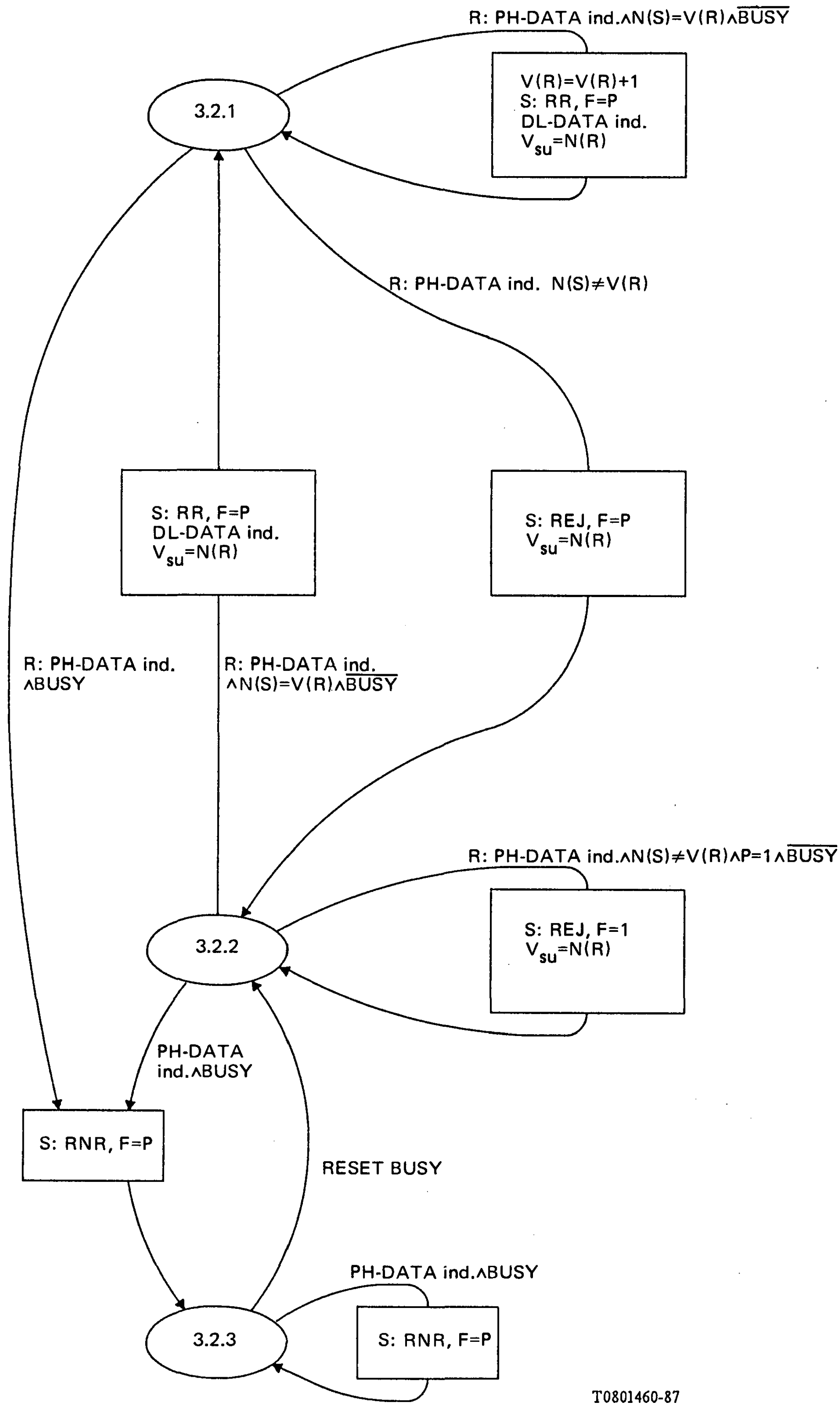


FIGURE D-16/T.70

State transition diagram HDLC
(3.2 information transfer phase, I-frame acknowledgement
in exception conditions with update of N(R))

D.3 *Summary of frame definitions*

D.3.1 *Invalid frame*

- frames not properly bounded by flags;
- frames containing addresses other than A or B;
- frames with frame check sequence (FCS) error;
- frames containing less than 32 bits between blags.

D.3.2 *Valid frames*

D.3.2.1 *Not expected frames*

NEF, not expected frames (for the receiver) which lead to a frame reject condition (excluding frames with a FRMR control field):

– a command or response control field that is undefined or not implemented;	Type W
– a frame with an information field which is not permitted or supervisory or unnumbered frame with incorrect length;	Type X
– an I-frame with an information field which exceeds the maximum established length;	Type Y
– a frame with an invalid N (R).	Type Z

D.3.2.2 *Expected frames*

- frames which must lead to a reaction (in accordance to the Recommendation) by the receiving station;
- frames which must be ignored only in determined states by the receiving station.

D.4 *X.21 service, controlled by the network layer*

D.4.1 *X.21 connection establishment*

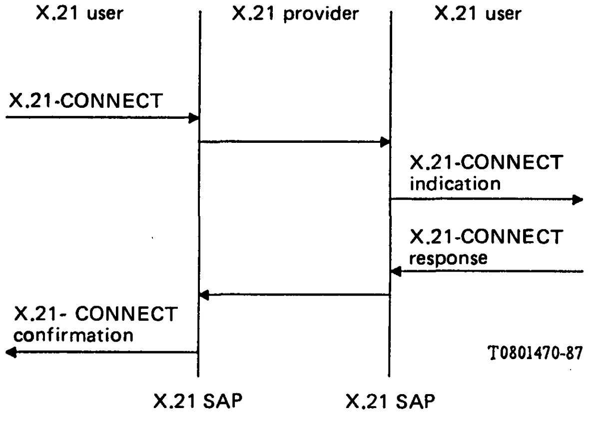


FIGURE D-17/T.70

Successful PHC establishment

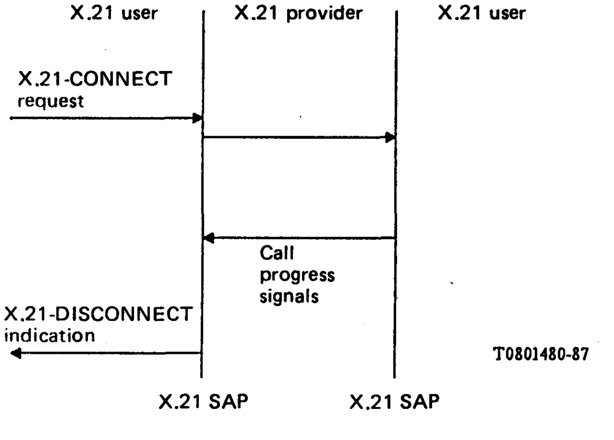


FIGURE D-18/T.70

Not successful PHC establishment

D.4.2 *X.21 connection release*

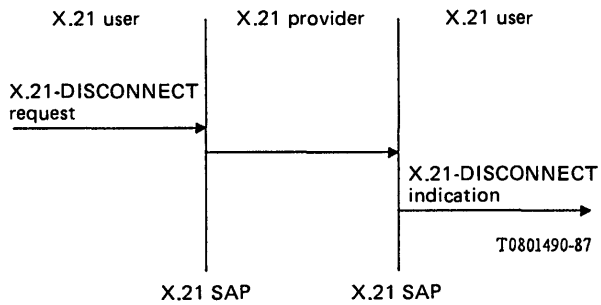


FIGURE D-19/T.70

Disconnection initiated by the X.21 user

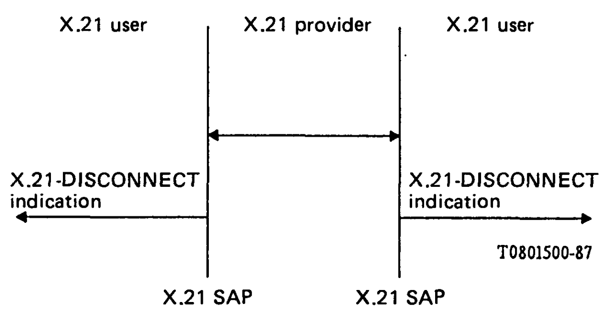
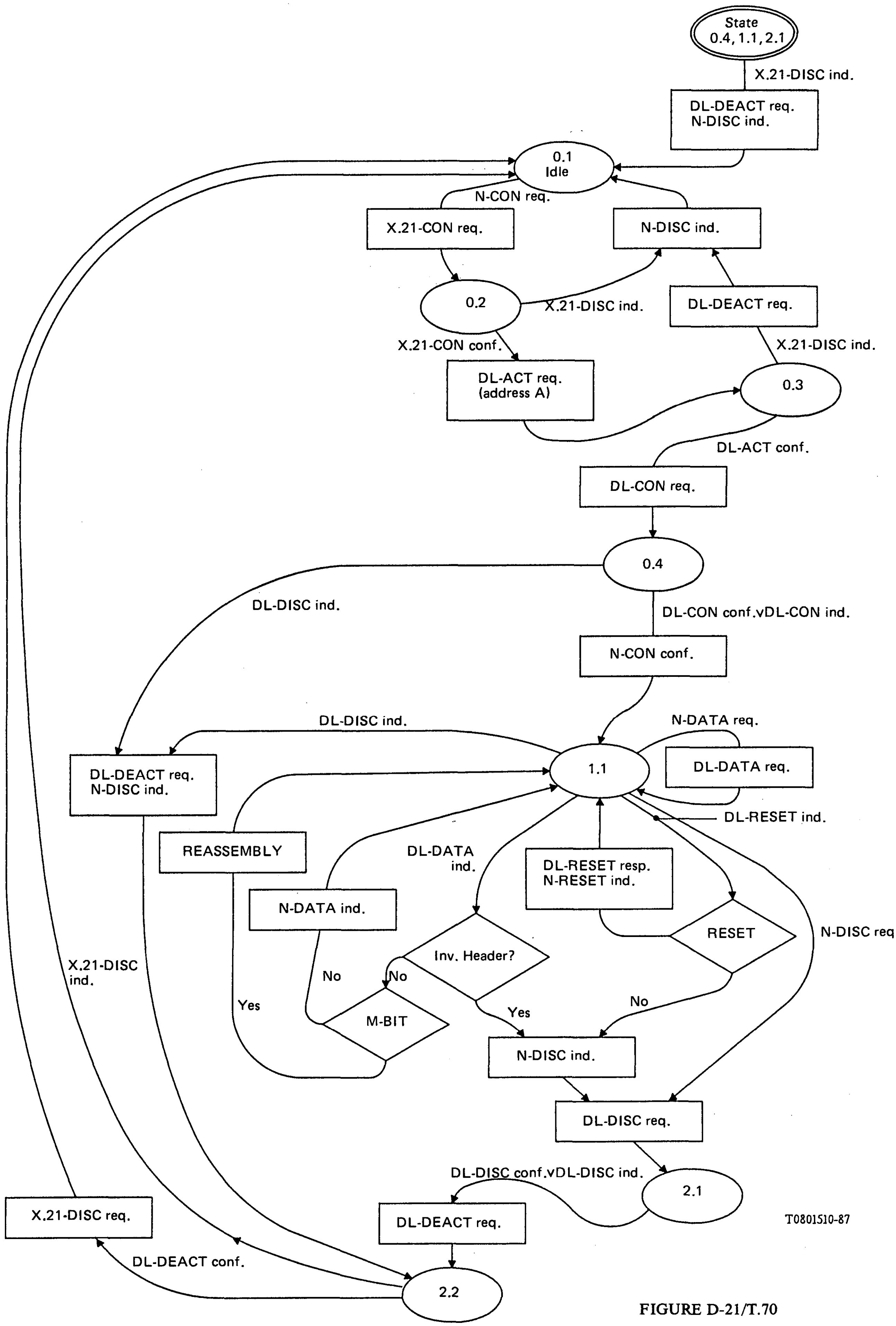


FIGURE D-20/T.70

Disconnection initiated by the X.21 provider



T0801510-87

FIGURE D-21/T.70

State transition diagram for the network layer (calling side)

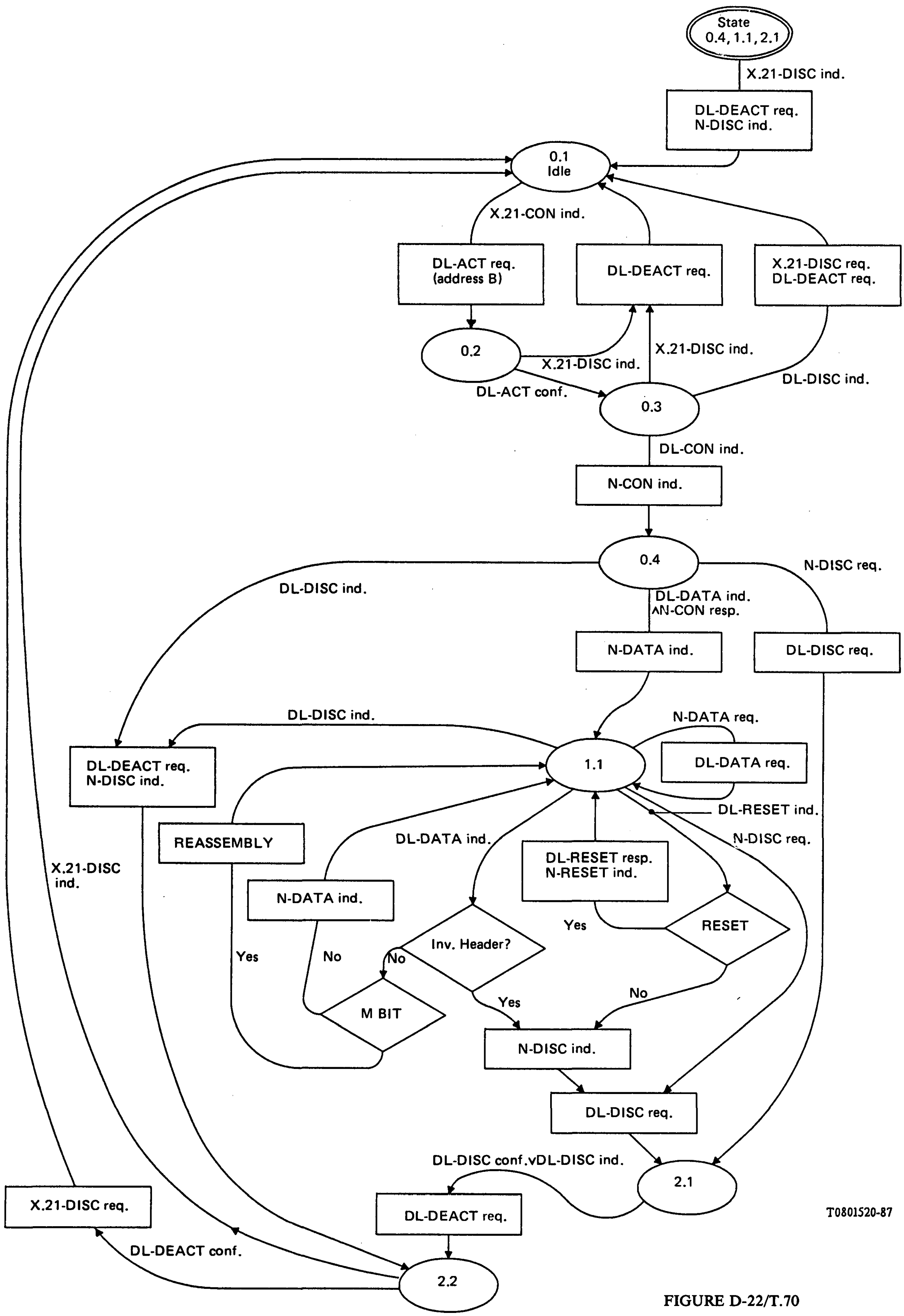


FIGURE D-22/T.70

State transition diagram for the network layer (called side)

TABLE D-1/T.70

Application rules regarding the network protocol data unit (NPDU)

Conditions ↓		Combination of conditions								
		a	b	c	d	e	f	g	h	i
C1	Transmit/receive	T	T	T	T	T	R	R	R	R
C2	NPDU length (octet)	> 2	> 2	> 2	> 2	< 3	> 2	> 2	> 2	< 3
C3	1st octet 01/< >	01	01	01	< >	*	01	01	< >	*
C4	2nd octet bits 1 to 7	0	0	< >	*	*	*	*	*	*
C5	2nd octet bit 8 (M-bit)	0	1	*	*	*	0	1	*	*
Actions/applications rules										
A1	Correct/acceptable	X				X (Note)				
A2	N-DISC ind., DL-DISC req.								X	X
A3	Not allowed		X	X	X	X				
A4	Error case								X	X

Note – The Teletex system has to accept as many NPDUs, that at least, the same number of octets can be received as contained in the maximum negotiable transport block size.

C Condition
A Action/application rule
T Transmit
R Receive
< > Not equal
* Not relevant
X Valid/applicable

LINK ACCESS PROTOCOL BALANCED (LAPB) EXTENDED FOR HALF-DUPLEX PHYSICAL LEVEL FACILITY

(Malaga-Torremolinos, 1984; amended at Melbourne, 1988)

The CCITT,

considering

(a) that the Teletex service will be introduced in different types of networks, i.e. circuit switched public data networks (CSPDN), packet switched public data networks (PSPDN) and public switched telephone networks (PSTN);

(b) that depending on the service provided by the physical level, the link level procedures may have to cater for a half-duplex transmission facility;

(c) that some Administrations are considering the provision of a Teletex service with a half-duplex transmission facility on the PSTN;

(d) that modems according to Recommendation V.26 *bis* are suitable for half-duplex transmission at 2400 bit/s on PSNTs,

unanimously declares

that this Recommendation defines the link level procedure using LAPB extended for half-duplex physical level service.

1 Introduction

1.1 *General*

1.1.1 Figure 1/T.71 shows the half-duplex transmission module (HDTM) for extending the use of LAPB for operation of Teletex terminals connected to the PSTN where use of half-duplex 2400 bit/s modems is planned. This is referred to in Recommendation T.70 as LAPX.

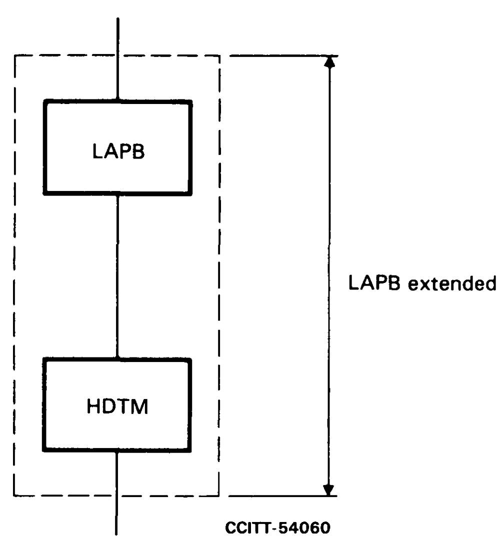


FIGURE 1/T.71

Teletex data link layer for PSTN based on LAPB plus the half-duplex transmission module (HDTM)

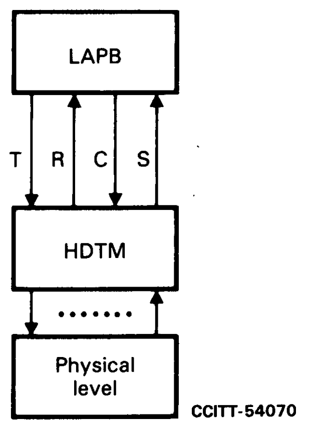
1.1.2 Before the HDTM begins operation the physical circuit must be established by the appropriate PSTN call control procedures. The operation of the HDTM is such that the calling DTE will initially have the right to transmit. For the link addressing conventions refer to Recommendation T.70.

1.2 *Architecture*

1.2.1 *Level relationships*

It is an objective to avoid modification of the definition of LAPB in order to adapt it for half-duplex operation. However, there is a functional requirement that the HDTM inhibit LAPB from sending frames during certain phases of the half-duplex procedure. The means of accomplishing this functional requirement is not defined.

The logical relationships between LAPB, the HDTM and the physical level are as shown in Figure 2/T.71.



T Transmit data functions
R Receive data functions
C Control functions
S Status functions

FIGURE 2/T.71

Level relationships

1.2.2 *Control (C) and status (S) functions*

The following logical functions are defined to describe the interactions between LAPB and the HDTM:

Control <TERM>

– Revert to the HDTM idle state since LAPB has entered the disconnected phase (equivalent to ADM of HDLC).

Status <OP-T>

– LAPB is enabled to send frames.

Status <INOP-T>

– LAPB is inhibited from sending frames.

2 State diagram and descriptions

2.1 *State diagram*

The state diagram shown in Figure 3/T.71 describes the procedure for controlling the right to transmit. The number in each ellipse is the state reference number.

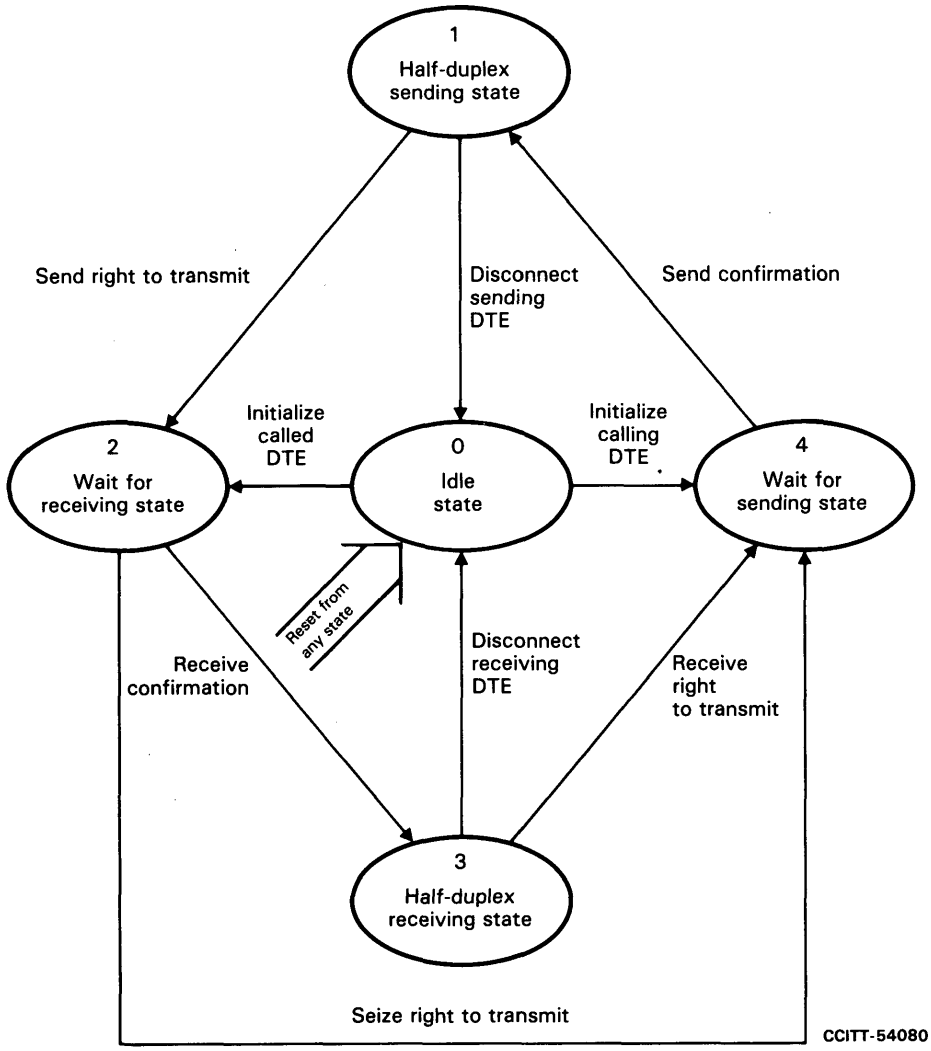


FIGURE 3/T.71

State diagram

2.2 *State definitions*

2.2.1 *Idle state (state 0)*

The DTE is in an inactive state. This is the initial state prior to call establishment and the final state after call termination.

2.2.2 *Half-duplex sending state (state 1)*

The DTE is in a half-duplex sending state, so that all signals generated by LAPB are passed to the physical level.

2.2.3 *Wait for receiving state (state 2)*

The DTE is awaiting indication that the remote DTE has entered the half-duplex sending state. No signals generated by LAPB are passed to the physical level.

2.2.4 *Half-duplex receiving state (state 3)*

The DTE is in a half-duplex receiving state, so that no signals generated by LAPB are passed to the physical level. The remote DTE is considered to be in the half-duplex sending state.

2.2.5 *Wait for sending state (state 4)*

The DTE is awaiting indication of the availability of the physical level for transmission of frames to the remote DTE. All signals generated by LAPB are passed to the physical level, but LAPB is inhibited from sending frames.

2.3 *Table of transitions between states*

Table 1/T.71 shows the events that cause transitions from one state to another, along with any resulting actions. This shows a generalized description of the operation of the HDTM.

2.4 *State definitions expressed in terms applicable to a modem interface*

The following definitions apply to the use of the HDTM with the V.26 *bis* modem interface, as an example.

2.4.1 *Idle state (state 0)*

Circuit 107 is OFF.

2.4.2 *Half-duplex sending state (state 1)*

Circuit 105, circuit 106 and circuit 107 are ON. LAPB is connected to circuit 103 and enabled to send frames.

2.4.3 *Wait for receiving state (state 2)*

Circuit 107 is ON, circuit 105 is OFF. LAPB is inhibited from sending frames and disconnected from circuit 103, which is held in the binary 1 condition. Timer T is running.

2.4.4 *Half-duplex receiving state (state 3)*

Circuit 107 is ON, circuit 105 is OFF. LAPB is inhibited from sending frames and disconnected from circuit 103, which is held in the binary 1 condition.

2.4.5 *Wait for sending state (state 4)*

Circuit 105 and circuit 107 are ON, and circuit 106 is OFF. LAPB is connected to circuit 103 but is inhibited from sending frames.

2.5 *Table of transitions between states expressed in terms applicable to a modem interface*

Table 2/T.71 shows, in terms of the V.26 *bis* modem interface, the events that cause a state transition and the resulting action(s).

2.6 *Timer T*

This timer is used to recover from an apparent failure of the remote DTE to take the right to transmit. To avoid a contention condition during this recovery process, different values of timer T are to be used by the called and calling DTE. A calling DTE uses the value T_a, and a called DTE uses the value T_b.

The values of T_a and T_b are system parameters and must be studied further in relationship to interworking requirements and other system parameters in Recommendation T.70.

TABLE 1/T.71

Description of state transitions

Present state	Event	Action	New state
0	Calling DTE: Data circuit established (e.g. data set ready, ready for data)	→	4
0	Called DTE: Data circuit established (e.g. data set ready, ready for data)	Start timer T	2
4	Indication of availability of the physical level for transmission	Send indication to the remote DTE that the half-duplex sending state has been entered Status <OP-T> (see Note 1)	1
1	Conclusion of transmission	Send request that remote DTE enter the half-duplex sending state (see Note 4) Start timer T Status <INOP-T> (see Note 2)	2
2	Reception of indication that the remote DTE has entered the half-duplex sending state	Stop timer T	3
2	Expiry of timer T	→	4
3	Reception of notification that the remote DTE is requesting a change in the direction of transmission	→	4
1	LAPB has entered a disconnected phase (i.e. Conrol <TERM>, see Note 3)	→	0
3	LAPB has entered a disconnected phase (i.e. Conrol <TERM>, see Note 3)	→	0
Any	Physical level has no circuit to a remote DTE	→	0

Note 1 – Status <OP-T> indicates to LAPB that the sending of frames is enabled.

Note 2 – Status <INOP-T> indicates to LAPB that the sending of frames is inhibited.

Note 3 – Control <TERM> indicates that LPAB has entered the disconnected phase (equivalent to ADM of HDLC).

Note 4 – HDTM uses the idle data link channel state indication (at least 15 contiguous 1's) for requesting that the remote DTE enter the half-duplex sending state.

TABLE 2/T.71

Description of state transitions in terms of the V.26 *bis* modem interface

Present state	Event	Action	New state
0	Calling DTE: Circuit 107 ON	Turn circuit 105 0N Connect LAPB to circuit 103	4
0	Called DTE: Circuit 107 ON	Start timer T	2
4	Circuit 106 ON	Enable sending of LAPB frames (see Note 1)	1
1	Transmission concluded (see Note 2)	Inhibit sending of LAPB frames Disconnect LAPB from circuit 103 Hold circuit 103 in the binary 1 condition Turn circuit 105 OFF (see Note 3) Start timer T	2
2	Reception of a flag	Stop timer T	3
2	Expiry of timer T	Turn circuit 105 ON Release circuit 103 from binary 1 condition Connect LAPB to circuit 103	4
3	Reception of 15 contiguous 1 bits (see Notes 4 and 5)	Turn circuit 105 ON Release circuit 103 from binary 1 condition Connect LAPB to circuit 103	4
1	LAPB has entered a disconnected phase	Turn circuit 105 0FF	0
3	LAPB has entered a disconnected phase	⟶	0
Any	Circuit 107 OFF	Turn circuit 105 OFF	0

Note 1 – It is necessary to ensure that at least one full flag is transmitted after circuit 106 comes ON. This flag may be the opening flag of the first frame.

Note 2 – The HDTM may determine that a transmission by the LAPB module has been concluded by either of the following:

– counting a sequence of contiguous flags on circuit 103 while in state 1,
– a time-out, T,
– a signal from another source, e.g., from a higher level.

However, if no frame is transmitted while in state 1, not less than five contiguous flags shall be sent in state 1 before entry into state 2.

Note 3 – It is recommended that circuit 105 not be turned OFF until 15 bit times after the binary 1 condition is established on circuit 103. This will assure transmission of an idle sequence to the remote DTE.

Note 4 – It is recognized that whether or not an idle sequence is sent by the remote DTE, the DTE will detect an idle sequence after circuit 109 goes OFF, since according to Recommendation V.26 *bis*, this will hold circuit 104 in the binary 1 condition.

Note 5 – It is understood that circuit 109 will go OFF. Entry into state 4 may be made dependent on this OFF condition, as an implementation option.

ANNEX A

(to Recommendation T.71)

Additional rules making for greater efficiency in half-duplex transmission

A.1 *General considerations*

- Greater efficiency is obtained in recovery situations.
- The application of these rules is optional.
- The application of these rules does not imply any incompatibility or entail any amendment of DTEs (or DCEs) which observe the procedures described in Recommendation T.71.

A.2 *Rules of operation*

1) Before the DTE (or DCE) gives the turn back, it ensures that it has acknowledged all the frames received and accepted before it received the turn.
2) If the DTE (or DCE) receives or takes the turn, it will always first retransmit all the I-frames which have not been acknowledged.
3) The DTE (or DCE) must replace the last RR frame in each turn, if any, by an REJ frame carrying the appropriate N(R).

Recommendation T.90

CHARACTERISTICS AND PROTOCOLS FOR TERMINALS FOR TELEMATIC SERVICES IN ISDN

(Melbourne, 1988)

CONTENTS

1 Scope

1.1 *General*

ISDN has been defined to support a wide range of voice and non-voice services, and applications, in the same network based on a multipurpose user-network interface.

This Recommendation describes the requirements for Telematic terminals, developed for ISDN application, and connected to an ISDN specified by the Recommendations of the I-Series.

This Recommendation covers Teletex, Group 4 facsimile, mixed mode and videotex terminals.

Terminal requirements to support other Telematic services are for further study.

Terminals developed for the provision of Telematic services in CSPDNs, PSPDNs and PSTNs, using Terminal adaptors to access the ISDN are not covered by this Recommendation.

Interworking with existing Telematic terminals connected to CSPDNs, PSPDNs and PSTNs, thereby maintaining the Telematic service integrity, should be possible, but is outside the scope of this Recommendation.

1.2 *Use of bearer capability*

This Recommendation is based on the use of bearer capabilities defined for the ISDN, using B-channels for the information transfer and virtual circuit call control and the D-channel for the call control.

The use of both, circuit-switched and packet-switched information transfer modes is defined.

The use of the frame mode information transfer as defined in Recommendation I.122 is for further study.

1.3 *Protocol architecture*

This Recommendation provides the application rules for other CCITT-Recommendations and ISO Standards with particular expansion aiming at the applicability for end-to-end (DTE-DTE) communication through the network as well as DTE-DCE interconnection and the OSI-Network Service.

The use of existing protocols for ISDN Telematic terminals different from those described in § 2, e.g. T.70, CSPDN minimum header is optional.

The optional implementation of more than one type of protocol, and use of the appropriate protocol on a per-call-basis for communication between Telematic terminals, following the protocols described in this Recommendation, and terminals using optional protocols, is the responsibility of the user of such optional protocols.

2 ISDN circuit-switched mode (DTE-DTE communication)

For this mode, the circuit-switched 64 kbit/s unrestricted information transfer capability shall be used.

For additional information regarding connection control, see § A.1 a).

For additional information regarding the information transfer phase, see § A.1 b).

2.1 *Protocol set*

The protocol set applicable to the circuit-switched mode (CS mode) is shown in Figure 1/T.90.

	Physical connection control	Virtual connection control and information transfer
Layer 3	Q.931	ISO 8208 (Note 3)
Layer 2	Q.921	X.75 LAPB (modified) (Note 2)
Layer 1	I.430/I.431 (Note 1)	
	<D-channel>	<B-channel>

Note 1 – See § 2.2.1.

Note 2 – Full-duplex single link procedures are defined as described in § 2.2.3.1. For service definitions and state transition diagrams for the data link layer within the B-channel, see Appendix III.

Note 3 – DTE-DTE conection is specified on the basis of ISO 8208 (September, 1987) as described in § 2.2.5.

The T.70 NL protocol (CSPDN minimum header, T.70 § 3.3) may optionally be supported in addition to the X.25 PLP and used on a per-call-basis. For further information, see Appendix II.

FIGURE 1/T.90

2.2 *Application rules for B-channel circuit-switched mode*

2.2.1 *Layer 1 physical layer interface characteristics*

The physical interface characteristics shall be in accordance with the I-Series Recommendations; I.430 (Basic user-network interface, Layer 1 specifications), I.431 (Primary rate user-network interface, Layer 1 specifications). This layer provides full a duplex transmission capability.

2.2.2 *Connection control phase*

Recommendation Q.921 shall apply.

2.2.3 *Layer 2 information transfer phase*

The link layer procedure shall consist of a fully symmetrical HDLC procedure as defined in Recommendation X.75 for single link operation. The use of other protocols (e.g. LAPD) is left for further study.

2.2.3.1 *Address procedure*

The following describes the application of the link addressing procedures of Recommendation X.75. Link addresses (A and B) shall be assigned dynamically on a per-call basis according to the following rules:

a) the calling terminal shall take address A;

b) the called terminal shall take address B;

c) commands and responses shall be transferred as shown in Figure 2/T.90;

d) A and B addresses are coded as follows:

Address	12345678
A	11000000
B	10000000

Note – The terminal will discard all frames received with an address other than A and B.

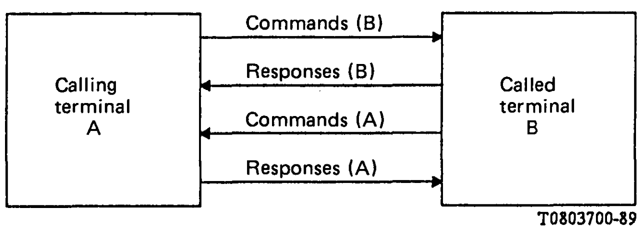


FIGURE 2/T.90

2.2.3.2 *Implementation rules*

In order to achieve full compatibility between different implementations, the rules below for the implementation of Recommendation X.75 shall be followed.

2.2.3.2.1 *General rules*

a) The 1984 version (*Red Book*) of CCITT Recommendation X.75, § 2, shall be used as the reference specification.

b) The term "STE" shall be read as "DTE".

c) At present the non-extended mode of operation (modulo 8) and the extended mode of operation (modulo 128) are defined and in use.

   The desire to improve the efficiency of satellite transmissions and the evolution towards the use of LAPD (modulo 128 only) in layer 2 of the B-channel is expected to lead to the use of modulo 128 as the common base modulo. However, the use of modulo 8 may be allowed.

   To facilitate interworking between terminal equipments using modulo 8 and 128 respectively a procedure based on, e.g. a negotiation mechanism using a low layer. Compatibility check between the endpoints should be defined. This is for further study.

d) Only the single link procedure (SLP) shall be used.

2.2.3.2.2 *Specific rules*

The following rules refer to the indicated sections and tables of Recommendation X.75.

a) *Table 1/X.75*

I-frames should not be sent with an empty I field
$N \geqslant 0$ and $N \leqslant N1\text{-}32$
received empty I-frames shall be treated as valid I-frames.

b) *§ 2.3.4.9*

Sub-paragraphs 5), 6) and 7) are not valid (shall not result in the sending of an FRMR). Instead the following actions shall be implemented:

- Not expected supervisory frames with the F bit set to 1 shall be ignored.
- Not expected UA or DM response shall be ignored.
- Frames with an invalid N(S) shall be responded to by sending REJ (see § 2.3.5.2.1 of Recommendation X.75).

Frames with an FRMR cntrol field shall not be responded by sending of an FRMR.

c) *Table 7/X.75*

Bits W, X, Y and Z set to 0 indicate that no reason for frame rejection is given.

d) *§ 2.3.5.3*

The DTE and the ISDN are not octet aligned and the last paragraph is therefore not valid.

e) *§ 2.3.5.5*

Higher layers should be notified when timer T3 expires (excessive idle state).

f) *§ 2.4.3*

Related to the first paragraph, read instead of "next response" "corresponding response".

g) *§ 2.4.4.1*

In the active channel state, the DTE shall transmit contiguous flags independent of the other DTE.

The calling DTE shall initiate the link by sending an SABM command with the P-bit set to 1.

h) *§ 2.4.4.4.1*

A condition for entering the disconnected phase is also that no unacknowledged DISC command exists, because of collision cases (see Recommendation X.75, § 2.4.4.5).

In the disconnected phase, it is the calling DTE which may initiate link set up.

i) *§ 2.4.5.9, fourth paragraph*

If a RNR is received, the DTE shall remain in the timer recovery condition (because the other DTE is still in the busy condition).

j) *§ 2.4.5.9, fifth paragraph*

If an RNR is received, the DTE shall not resume I-frame transmission or retransmission.

k) *§ 2.4.5.9, last paragraph*

If the transmission attempt variable is equal to N2, the DTE shall enter the disconnected phase.

l) *§ 2.4.7.3*

In the frame rejection condition, the DTE shall only check the commands and react with an FRMR according to the P-bit.

The frame rejection condition is cleared when the DTE receives an SABM, or, receives or transmits a DISC command.

m) *§ 2.4.7.3, second paragraph*

Only the DTE which caused the FRMR condition may try to reset the link.

n) *§ 2.4.7.3, third paragraph* (see Note 1)

After N2 attempts to get the other DTE to reset the link, the DTE shall enter the disconnected phase.

o) *§ 2.4.8.1* (see Note 2)

The timer T1 shall be started at the end of frame transmission. The value of T1 depends on the data signalling rate, the frame length, the value of N2, and a fixed time representing both T2 and the transmission delay [see item r)].

A value is recommended between 2.5 and 7 seconds. Consideration of a specific value requires further study.

p) *§ 2.4.8.2* (see Note 2)

$T1 > T2$
$T2 < 1$ second

Depending on the acknowledgement strategy used, the DTE designer may regard T2 as a design parameter only, in which case the DTE is not obliged to implement a corresponding timer.

q) *§ 2.4.8.3, second paragraph*

$T3 \leqslant 60$ seconds
$T3 \geqslant 30$ seconds

r) *§ 2.4.8.4*

$N2 \geqslant 60$ seconds $\div$ T1

s) *§ 2.4.8.5*

$N1 = 2112 + (n \times 1024)$ bits;
$n = 0$ or 2 or 6 or 14.

t) *§ 2.4.8.6* (see Notes 2,3)

$k = 7$

Note 1 – It is not meaningful to reset the link if the other DTE is not responding for $N2 \times T1$.

Note 2 – The acknowledgement strategy used by the receiving DTE should be independent of any knowledge about the value of k used by the sending DTE. This can be achieved by either acknowledging every correctly received I-frame as soon as possible or by implementing an acknowledgement timer, i.e., a T2 timer as defined above [see item p)].

Note 3 – Further study is needed on a mechanism for negotiation of k.

2.2.4 *Layer 3 – connection control phase*

Recommendation Q.931 shall apply. All encodings should be derived from the relevant section in Q.931.

Three information elements (IEs) are of particular interest to terminals accessing Telematic services. See Annexes B and M of Recommendation Q.931 for further information.

- Bearer capability (BC) information element. The BE IE is used to carry information of interest to the bearer service providing network. The BE IE is required to be generated by the calling side, and must be examined by the called side.
- Low layer compatibility (LLC) information element. The LLC IE is used to carry information about protocols at and below the network layer of interest only to the two endsystems. The LLC IE shall be generated by the calling side, and should be examined, if present, by the called side.
- High layer compatibility (HLC) information element. The HLC IE is used to carry information between the endsystems related to protocols above the network layer. The HLC IE shall be generated by the calling side, and should be examined, if present, by the called side.

Fields in bearer capability (BC), low layer compatibility (LLC), and high layer compatibility (HLC) information elements (IE) to be conveyed at the S/T reference point of the user-network interface during the call establishment phase, shall be set to the values defined below.

2.2.4.1 *Bearer capability (BC)*

a) Mandatory fields, to be set to the fixed values (the value to be set is given in parentheses following each field description):

   - Coding standard – octet 3 (CCITT standardized coding, as defined below).
   - Information transfer capability – octet 3 (unrestricted digital information, see Note).
   - Transfer mode – octet 4 (circuit mode).
   - Information transfer rate – octet 4 (64 kbit/s).

b) Fields not required in the default case (these may be explicitly encoded):
   - Structure – octet 4a.
   - Configuration – octet 4a.
   - Establishment – octet 4a.
   - Symmetry – octet 4a.

c) Fields to be omitted due to no necessity:
   - All other fields.

Note – The selection whether to use unrestricted or restricted information transfer capability is out of the scope of this Recommendation.

2.2.4.2 *Low layer compatibility (LLC)*

The LLC IE shall be encoded as follows:

a) Fields to be set to fixed values (the value to be set is given in parentheses following each field description).

Details of coding points and the relevant encodings are for further study.

2.2.4.3 *High layer compatibility (HLC)*

The HLC IE shall be encoded as follows:

a) Fields to be set to fixed values (the value to be set is given in parentheses following each field description).
   - Coding standard – octet 3 (CCITT standardized coding, as defined below).
   - Interpretation – octet 3 (first high layer characteristics identification to be used in the call).
   - Presentation method of protocol profile – octet 3 (high layer protocol profile).

b) Fields with variable content:
   - High layer characteristics identification – octet 4 (e.g., Facsimile Group 4, Teletex).

To maximize the usefulness of HLC checking:

1) the calling telematic terminal shall select the HLC element according to the type of document to be tranferred;

2) the called terminal holds a list of HLC elements describing its receiving capabilities. It will accept an HLC element corresponding to any one of these.

This scheme is illustrated in Table 1/T.90.

2.2.5 *Layer 3 – virtual connection control and information transfer*

ISO 8208 (1987) shall apply.

Note – This protocol, based on the 1984 version of Recommendation X.25, is partially expanded in order to include DTE-DTE application. In particular, the following sections of ISO 8208 are referred:

- § 3.2: Differences in DTE/DTE and DTE/DCE operation
- § 3.3: Operating over circuit-switched connections
- § 4.5: Determining "DTE" or "DCE" characteristics

In addition, the following points should be noted when using this protocol.

a) Calling DTE shall send a RESTART REQUEST packet, begin the restart procedure, and establish virtual circuits. See § 3.3 of ISO 8208.

b) The qualifier bit in data packets should always be set to "0".

c) The delivery confirmation bits in all packets should be set to "0".

d) Normal X.25 reset procedures will apply.

e) Each control block or data block of the transport layer shall be carried in a complete data packet sequence.

f) The terminal should not send a DTE REJECT packet.

g) In case of Group 4 facsimile and Teletex, terminals shall use a specific protocol identifier within CALL REQUEST/INCOMING CALL packets. This identifier is represented by the first octet of the call user data field (remaining octets, if any should be ignored) as shown below:

bit	87654321
octet	00000010

The use of this protocol identifier for Videotex is for further study.

TABLE 1/T.90

Use of HLC codes by various Telematic terminals

Telematic service terminals	HLC codes	
	Sent from calling terminals (Notes 2, 3)	Accepted by receiving terminals (Note 4)
Teletex basic	Basic Teletex	Basic Teletex
Teletex mixed mode	Basic Teletex Mixed mode (Note 1)	Basic Teletex Mixed mode
Group 4 facsimile class 1	Group 4 facsimile	Group 4 facsimile
Group 4 facsimile class 2	Group 4 facsimile	Group 4 facsimile Mixed mode Basic Teletex
Group 4 facsimile class 3	Group 4 facsimile Mixed mode Basic Teletex (Note 1)	Group 4 facsimile Mixed mode Basic Teletex

Note 1 – In case that the calling terminal is Teletex, Mixed mode or group 4 facsimile class 3, only one element shall be sent depending on originating document type.

Note 2 – For multi-service telematic terminals sending more than one document in the same call, the HLC shall indicate the maximum requirement for that call.

For example, when sending a Teletex and a Mixed mode document, the Mixed mode HLC element shall be sent.

Note 3 – When the calling terminal only wishes to receive a document from a called terminal (polling), it shall know in advance the type of document it expects to receive in order to send the appropriate HLC element.

Note 4 – Appendix I provides additional information in order to cater for cases where calls for facsimile equipment are incoming from networks not able to convey HLC information.

2.2.6 *Layer 3 – packet size (NPDU block length)*

The rules for packet size negotiation are given in § 15.2.2.1.1 of ISO 8208. The values for this Recommendation are restricted to 256, 512, 1024 and 2048 octets.

3 ISDN packet switched mode (DTE-DCE communication)

3.1 *Protocol set*

The protocol set applicable to the packet-switched mode (PS mode) is shown in Figure 3/T.90.

	Access connection control	Virtual connection control and information transfer
Layer 3	Q.931	X.25 PLP
Layer 2	Q.921	X.75 LAPB
Layer 1	I.430/I.431 (Note)	
	<D-channel>	<B-channel>

Note – See § 2.2.1.

FIGURE 3/T.90

3.2 *Application rules for B-channel packet-switched mode*

3.2.1 *Layer 1 – physical layer interface characteristics*

See § 2.2.1.

3.2.2 *Layer 2 – link layer procedure*

Recommendation X.31 shall apply, so that the applied protocols are as follows:

- Connection control is to be achieved using Recommendation Q.921 in the D-channel.
- Virtual connection control and information transfer is to be achieved using Recommendation X.25 LAPB in the B-channel.

3.2.3 *Layer 3 – network layer procedure*

Recommendation X.31 shall apply, so that protocols to be applied and application rules are as follows.

3.2.3.1 *Connection control phase*

Recommendation Q.931 and the packet layer protocol of Recommendation X.25 shall apply.

Fields in the bearer capability (BC) information element (IE) to be conveyed at the S/T reference point of the user-network interface during the call establishment phase, shall be set to the values defined below.

Recommendation Q.931 shall apply. All encodings should be derived from the relevant paragraph in Q.931.

- Bearer capability (BC) information element. The BC IE is used to carry information of interest to the bearer service providing network. The BC IE is required to be generated by the calling side, and must be examined by the called side.

3.2.3.1.1 *Bearer capability (BC)*

a) Mandatory fields, to be set to the fixed values (the value to be set is given in parentheses following each field description):

- Coding standard – octet 3 (CCITT standardized coding as defined below).
- Information transfer capability – octet 3 (unrestricted digital information – Note).
- Transfer mode – octet 4 (packet mode).
- User information layer 1 protocol – octet 5 (CCITT standardized rate adaption Recommendation X.31 HDLC flag stuffing).
- User information layer 2 protocol – octet 6 (CCITT Recommendation X.25, link level).
- User information layer 3 protocol – octet 7 (CCITT Recommendation X.25, packet layer).

b) Fields not required in the default case (these may be explicitly encoded):

- Structure – octet 4a.
- Configuration – octet 4a.
- Establishment – octet 4a.
- Symmetry – octet 4a.

c) Fields to be omitted due to no necessity:

- All other fields.

Note – The selection whether to use unrestricted or restricted information transfer capability is out of the scope of this Recommendation.

The high layer compatibility information element (HLC) is not used in PS mode. The use of the HLC in future evolutions of the ISDN packet mode service is for further study.

The low layer compatibility information element (LLC) is not used in PS mode. The use of the LLC in future evolutions of the ISDN packet mode service is for further study.

3.2.3.2 *Virtual connection control and information transfer*

Recommendation X.25 packet layer protocol applies. Item b) and items d) through g) of the application rules specified in § 2.2.5 apply.

4 Provision of the OSI network service (OSI NS)

4.1 *Rationale for considering the OSI NS*

The evolution and realization of the bearer services and teleservices in the ISDN environment and the recognized protocol base within the CCITT directs – as far as the network layer of the communication archichecture is concerned – to the use of the OSI NS. In order to lay the grounds for the integrity of the services under these conditions, the application rules for the network layer protocol (see Note) need to be defined correctly.

Note – In the ISDN circuit-switched mode, support of the OSI NS is provided entirely by the B-channel X.25 packet layer protocol, and is available once the ISDN call has been connected by other means. The provision of the OSI NS is for further study.

4.2 *Architecture/available ISO Standards and CCITT Recommendations*

Because of the structure of the ISDN which makes use of protocol stacks different for connection control and information transfer, the OSI NS may be provided in different ways. The approach using the network layer protocol in the B-channel is based in principle on

- CCITT Recommendation X.213;
- OSI 8208;
- OSI 8878.

The use of the D-channel (Recommendation Q.931) or the relevant protocols defined for future packet oriented information transfer modes (see Recommendation I.122) for the provision of the OSI NS is for further study.

4.3 *Requirements for the OSI NS*

To balance the expenditure for the development of Telematic terminals under the consideration of the OSI NS, the requirements may be limited to the necessary minimum.

This can be obtained by providing the capability of terminating, in the case of an incoming call for both the circuit-switched (CS) and packet-switched (PS) cases, the layer 3 protocol to provide the obligatory functions of the OSI NS only, and in at least a minimum way, in order to appear to the calling terminal to be an OSI terminal at layer 3. In the case of an outgoing call, calling terminals can initiate an OSI communication, as long as all the relevant facilities are supported, if necessary at any time.

4.3.1 *Minimum requirements for the OSI NS*

Table 2/T.90 shows the list of the X.25 PLP optional user facilities which are proposed for the use in relation to the OSI NS in this Recommendation.

TABLE 2/T.90

X.25 PLP optional user facilities

Optional user facility [a]		Used for support of an incoming call [b]	Used for support of an outgoing call
13.13 [c]	Throughput call negotiation	Yes	Optional [d]
13.16 [c]	Fast select	Yes	Optional [d]
13.28 [c]	Transit delay selection and indication (TDSAI)	Yes	Optional [d]
14.1 [c]	Calling address extension	Yes	Optional [d]
14.2 [c]	Called address extension	Yes	Optional [d]
14.3 [c]	Minimum throughput class negotiation	Yes	Optional [d]
14.4 [c]	End-to-end transit delay negotiation (EETDN)	Yes	Optional [d]
14.5 [c]	Expedited data negotiation	Yes	Optional [d]

[a] As the D-bit is always set to 0 in the circuit-mode case, the receipt confirmation selection requirement is satisfied for this case.

[b] To fulfill at least the minimum functionality of the OSI NS (clarification found if necessary in § 4.3.2).

[c] Refers to the relevant section in ISO 8208.

[d] May optionally be invoked for a Telematic communication. These must be supported if initiating a communication with an OSI terminal.

4.3.2 *Minimum functionality when receiving a call from a system using the OSI NS*

The following text represents a possible way to achieve the minimum functionality when receiving a call from a system using the OSI NS. (Refer to both ISO 8878 and ISO 8208.)

13.13 *Throughput class negotiation*: When replying to INCOMING CALL/CALL REQUEST, a throughput class facility request need not be made in the CALL ACCEPTED packet. If a throughput class facility request was not made in the CALL ACCEPTED packet, then this would mean that the throughput classes applying to the call would be those that have been indicated in the INCOMING CALL/CALL REQUEST packet.

13.16 *Fast selection* shall be supported for the full OSI NS (full 128 octets of NS-user data available). The receipt of a CALL REQUEST packet which does not have the value "02" in the first octet of the call user data field would be considered an error [connection rejection – reason unspecified (permanent condition)] by a Telematic terminal which only supports a minimum functionality (see Note). Receipt of a CALL REQUEST packet which does have the value "02" in the first octet of the call user data field indicates Telematic service operating according to Recommendation T.70 (layer 4 only).

13.28 Transit delay selection and indication (TDSAI): This must be accepted when received. However, if the reply to be coded in the "cumulative transit delay subfield" of the EETDN facility is "unknown" (i.e., FF hexadecimal) then the value in the TDSAI field could be ignored.

14.1 Calling address extension facility : This must be accepted when receiving a call.

14.2 Called address extension facility : This must be accepted when receiving a call.

14.3 Minimum throughput class negotiation : if a terminal reacts to the appearance of a throughput class facility request in the INCOMING CALL packet by not sending a throughput class facility request in the CALL ACCEPTED packet, then the minimum throughput class negotiation facility could be ignored.

14.4 End-to-end transit delay notification (EETDN) : When replying this could contain the value "unknown" (i.e., FF hexadecimal).

14.5 Expedited data negotiation : This is used to negotiate the non-use of expedited data (must be used in the CALL ACCEPTED packet).

Note – The use of the value "02" in the ISDN circuit-switched mode is questioned, as there is already coding to indicate Telematic services in the HLC information element.

5 Additional X.25 optional user facilities

In addition to the facilities mentioned in § 4 that should be supported by Telematic terminals in order to comply with the OSI NS, additional facilities/functionalities must be supported as a consequence of:

- the use of Recommendation X.25 PLP for the provision of the OSI NS (this protocol allows layer 3 multiplexing and flow control);
- the provision of various Recommendation X.25 originated user facilities;
- the provision of various service oriented user facilities by some networks (i.e., additional facilities) or by all networks (i.e., essential facilities) as defined by Recommendation X.2.

There is no need for the provision of the additional service oriented user facilities in the circuit-switched case. The X.25 originated user facilities may be used in the circuit-switched case.

5.1 *Categories of additional functionalities* (see Note)

- *X.25 originated user facilities*

  13.1 On-line facility registration

  13.12 Flow control parameter negotiation

- *Service oriented user facilities (network based)*

  13.14 Closed user groups (CUG) selection

  13.14 CUG with outgoing access selection

  13.18 Reverse charging

  13.21 Network user identification

  13.22 Charging information

  13.23 RPOA selection

  13.26 Called line address modified notification

  13.27 Call redirection notification

Note – D-bit modification is not supported.

5.2 *Functionalities*

- *X.25 originated user facilities*

  1) *On-line facility registration*

     The use of this facility shall be restricted to the modification of the range of logical channels. For the default values, Telematic terminals support a single two-way logical channel (i.e., LTC = HTC = 1, LIC = HIC = 0, LOC = HOC = 0).

  2) *Flow control parameter negotiation*

     Packet size and window size parameters may be negotiated. They should use only default values:

     2048 octets for the packet size, seven for the window size. When the parameter negotiation is indicated in an INCOMING CALL packet, they shall respond properly in the CALL ACCEPTED packet.

     Note – Since the maximum TPDU length is 2048 octets, and segmentation should also be avoided, the maximum default length of layers 3 and 2 should be larger than 2048 octets.

- *Service oriented user facilities (network based)*

  1) *Closed user group selection* (Essential in Recommendation X.2) *and CUG with outgoing access selection* (Additional in Recommendation X.2) (13.14).

     These facilities may optionally be requested from Telematic terminals (i.e., outgoing call only). CUG information received in an INCOMING CALL packet may be ignored.

  2) *Reverse charging* (13.18)

     This facility may be supported by some networks, and applied on a per call basis, the possibility of requesting reverse charging in outgoing calls is optional for Telematic terminals, but they must be able to properly handle and respond to the incoming call at the called side.

     (As a default, calls should be rejected.)

  3) *Network user identification* (13.21)

     This facility may be applied by networks on a per call basis, following subscription made by prior arrangement for the agreed period of time.

  4) *Charging information* (13.22)

     This facility may be provided by some networks on a per call basis, following subscription made by prior arrangement for the agreed period of time. The information may be handled or processed normally.

     As a minimum requirement, it may be ignored.

  5) *RPOA selection* (13.23)

     This facility may be provided by some networks on a per call basis, following subscription made by prior arrangement for the agreed period of time.

     As a minimum requirement, it may be ignored.

6) *Called line address modified notification* (13.26)

This facility may be provided by some networks on a per call basis, without any particular user request. This information may be processed normally.

As a minimum requirement, it may be ignored.

7) *Call redirection notification* (13.27)

This facility may be provided by some networks on a per call basis, without any particular user request. This information may be processed normally.

As a minimum requirement, it may be ignored.

6 Interactions between the D-channel and B-channel

Communication between the D-channel and B-channel is not synchronized in relation to each other by the ISDN and therefore information exchange via these channels can be accomplished independently and simultaneously. As a consequence of this, messages sent in the D-channel and B-channel in a distinct relationship to each other may be received in a different order.

In order to achieve an orderly operation of the protocols in all Telematic installations it is necessary to have an additional procedure satisfying the respective requirements.

This model, architecture, and primitives of this additional procedure is left for further study. One possible approach is described in Appendix IV.

7 Supplementary services

For application and description see Recommendations F.161, F.200, I.241 and I.25x Series (depending on the type of supplementary service).

8 Terminal response time

(For further study.)

9 Synchronization

One of the characteristics of ISDN is that there is no end-to-end signalling about activation of the protocol instances.

An instance of the data link protocol should only send its first frame when the peer entity is ready to receive it.

To achieve this adequately, the following procedure shall be used.

The sender and receiver follow the sequence:

1) Send "1"-bits until notified of B-channel establishment.
2) Activate the receiver.
3) Send flags.
4) Wait until the first flag arrives from the peer.
5) Consider the peer as active and start communication.

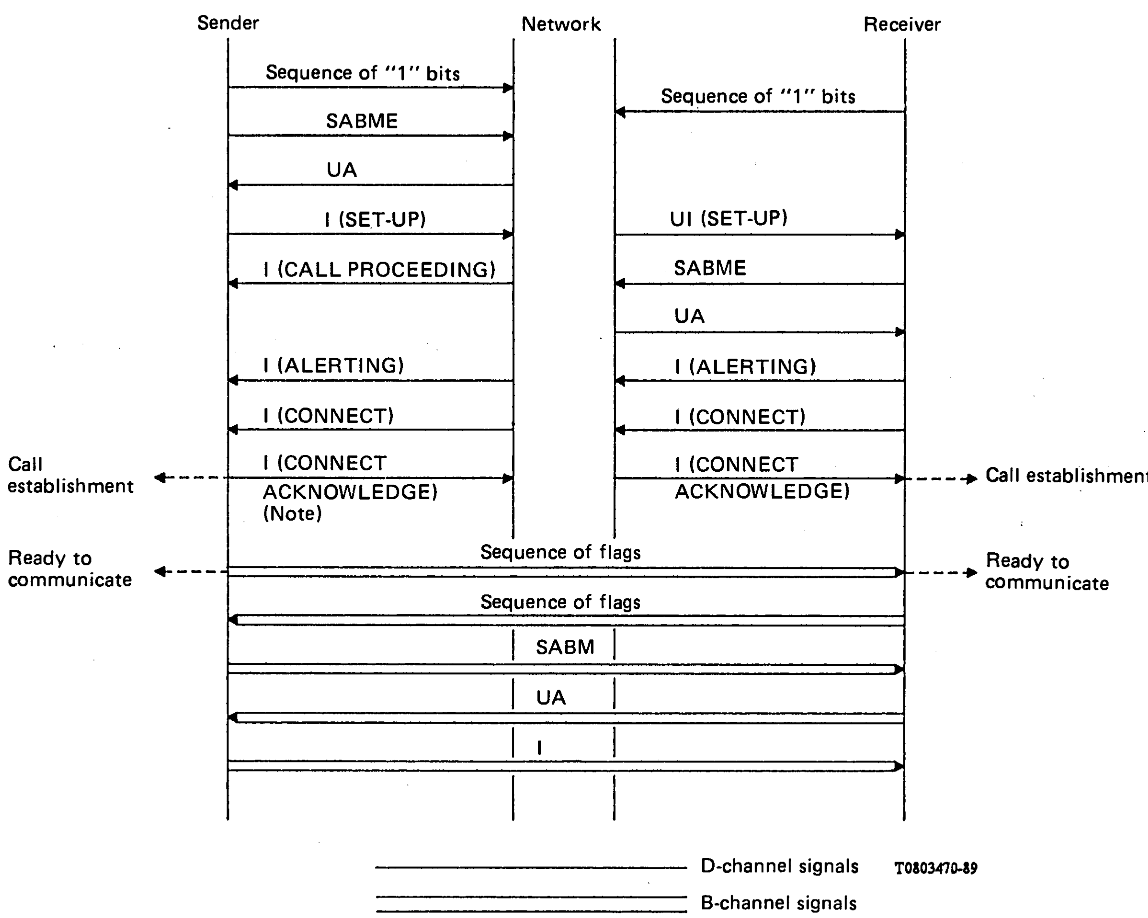
The sequence diagram describing the operation of sender and receiver is shown in Figure 4/T.90.

10 Higher layer protocols

The basic requirements of the Group 4 facsimile service are described in § 1.2.2 of Recommendation F.161. The basic requirements of the Teletex service are described in § 1.2.2 of Recommendation F.200.

10.1 *Transport layer*

The rules given in § 5.3.2 of Recommendation T.70 regarding transport protocol data unit (TPDU) block length are adopted in principle but with the additional provision that the negotiation mechanism is mandatory (e.g., for more efficient communication via satellite links).



Note – The provision of the CONNECT ACKNOWLEDGE signal is optional.

FIGURE 4/T.90

The sequence of synchronisation on layer 2

ANNEX A

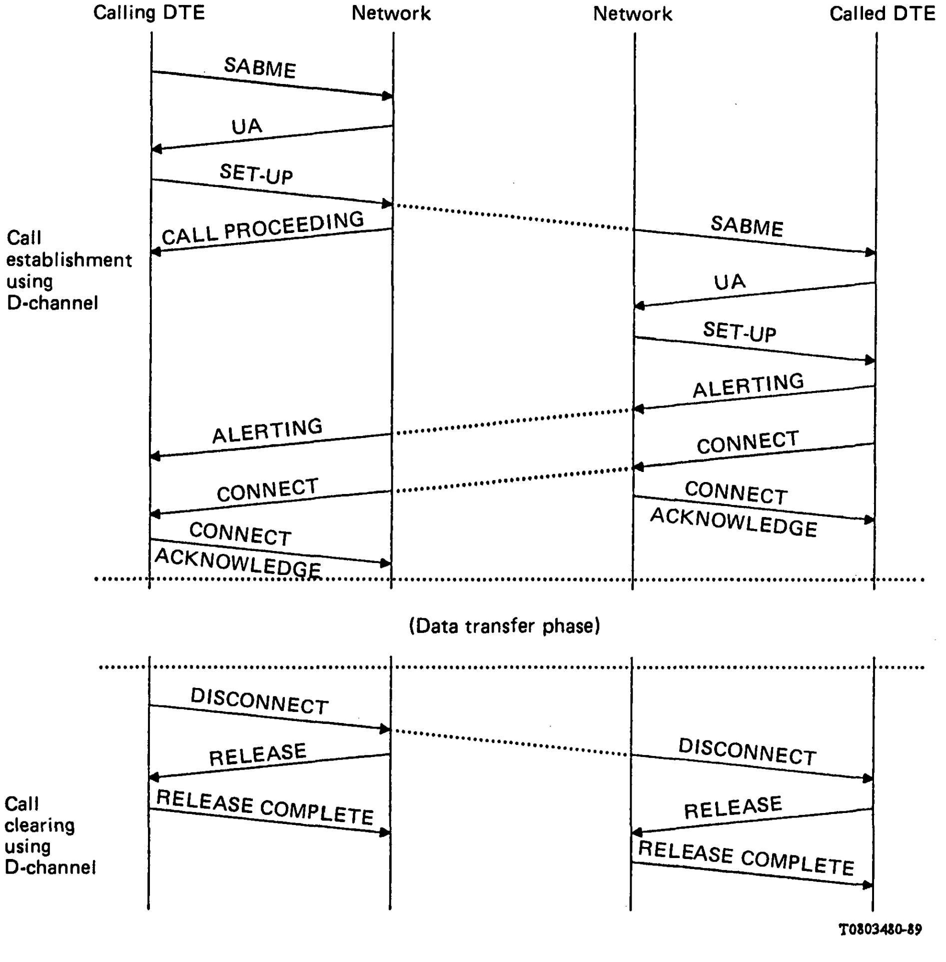
(to Recommendation T.90)

Procedures for connection establishment, connection release and information transfer

Procedures shown below are not the requirements to the terminals for Telematic services, but for reference only.

A.1 *B-channel circuit-switched mode*

a) *Connection control phase*

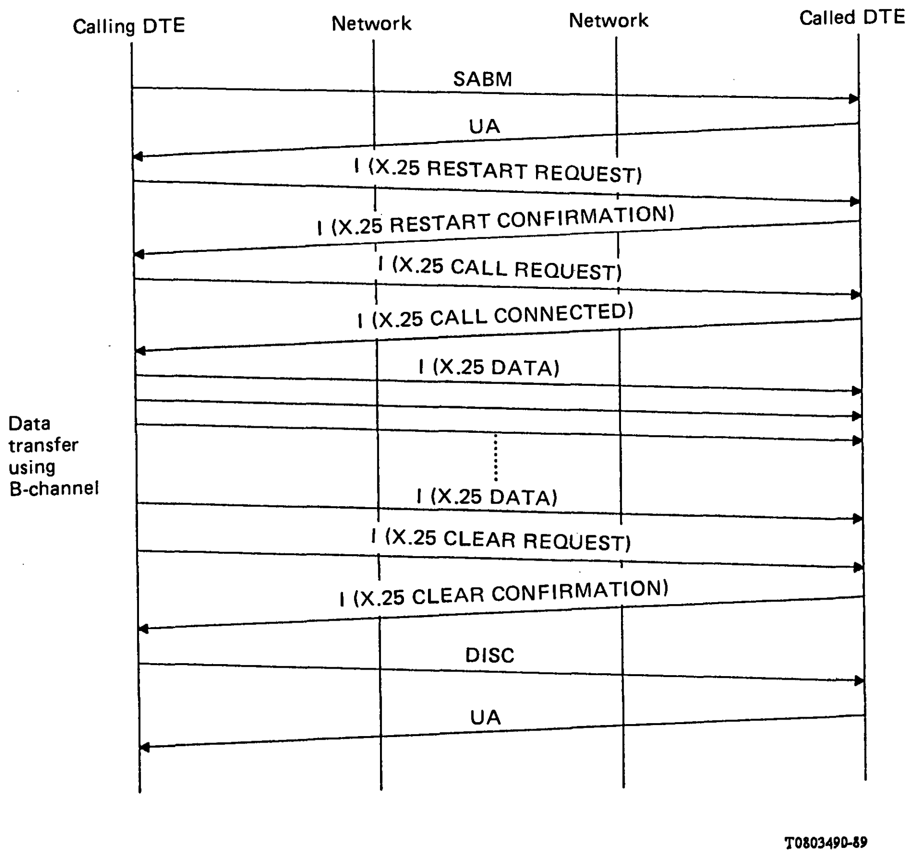


Note 1 – This example shows the procedure in the case where the configuration is point-to-point, and the layer 2 link has not been established. Some signals can be omitted in this situation.

Note 2 – SABME, DISC and UA are specified by Recommendation Q.921 (layer 2). All others are specified by Recommendation Q.931 (layer 3).

FIGURE A-1/T.90

b) *Information transfer phase*



Note – SABM, DISC and UA are specified by Recommendation X.75 (layer 2). All others are specified by Recommendation X.25 PLP (layer 3).

FIGURE A-2/T.90

A.2 *Packet-switched mode*

See relevant signal procedures described in Recommendation X.31.

APPENDIX I

(to Recommendation T.90)

Consideration of incoming calls for facsimile terminals from networks without HLC provision

I.1 In order to cater for the case where calls are incoming from networks not able to convey HLC information (e.g., PSTN, switched 64 kbit/s networks) it must be possible for a G4/G3 terminal to accept calls in some cases without the explicit provision of an HLC field. In this case, the directory (E.164) number must be the over-riding determinant of whether the terminal responds (provided the BC matches). This may involve subscription to "multiple subscriber number" (MSN) supplementary service.

I.2 The three distinct cases likely to occur are:

i) incoming calls from PSTN;

ii) incoming calls from switched 64 kbit/s network (not ISDN);

iii) incoming calls from ISDN.

It is recommended that the following criteria should be used by the terminal to determine whether, and in what mode it should answer the call:

i) *Incoming calls from PSTN*

In this case, the G3/G4 machine should answer the call in G3 mode (including modem and codec functions) if the following criteria are fulfilled:

a) Called ISDN number (E.164) matches number allocated to terminal;
b) BC = 3.1 kHz audio or speech;
c) Call progress indicator (in Q.931 SETUP) = non-ISDN source;
d) HLC = not present;
e) Subaddress = not present.

ii) *Incoming call from switched 64 kbit/s network (not ISDN)*

In this case, the G3/G4 machine should answer the call in G4 mode (no modem or codec functions) if the following criteria are fulfilled:

a) Called ISDN number matches number allocated to terminal;
b) BC = 64 kbit/s;
c) Call progress indicator = non-ISDN source;
(*Note*) – It may not always be possible to determine whether source is ISDN or 64 kbit/s switched);
d) HLC = not present;
e) Subaddress = not present.

iii) *Incoming call from ISDN*

In this case, the G3/G4 machine must answer the call in G4 mode if the following criteria are fulfilled:

a) Called ISDN number matches number allocated to terminal;
b) BC = 64 kbit/s;
c) Call progress indicator [not valid];
d) HLC = G4 Teleservice;
e) Subaddress = if present, this must match terminal subaddress.

I.3 *HLC to be used when polling or sending*

A G3/G4 terminal attempting a G4 call across the ISDN for either polling or sending will send HLC = G4 Fax.

A G3/G4 terminal reattempting a call in G3 mode, following failure with appropriate cause in G4 mode, will establish a 3.1 kHz audio BS with no HLC.

I.4 *HLC to be used by terminal adaptor supporting G3 machines on ISDN*

a) Called ISDN number matches allocated to the terminal adaptor;
b) BC = 3.1 kHz audio or speech;
c) Call progress indicator = non-ISDN source (from PSTN) = [Not valid];
d) HLC = G3 Teleservice (from ISDN);
e) Subaddress = if present this must match terminal subaddress.

APPENDIX II

(to Recommendation T.90)

Optional usage of T.70 NL protocol

II.1 *Information transfer phase*

The T.70 NL option being used by calling DTE and supported by the called DTE.

The network layer shall for the call control phase be as defined in § 2.2.4. The information transfer phase shall be implemented as defined in Recommendation T.70, § 3.3.3.

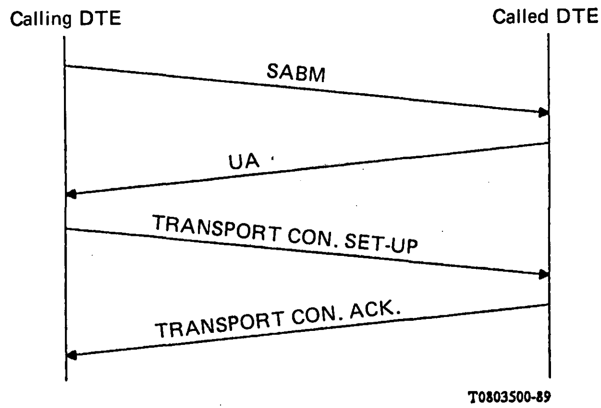


FIGURE II-1/T.90

II.2 *Information transfer phase*

The T.70 NL option being proposed by the calling DTE but not supported by the called DTE.

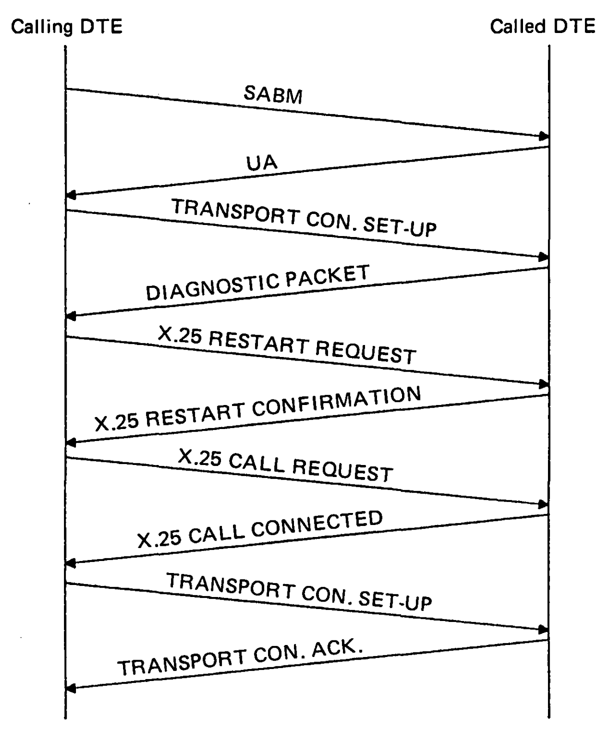


FIGURE II-2/T.90

APPENDIX III

(to Recommendation T.90)

Service definitions and state transition diagrams for the data link layer within the B-channel (CS-mode)

This Appendix contains the result of experience of several implementations of the prescribed link layer for telematic services. This description has been found to be useful in some Administrations for support of conformance testing.

Additional work may be needed in the area of ISDN management and maintenance, however, no clear set of requirements is available at this time. The support of the management and maintenance work is left for further study.

In addition, depending on the further work on the link layer, particularly related to the base modulus for I-frames, some editing may be required (e.g., SABM may become SABME).

Note – Reference to the appropriate paragraph of T.70 or an additional explanation is needed.

III.1 *Service definitions*

III.1.1 *Physical service used by HDLC*

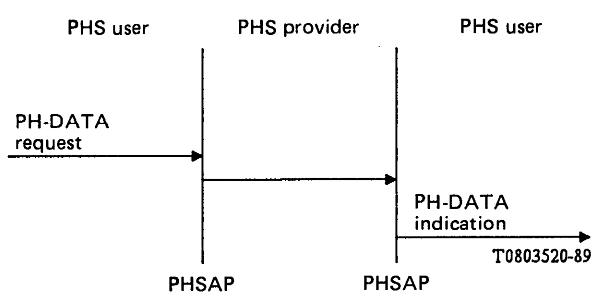


FIGURE III-1/T.90

PH-data transfer

III.1.2 *Data link service (HDLC)*

III.1.2.1 *Data link connection establishment*

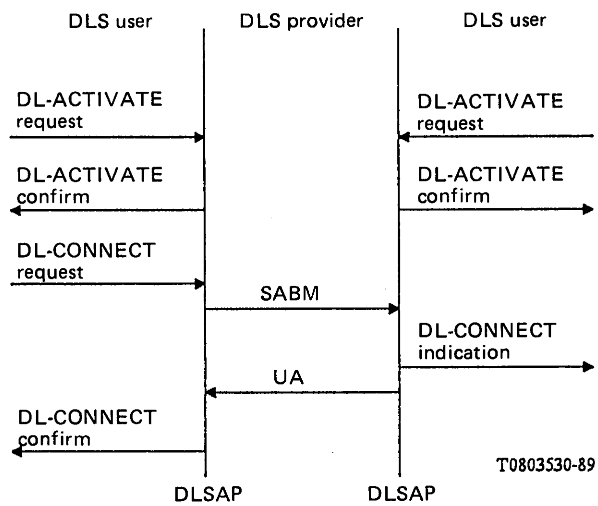


FIGURE III-2/T.90

Successful DLC-establishment

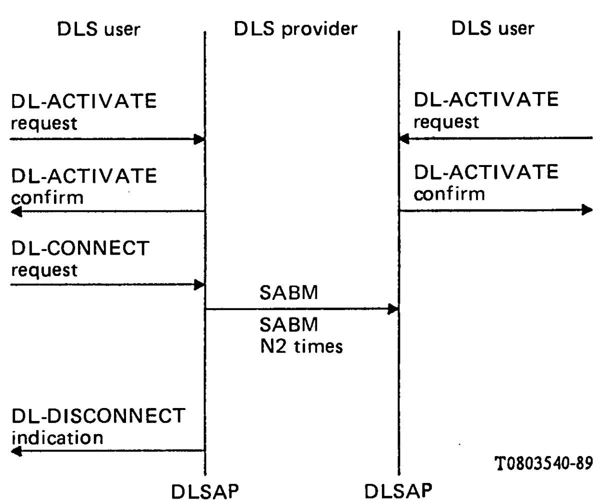


FIGURE III-3/T.90

Not successful DLC-establishment

III.1.2.2 *Data link transfer phase*

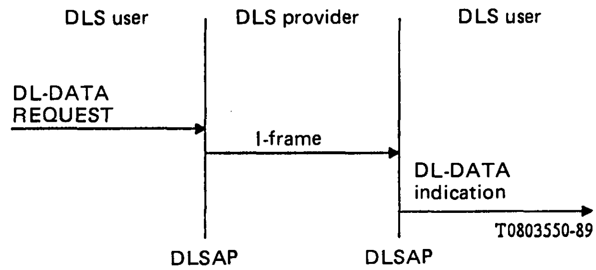


FIGURE III-4/T.90

DL-data transfer

III.1.2.3 *Data link release*

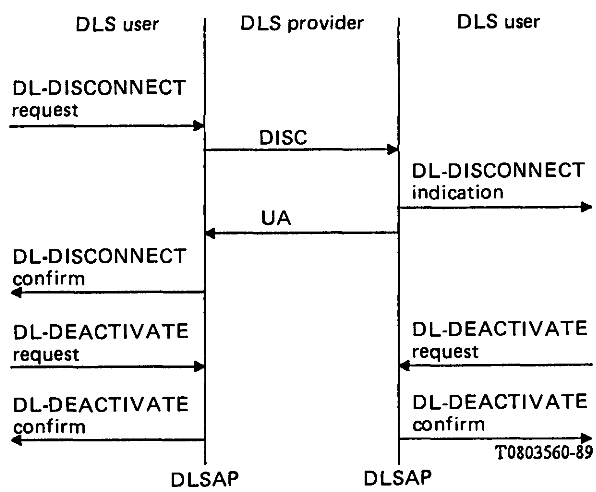


FIGURE III-5/T.90

DL-release initiated by DL-user

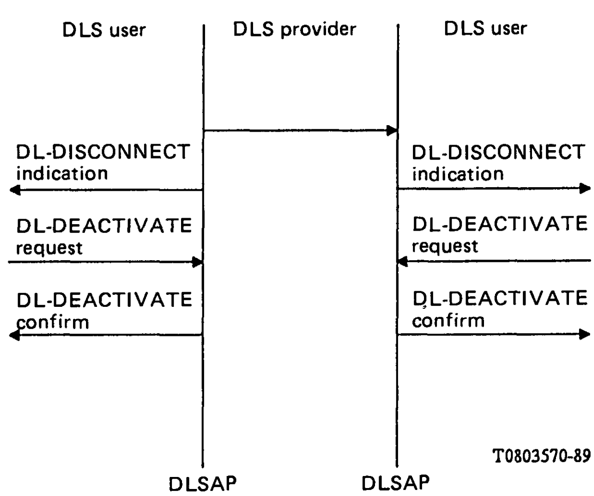


FIGURE III-6/T.90

DL-release initiated by DL-provider

III.1.2.4 *Data link resetting*

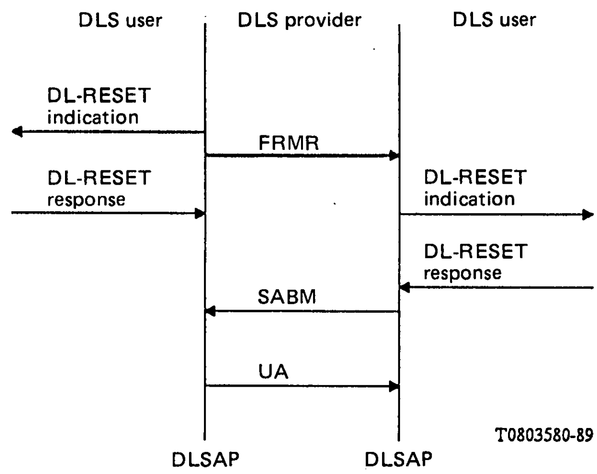


FIGURE III-7/T.90

Successful resetting

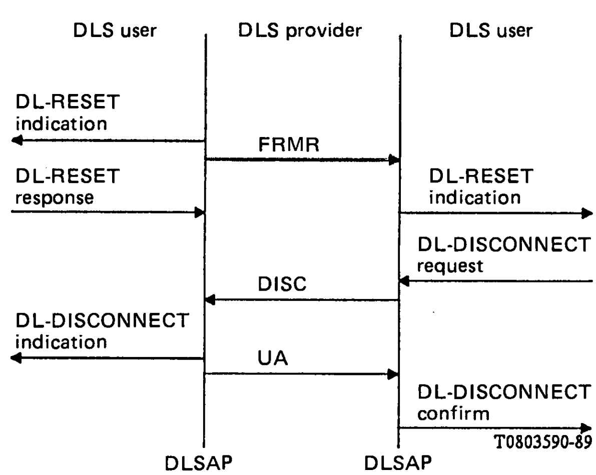


FIGURE III-8/T.90

Resetting not accepted by the receiver of FRMS

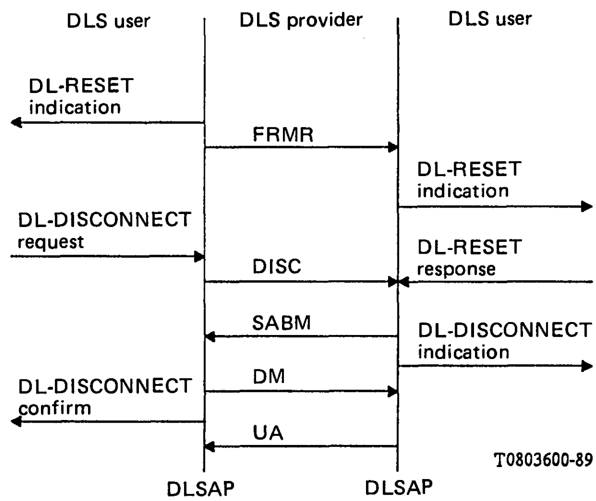


FIGURE III-9/T.90

Resetting not supported by the transmitter of FRMR

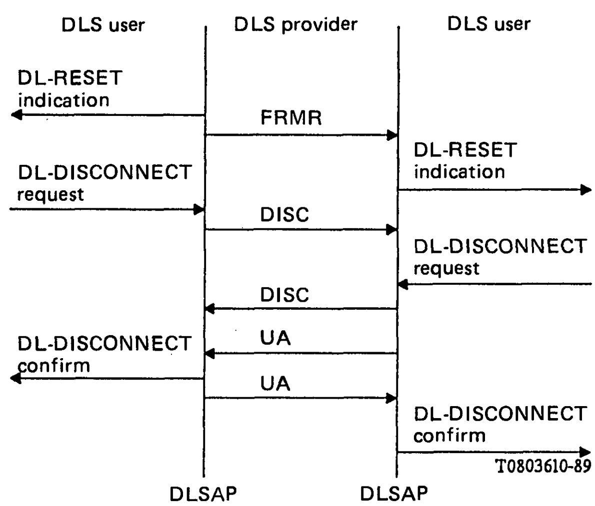


FIGURE III-10/T.90

Resetting not accepted by both

III.2 *State transition disgrams HDLC*

III.2.1 *The relation between the diagrams*

The following diagrams describe the HDLC procedure as one functional unit. The first page comprises the whole protocol and the following pages give the details to specific states.

III.2.2 *Abbreviations*

ABM	Asynchronous Balanced Mode
ADM	Asynchronous Disconnected Mode
R:xxx	Receive xxx (Command or Response)
R:Cxxx	Receive A Commmand
R:Rxxx	Receive A Response
S:xxx	send xxx
F	Final bit
P	Poll bit
XXX	Not this condition
RC	Redrive Counter
RCB	Redrive Counter Busy
IC	I-Frame counter
V_{su}	Variable for sequence updating

III.3 *Summary of frame definitions*

III.3.1 *Invalid frames*

- frames not properly bounded by flags,
- frames containing addresses other than A or B,
- frames with Frame Check Sequence (FCS) error,
- frames containing less than 32 bits between flags.

III.3.2 *Valid frames*

III.3.2.1 *Not expected frames*

NEF, not expected frames (for the receiver) which lead to a frame reject condition (excluding frames with an FRMR control field).

– a command or response control field that is undefined or not implemented,	Type W
– a frame with an information field which is not permitted or supersivory or unnumbered frame with incorrect length,	Type X
– an I-frame with an information field which exceeds the maximum established length,	Type Y
– a frame with an invalid N (R),	Type Z

III.3.2.2 *Expected frames*

- frames which must lead to a reaction (in accordance to the Recommendation) on the receiving station,
- frames which must be ignored only in determined states on the receiving station.

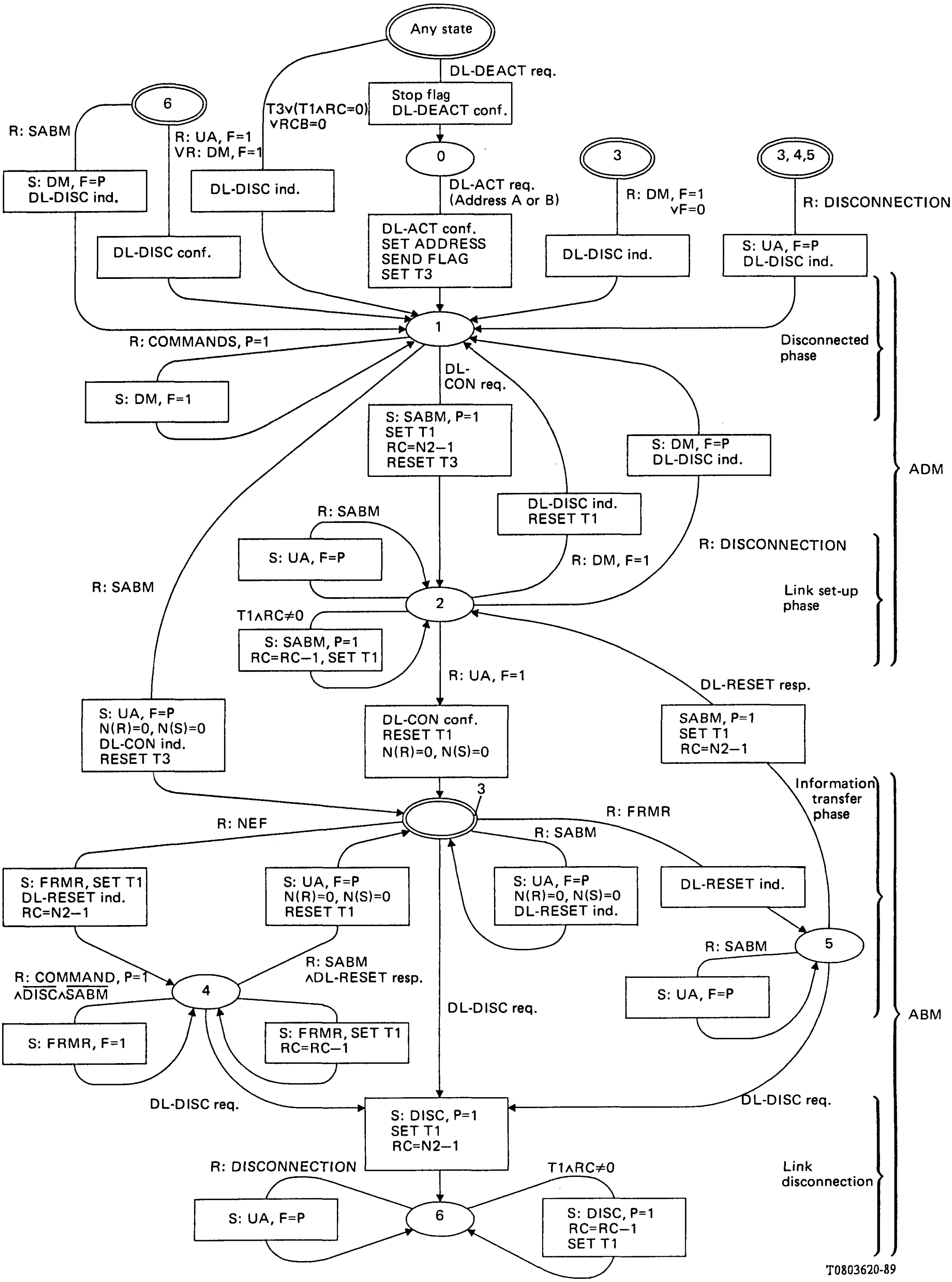
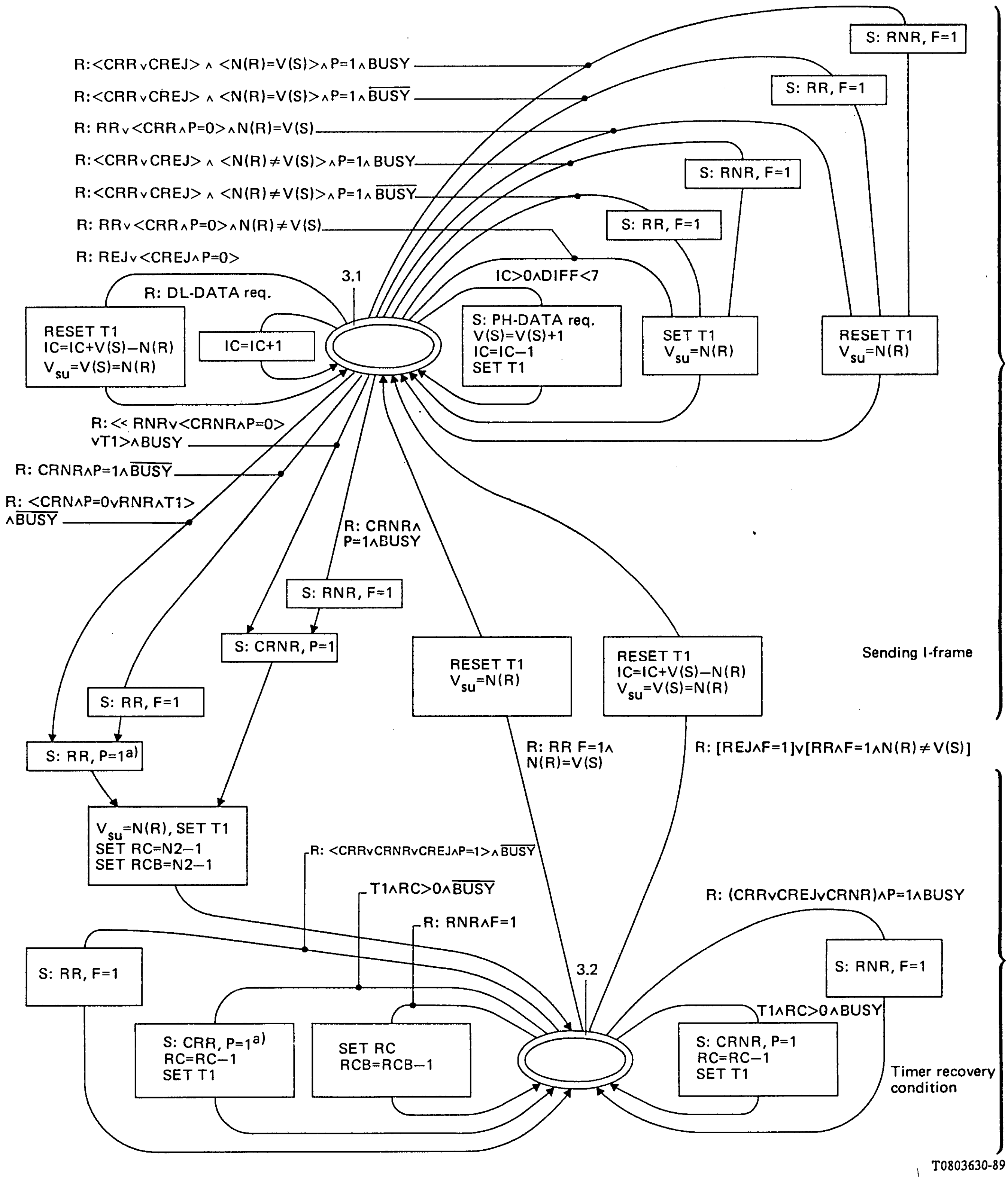


FIGURE III-11/T.90

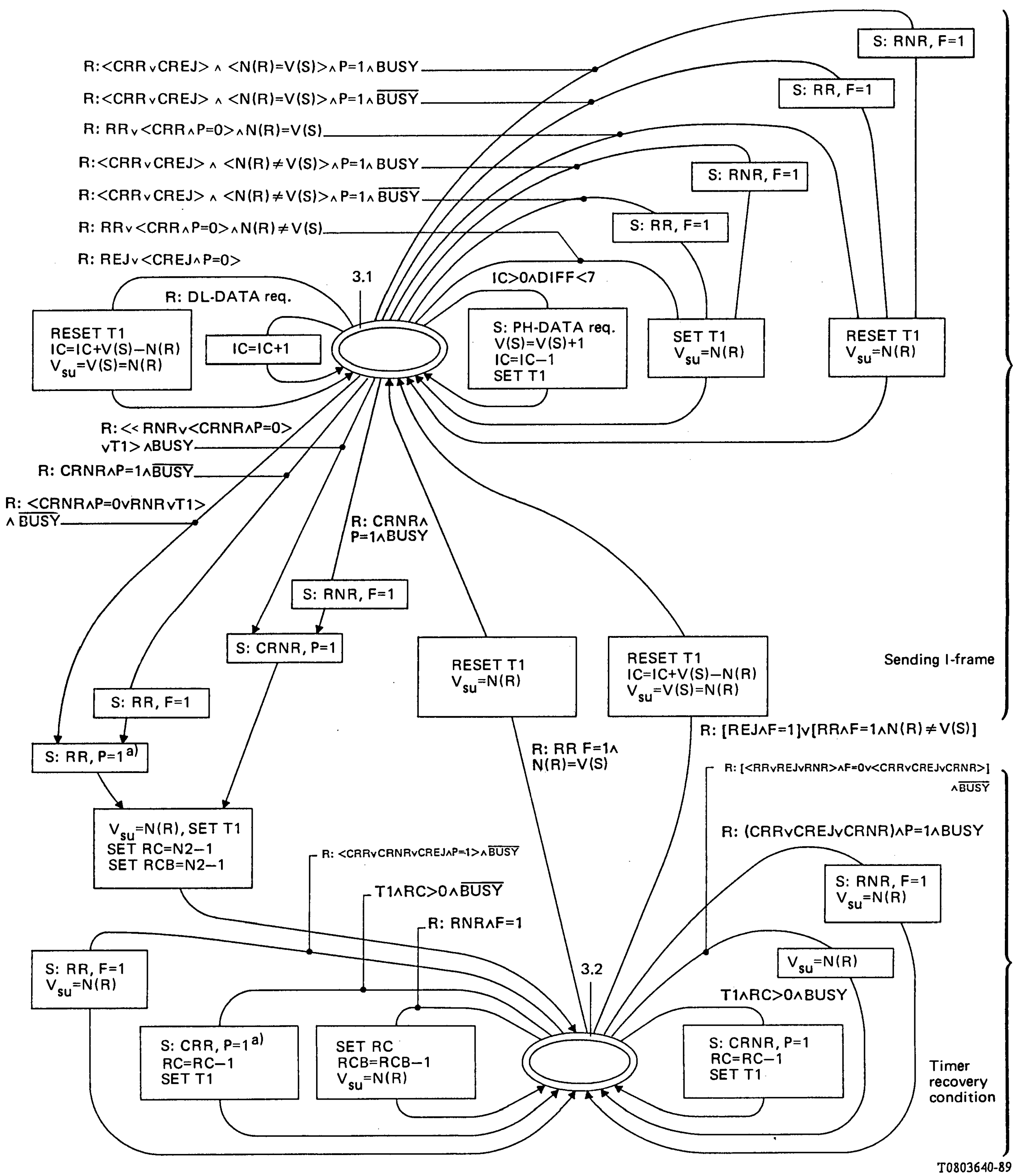
State transition diagram HDLC



a) Alternatively to RR, P = 1 it is allowed to send PH-DATA req. P = 1 or CREJ, P = 1.

FIGURE III-12/T.90

State transition diagram HDLC
(3 Information transfer phase, I-frame control)



a) Alternatively to RR, P = 1 it is allowed to send PH-DATA req. P = 1 or CREJ, P = 1.

FIGURE III-13/T.90

State transition diagram HDLC
(3 information transfer phase, I-frame control with update of N(R)
in timer recovery condition)

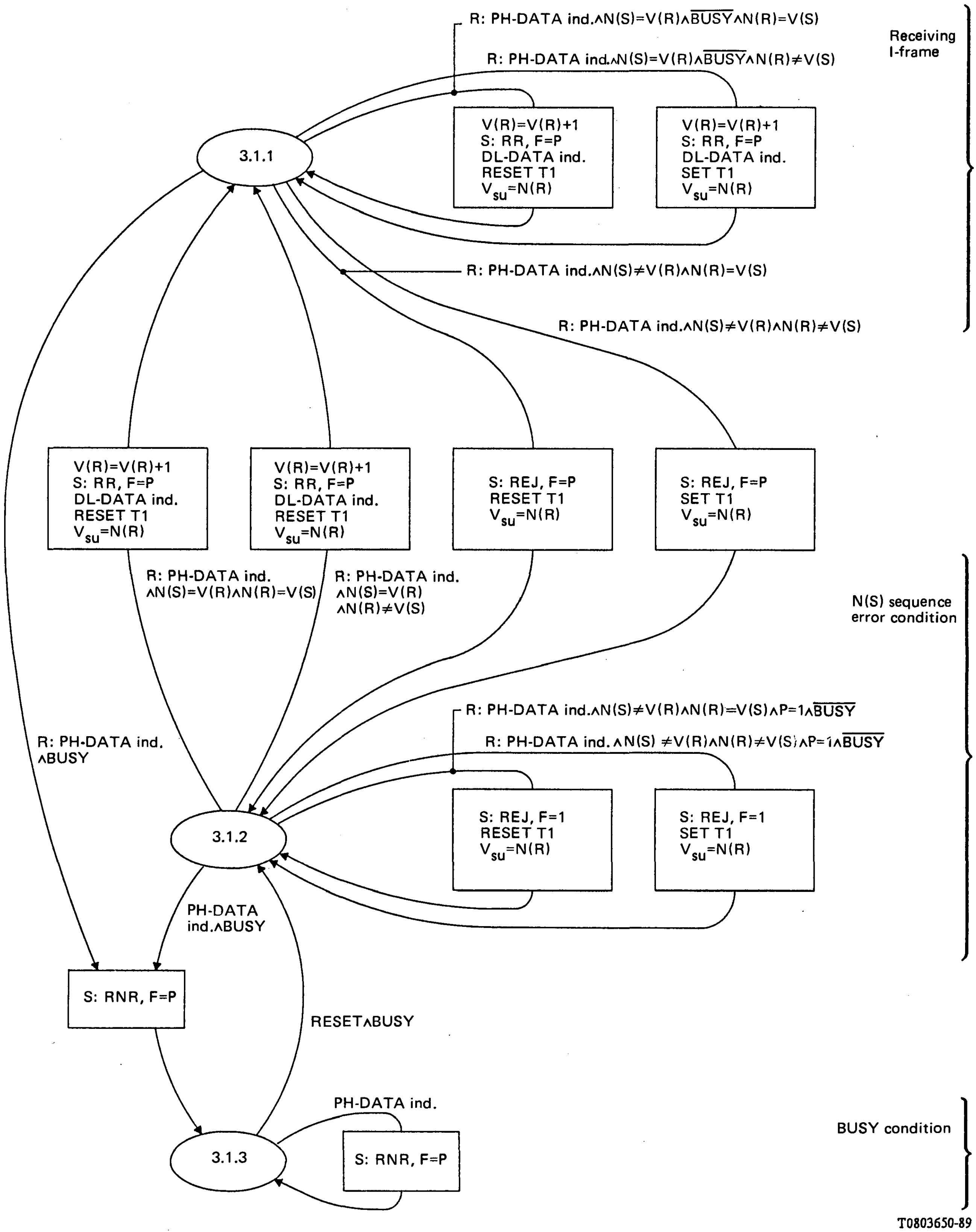


FIGURE III-14/T.90

State transition diagram HDLC
(3.1 information transfer phase, I-frame acknowledgement)

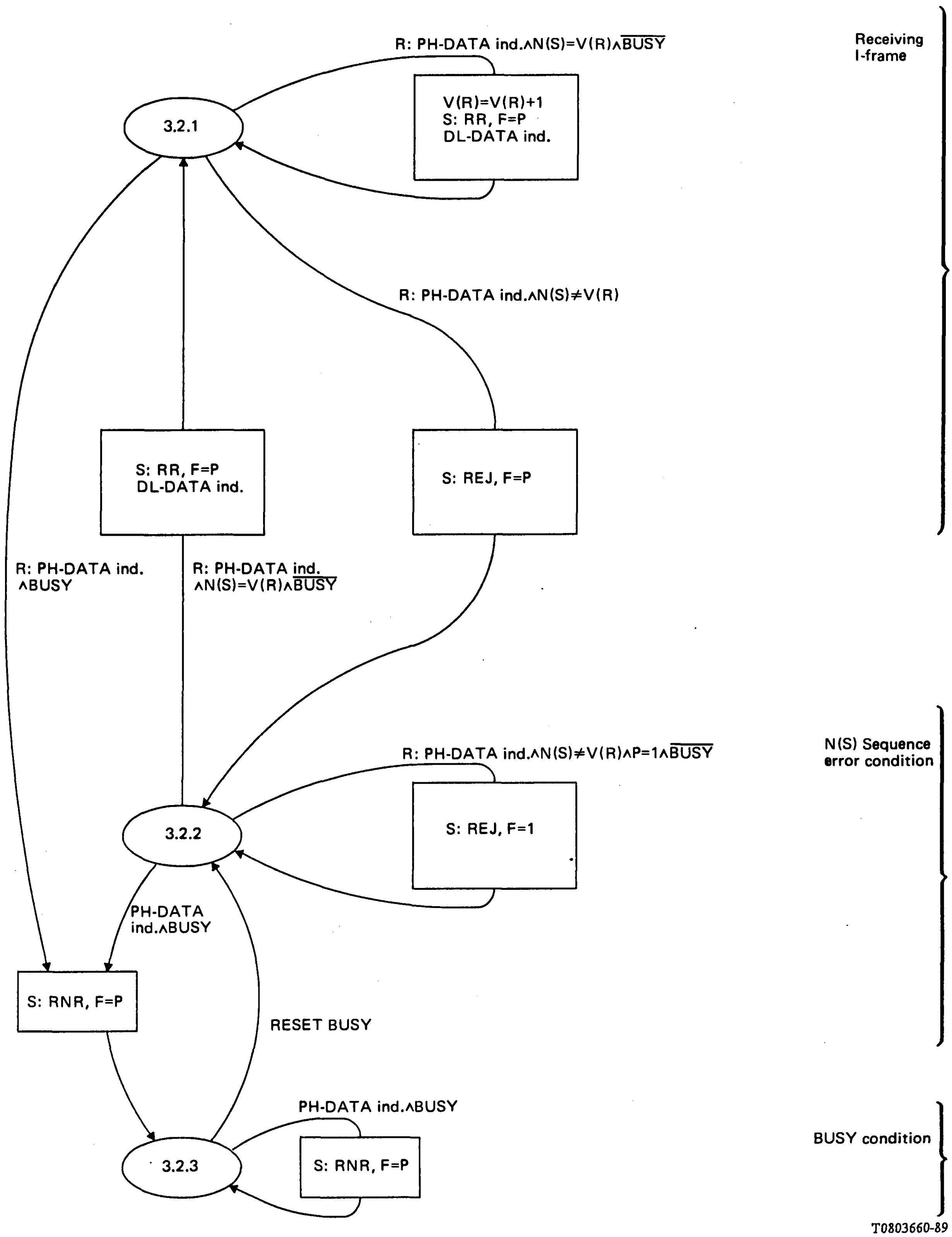


FIGURE III-15/T.90

State transition diagram HDLC
(3.2 information transfer phase, I-frame acknowledgement in exception conditions)

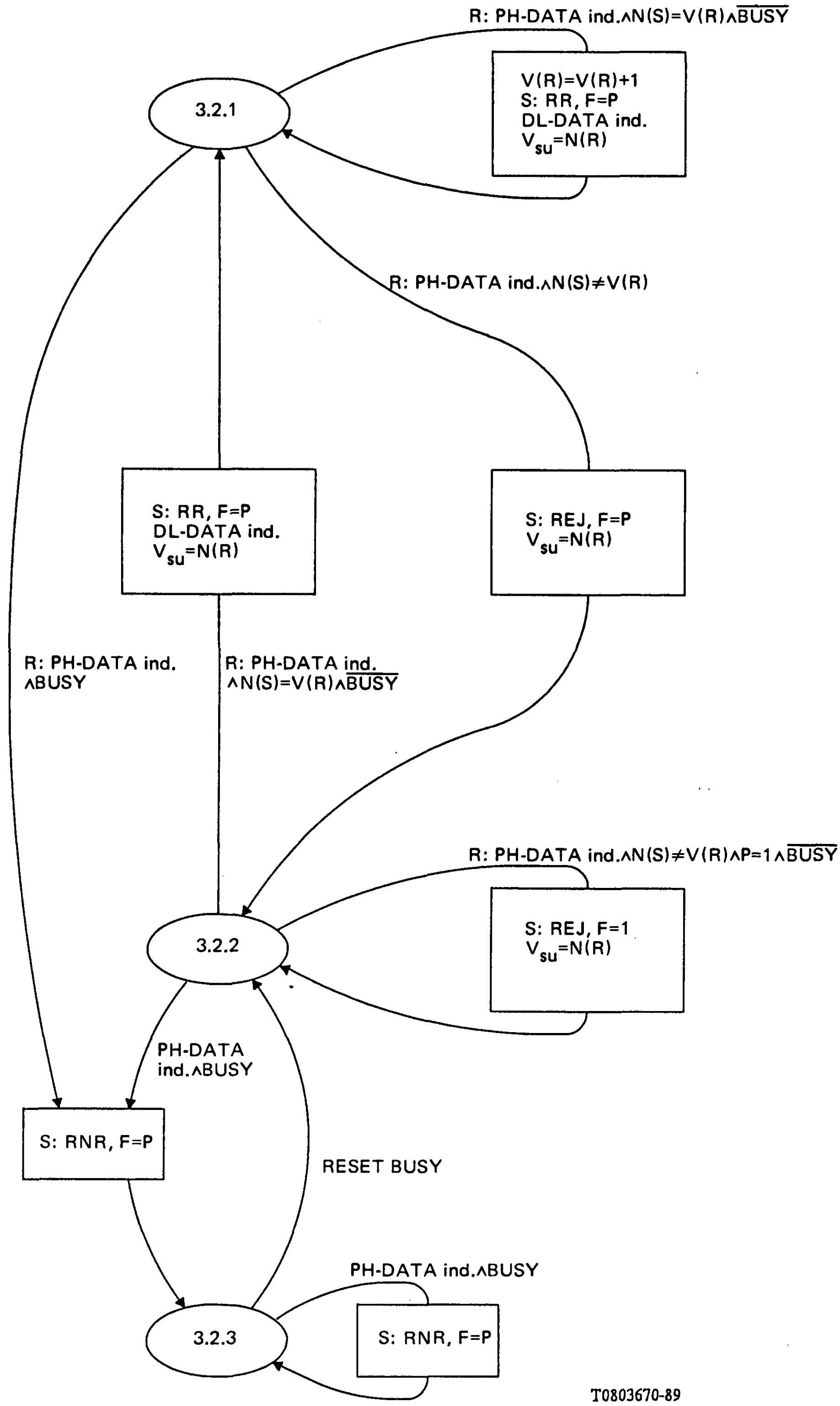


FIGURE III-16/T.90

State transition diagram HDLC
(3.2 information transfer phase, I-frame acknowledgement in exception conditions with update of N(R))

APPENDIX IV

(to Recommendation T.90)

Possible model for telematic endsystems taking into account the D-channel/B-channel coordination function

Layers 4 to 7	Telematic protocols layers 4 to 7	
Layer 3	D-/B-channel coordination function	
	Q.931	X.25 PLP plus application rules
Layer 2	Q.921	X.75 LAPB plus application rules
Layer 1	I.430/I.431	
	<D-channel>	<B-channel>

Note – D-channel signalling functions of concern to the terminal-user-interface but not supported by the layer 4-7 services are to be handled separately from these services.

FIGURE IV-1/T.90

Model for telematic endsystems taking into account the D-channel/B-channel coordination function

There are various ways of specifying the layer 3 covering the coordination function. In principle, the layer 3 can either be specified as a monolith or as a set of individual modules.

The structuring into the three modules:

– Layer 3 D-channel
– Layer 3 B-channel and
– Layer 3 D-/B-channel coordination.

is obvious, as the first two modules are almost ready-made, available thus leaving the coordination module to be specified from the functionality point of view. The implementation itself is in the responsibility of the manufacturer.

INTERNATIONAL INFORMATION EXCHANGE FOR INTERACTIVE VIDEOTEX

(Geneva, 1980; amended at Malaga-Torremolinos, 1984)

CONTENTS

Preamble

The CCITT,

considering

(a) that there is increasing interest in public network-based new interactive information retrieval services using domestic television receivers suitably supplemented, or other apparatus, as terminal equipment;

(b) that the CCIR is studying standards for broadcast *Teletext* services for general reception and has expressed a view that it is desirable that terminal equipment compatibility should exist between broadcast Teletext systems for general reception and public network-based data bank systems;

(c) that such services should be provided over public networks in accordance with CCITT Recommendations and may be required to operate as an international service;

(d) that such services may interwork with terminals provided for text communication services (Teletex for example);

(e) that some Administrations intend to have an early introduction of, or have already introduced, public interactive *Videotex* services;

unanimously recommends

that the following technical provisions be applied for international information exchange for interactive Videotex service.

1 Purpose and scope of the Recommendation

1.1 *Purpose*

1.1.1 The purpose of this Recommendation is:

a) to facilitate an orderly introduction of early Videotex services (including the continuation of existing services, with a clear identification of potential enhancements) that need to be considered in future developments;

b) to identify parameters needed to design Videotex terminals; and

c) to provide technical recommendations desirable for potential interworking of other services with Videotex services.

1.2 *Scope*

1.2.1 This Recommendation describes the characteristics of coded information that is exchanged between countries participating in the international interactive Videotex service (as described in Recommendation F.300) and defines the display features corresponding to its various elements.

1.2.2 Videotex systems are text communication systems having in addition the capability of a given level of pictorial representation and a repertoire of display attributes. The text and the pictures obtained are intended to be displayed using the current television (TV) raster standards of the different countries.

1.2.3 Different options are offered as a choice for the Administrations to implement their national services. Substantial degrees of compatibility exist between these options, but some transcoding may be necessary to facilitate interworking.

1.2.4 For the international service, four different options for representing pictorial information have been recognized:

a) mosaic character sets;

b) geometric system;

c) dynamically redefinable character sets;

d) photographic representation.

These options are not mutually exclusive and it is possible that systems may develop using two or more options.

1.2.5 For international interworking, two categories of TV systems have to be considered:

a) systems having a vertical resolution of 525 lines per TV frame at 30 TV frames per second;

b) systems having a vertical resolution of 625 lines per TV frame at 25 TV frames per second.

1.2.6 Interworking problems at the pictorial level between countries having different recognized pictorial systems and/or television standards require further study.

1.2.7 This Recommendation is structured as follows:

§§ 1, 2 and 3	deal with the features common to all the options;
§ 4	deals with the coding of characters of the Videotex alphanumeric repertoire defined in Annex B;
§ 5	deals with the alphamosaic option;
§ 6	deals with the alphageometric option;
§ 7	deals with the dynamically redefinable character sets (DRCS) option;
§ 8	deals with the alphaphotographic option;
§ 9	deals with future enhancements and identifies features requiring further study such as: audio, downloaded software, motion, etc.;
§ 10	deals with line and end-to-end protocols;
§ 11	deals with interworking with other services.

Some of these parts have not been completed, and therefore contain guidelines towards future extensions rather than a complete technical specification.

2 General Videotex coding structure

2.1 *General*

2.1.1 The basis of the coding structure for the Videotex service is Recommendation T.50 and the international standards ISO 2022, ISO 6937 for the 7-bit environment. Specifically, the shift-in code SI (0/15) invokes the G0 set for alphanumeric text mode of operation, and the shift-out code SO (0/14) invokes the G1 set, for all the models (see Annex A). The use of the 8-bit coding scheme is for further study.

2.1.2 In addition to the provisions made by ISO 2022, the transmission of alphabetic characters having diacritical signs is effected by transmitting the code representing the diacritical mark together with the code of the basic alphabetic character.

2.1.3 The different options are designated (and invoked) by specific escape sequences.

2.2 *Designation and invocation in the context of the alphamosaic option*

2.2.1 Two different modes for the alphamosaic option have been identified. They differ in their display control sets. These control sets are designated as the C1 set by the following control sequences: ESC 2/2 4/0 for the serial mode and ESC 2/2 4/1 for the parallel mode, as assigned by ISO. Individual controls are represented by: ESC F_e sequences.

2.2.2 The mosaic graphics set is designated (in the parallel mode) as the G1 set by an escape sequence ESC 2/9 6/3 as allocated by ISO.

2.3 *Designation and invocation in the context of the alphageometric option*

2.3.1 The alphageometric coding scheme is to be designated and invoked by the escape sequence ESC 2/5 (5/x) in accordance with § 5.3.8 of ISO 2022 standard. This designates and invokes a complete code with interpretation as follows.

2.3.2 All the meanings and interpretation of Recommendation T.50 and ISO 2022 remain the same, including C0, G0 and G2 with the exception of SI and SO. The codes of the G1 set and their meanings and interpretations are as described in § 6.

2.3.3 The designation and invocation of the complete code by the sequence ESC 2/5 (5/x) is to be terminated only by ESC 2/9 (F) or ESC 2/13 (F), designating a normal G1 set.

2.4 *Designation and invocation in the context of DRCS*

2.4.1 A DRCS is a set of characters whose shapes are sent from the service and down-loaded via the line. It may be used to represent alphabetic characters, special symbols, or picture element symbols for constructing fine graphics. Once loaded, the DRCS are regarded as members of a library that can be designated by appropriate ESC sequences as G0, G1, G2, G3 sets. One scheme is described in § 7 in the context of a general architecture.

2.5 *Designation and invocation in the context of the alphaphotographic option*

(For further study.)

3 Common features

3.1 *General*

3.1.1 The features pertaining to individual systems will be described in the corresponding paragraphs. The common features comprise common display features and common control functions.

3.2 *Common display features*

3.2.1 The *defined display area* is that rectangular position of the display in which all text and pictorial images may be presented (see Figure 1/T.100).

3.2.2 The *border area* is that part of the visible display of a terminal that is outside the defined display area (see Figure 1/T.100).

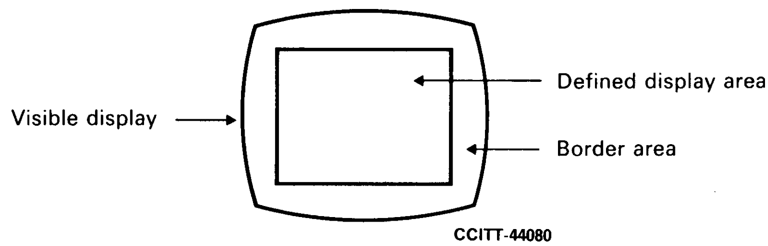


FIGURE 1/T.100
Common display features

3.3 *Common format effector and code extension control functions*

3.3.1 *General*

3.3.1.1 The format effector control functions described for the Videotex system permit the active drawing position to be moved on the visible display area. These are taken from the C0 set (see Figure 2/T.100) together with the *Space* character 2/0. In order to permit interworking between Videotex and other text communications services, these control functions have functional compatibility to the extent possible with the basic C0 control set utilized by these other services.

3.3.2 *Format effector controls*

3.3.2.1 Some of the format effector control functions may be used from terminal to computer with different meanings.

3.3.2.2 *Active position backward (APB)*

This control function causes the active position to be moved one character position backwards on the same row. APB on the first character position on the row moves the active position to the last character position of the preceding row. APB on the first character position on the first row moves the active position to the last character position of the last row.

3.3.2.3 *Active position forward (APF)*

This function causes the active position to be moved to the next character position forward on the same row. At the last position on the row, this control moves the active position to the first character position on the following row. APF on the last character position of the last row moves the active position to the first character position of the first row.

3.3.2.4 *Active position down (APD)*

This function causes the active position to be moved to the equivalent character position on the following row. APD on the last row moves the active position of the equivalent character position of the first row of the display frame or causes a roll-up to be made.

				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
b4	b3	b2	b1		0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	②						
0	0	0	1	1	②	③						
0	0	1	0	2	②	③						
0	0	1	1	3	②	③						
0	1	0	0	4	②	③						
0	1	0	1	5	ENQ	②						
0	1	1	0	6	②	②						
0	1	1	1	7	①	②						
1	0	0	0	8	APB	CAN						
1	0	0	1	9	APF	SS2						
1	0	1	0	10	APD	①						
1	0	1	1	11	APU	ESC						
1	1	0	0	12	CS	①						
1	1	0	1	13	APR	SS3						
1	1	1	0	14	SO	③						
1	1	1	1	15	SI	③						

CCITT-44090

Note 1 — Reserved for future study.

Note 2 — Reserved for transmission control characters. Their use in Videotex is for further study.

Note 3 — The definitions of these control functions are given in the relevant options.

Note 4 — As in all the code tables in this Recommendation, the shaded positions do not belong to the character set described.

FIGURE 2/T.100

The primary set of control functions for international interactive Videotex

3.3.2.5 *Active position up (APU)*

This function causes the active position to be moved to the equivalent character position on the preceding row. APU on the first row moves the active position to the equivalent character position on the last row of the same display frame.

3.3.2.6 *Clear screen (CS)*

This function causes the screen to be cleared and causes the active position to be moved to the first character position on the first row.

3.3.2.7 *Active position return (APR)*

This function causes the active position to be moved to the first character position of the same row.

3.3.2.8 *Space (SP)*

A control function that causes the active position to be moved one character width forward on the same row. It is also regarded as a graphic character with no foreground. In those systems that define an explicit background, the space copies the background colour into the active position and moves the active position one character width forward. If used in conjunction with the inversion attribute it copies the foreground colour into the active position and moves the active position one character width forward.

3.3.2.9 *Cancel (CAN)*

A control function that fills all the character positions of the row, after the active position, with spaces and returns the active position to its original value.

3.3.3 *Code extension control functions*

3.3.3.1 Code extension control functions are used to expand the capability of the 7-bit code beyond 128 different characters or functions. Code extension functions alter the meaning of a number of characters following them.

3.3.3.2 *Escape (ESC)*

A control character that is used to provide additional control functions other than transmission control functions and that alters the meaning of a limited number of contiguously following bit combinations in the manner specified in Recommendation T.51.

3.3.3.3 *Control sequence introducer (CSI)*

A code extension control function that is used to provide coded representations for additional control functions, in particular for control functions with parameters such as presentation control functions.

3.3.3.4 *Shift-out (SO)*

A control character that is used in conjunction with the *Shift-in* character to extend the graphic character set of the code and that alters the meaning of the bit combinations of columns 2-7 of the code table, until the occurrence of the *shift-in* character, except that the meaning of the bit combinations corresponding to the *space* character and the *delete* character (positions 2/0 and 7/15) are unaffected.

3.3.3.5 *Shift-in (SI)*

A control character, used in conjunction with the *shift-out* character, that reinstates the former meanings of the bit combinations of columns 2-7 of the code table.

3.3.3.6 *Single shift (SS2)*

This character alters the meaning of the single-bit combination following it. That bit combination must be one of those from columns 2-7 except 2/0 and 7/15. The meaning of the bit combination concerned is derived from an appropriately designated G2 graphic set.

3.3.3.7 *Single shift (SS3)*

This character alters the meaning of the single-bit combination following it. That bit combination must be one of those from columns 2-7 except 2/0 and 7/15. The meaning of the bit combination concerned is derived from an appropriately designated G3 graphic set.

3.4 *Miscellaneous*

3.4.1 *Null (NUL)*

This function may occur in non-transparent modes in the received bit stream at the terminal. It shall be regarded as a time filler and discarded.

3.4.2 *Enquiry (ENQ)*

A control character used as a request for a response from a remote station, which response may include station identification and/or station status.

3.5 *Coding of control functions*

3.5.1 A proposed coding of the control functions described is shown in Figure 2/T.100 as a C0 set, except for CSI which is coded in the C1 set.

4 Representation of alphanumeric characters in a Videotex system

4.1 *General*

4.1.1 The repertoire for the Latin alphabet is shown in Annex B. The repertoire is derived from ISO 6937. Terminals capable of displaying a subset of the Videotex repertoire shall be permitted.

4.1.2 Character repertoires for non-latin based languages can be accommodated in a similar manner to the latin alphabet. (For further study.)

4.2 *Coding*

4.2.1 Section 4.2 describes the coding of characters the shape of which are stored in the terminal. Some languages require that consecutive letters or diacritical marks will be joined and that no space appear between the characters. When an intersymbol space is required, it will be part of the character description.

4.2.2 The code tables are shown in Figures 3/T.100 and 4/T.100. The code combinations representing characters not included in the Videotex repertoire shall not be transmitted.

4.2.3 All the permitted combinations may be expected in the international exchange of information between two national services. It is the responsibility of Administrations to decide whether this exchange is a direct terminal to data-base operation or has to be performed through a gateway. See Recommendation F.300.

4.2.4 The graphic characters from columns 2, 3, 5, 6 and 7 of the supplementary set are invoked one at a time by SS2.

4.2.5 A character with a diacritical mark is transmitted by the sequence SS2, a character from column 4 from the supplementary set, and the appropriate character from the primary set. The diacritical marks are non-spacing.

4.2.6 The ISO registration of graphics character sets will indicate any special features such as their use in conjunction with other graphic character sets or non-spacing characters, etc.

4.2.7 For languages based on other than the Latin alphabet further study is required.

5 Alphamosaic option

5.1 *General*

5.1.1 In the alphamosaic option, the display frame is composed of defined character positions which may be occupied by any of the characters of the repertoire. The repertoire is composed of the alphanumeric repertoire and a mosaic repertoire. The mosaic repertoire is formed by dividing the character space into a matrix of 2×3 elements. There are 63 different combinations of these elements.

				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
b4	b3	b2	b1		0	1	2	3	4	5	6	7
0	0	0	0	0				0	@	P		p
0	0	0	1	1			!	1	A	Q	a	q
0	0	1	0	2			"	2	B	R	b	r
0	0	1	1	3				3	C	S	c	s
0	1	0	0	4				4	D	T	d	t
0	1	0	1	5			%	5	E	U	e	u
0	1	1	0	6			&	6	F	V	f	v
0	1	1	1	7			'	7	G	W	g	w
1	0	0	0	8			(	8	H	X	h	x
1	0	0	1	9			)	9	I	Y	i	y
1	0	1	0	10			*	:	J	Z	j	z
1	0	1	1	11			+	;	K		k	
1	1	0	0	12			,	<	L		l	|
1	1	0	1	13			-	=	M		m	
1	1	1	0	14			.	>	N		n	
1	1	1	1	15			/	?	O	①	o	

CCITT-44100

① Position 5/15 can be displayed as "low line", —, or as "number sign" #, to represent the terminator function required for some existing Videotex services.

FIGURE 3/T.100

The primary set of graphic characters for international interactive Videotex

				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
b4	b3	b2	b1		0	1	2	3	4	5	6	7
0	0	0	0	0				°		―	Ω	ĸ
0	0	0	1	1			¡	±	`		Æ	æ
0	0	1	0	2			¢	²	´		Đ	đ
0	0	1	1	3			£	³	^		ª	ð
0	1	0	0	4			$	×	~		Ħ	ħ
0	1	0	1	5			¥	µ	¯			ı
0	1	1	0	6			#	¶	˘		Ĳ	ĳ
0	1	1	1	7			§	·	˙		Ŀ	ŀ
1	0	0	0	8			¤	÷	¨		Ł	ł
1	0	0	1	9			‘	’			Ø	ø
1	0	1	0	10			“	”	˚		Œ	œ
1	0	1	1	11			«	»	¸		º	ß
1	1	0	0	12			←	¼			Þ	þ
1	1	0	1	13			↑	½	˝		Ŧ	ŧ
1	1	1	0	14			→	¾	˛		Ŋ	ŋ
1	1	1	1	15			↓	¿	ˇ		ŉ	

CCITT-44111

FIGURE 4/T.100

The supplementary set of graphic characters for international interactive Videotex

5.1.2 Two modes have been identified, which are known as *serial* and *parallel* modes respectively. The two modes are distinguished by their display control sets which are coded in C1 sets, designated and represented by ESC F_e sequences as described in § 2.2.1.

5.1.3 The two modes have common features and specific features described in §§ 5.2 to 5.4 below.

5.2 *Common control functions*

5.2.1 *General functions*

The *active position home (APH)*

This function causes the active position to be moved to the first position of the first row. Its coded representation is 1/14 in Figure 2/T.100.

5.2.2 *Device control functions*

The following device control functions have been defined.

5.2.2.1 *Definitions*

cursor on (CON)

F: curseur en marche (CON)

S: cursor activo (CON)

The cursor on (CON) causes the active position to be visualized as a marker.

cursor off (COF)

F: curseur arrêté (COF)

S: cursor inactivo (COF)

The cursor off (COF) causes the active position to be displayed in the same way as other character positions.

device stop (DSP)

F: arrêt dispositif (DSP)

S: detención de dispositivo (DSP)

The device stop (DSP) causes a designated terminal device to stop.

device start (DST)

F: mise en marche dispositif (DST)

S: arranque de dispositivo (DST)

The device start (DST) causes a designated terminal device to start.

device wait (DW)

F: dispositif en attente (DW)

S: espera de dispositivo (DW)

The device wait (DW) causes a designated terminal device to pause.

5.2.2.2 *Coding*

CON is coded 1/1, COF is coded 1/4 in the C0 set. DSP, DST and DW functions are coded as 3-character sequences of the Form ESC 3/x (P), where x = 7, 6 and 5 respectively, and P is a parameter that designates a particular device.

5.3 *Serial mode*

5.3.1 *General*

5.3.1.1 The serial mode is based on the assumption that changes in character attributes normally occur in interword spacings. This results in control characters being serially stored in the page memory and normally results in their display on the screen as a rectangle in the prevailing background colour.

5.3.1.2 The C1 set for the serial mode is given in Figure 5/T.100. Display controls of the serial set causes the active position to be moved one character position forward. In that case, the position thus vacated is to be generally displayed as a space. The display control *hold mosaics* ESC 5/14 may modify this situation.

5.3.2 *Display control functions*

5.3.2.1 The (F_e) codes are listed as follows:

5.3.2.2 Alpha red
Alpha green
Alpha yellow
Alpha blue
Alpha magenta
Alpha cyan
Alpha white

Controls functions that cause the currently designated and invoked alphanumeric set to be displayed in the indicated colour until the occurrence of an explicit colour control or the end of a row.

5.3.2.3 *Flashing*

A control function that causes the characters following it in the same row to be displayed alternately as they would normally be displayed, and as spaces, in the prevailing background colour, under the control of a timing device in the receiver.

5.3.2.4 *Steady*

A control function that causes the action of *flashing* to be stopped.

5.3.2.5 *Start box*

Reserved for starting the action of defining a picture area in a page of text. (For further study.)

5.3.2.6 *End box*

Reserved for terminating the action of boxing. (For further study.)

5.3.2.7 *Normal height*

A control function that causes the graphic characters following it to occupy one character position each.

5.3.2.8 *Double height*

A control function that causes the characters following it to occupy each its active positive and the corresponding position on the following row.

5.3.2.9 Mosaics red
Mosaics green
Mosaics yellow
Mosaics blue
Mosaics magenta
Mosaics cyan
Mosaics white

Control functions that cause the mosaic graphic set to be displayed in the indicated colour until the occurrence of an explicit colour control or the end of the row. Unallocated code table positions (4/0-5/15) cause the characters of the currently designated and invoked alphanumeric set to be displayed. This is defined as *blast-through* operation.

b4	b3	b2	b1		0	1	2	3	4	5	6	7
				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
0	0	0	0	0					①	①		
0	0	0	1	1					Alpha red	Mosaics red		
0	0	1	0	2					Alpha green	Mosaics green		
0	0	1	1	3					Alpha yellow	Mosaics yellow		
0	1	0	0	4					Alpha blue	Mosaics blue		
0	1	0	1	5					Alpha magenta	Mosaics magenta		
0	1	1	0	6					Alpha cyan	Mosaics cyan		
0	1	1	1	7					Alpha white	Mosaics white		
1	0	0	0	8					Flashing	Conceal display		
1	0	0	1	9					Steady	Conti-guous mosaics		
1	0	1	0	10					End box ①	Sepa-rated mosaics		
1	0	1	1	11					Start Box ①	①		
1	1	0	0	12					Normal height	Black back-ground		
1	1	0	1	13					Double height	New back-ground		
1	1	1	0	14					①	Hold mosaics		
1	1	1	1	15					①	Release mosaics		

CCITT-44120

① Reserved for further study.

Note — This coding represents the final bit combination of ESC F_e sequences in a 7-bit code.

FIGURE 5/T.100

The supplementary set of control functions-serial mode

5.3.2.10 *Conceal display*

A control function that causes all characters following it, although stored in the receiver, to be displayed as spaces until the user chooses to reveal them.

5.3.2.11 *Contiguous mosaics*

A control function that causes the mosaic set to be displayed as represented in Figure 6/T.100 with all cells being contiguous.

5.3.2.12 *Separated mosaics*

A control function that causes the mosaics set to be displayed as represented in Figure 6/T.100 with all cells being separated by the prevailing background colour.

5.3.2.13 *Black background*

A control function that causes the background colour to be black.

5.3.2.14 *New background*

A control function that causes the current colour as defined by previous colour control functions to become the new background colour. The foreground colour is unchanged.

5.3.2.15 *Hold mosaics*

A control function that causes the character positions occupied by display controls to be displayed by repetition of the last displayable mosaic character.

5.3.2.16 *Release mosaics*

A control function that causes the action of *hold mosaics* to be stopped.

5.3.3 *Mosaic graphics*

5.3.3.1 The *serial mosaic* graphic set is given in Figure 6/T.100 and the default conditions of the mode are shown in Table 1/T.100.

5.4 *Parallel mode*

5.4.1 *General*

5.4.1.1 The *parallel* mode is based on an explicit description of the display frame. This means that the active position is moved only by action of the format effectors or at the reception of spacing display characters. All other functions, including display functions, are non-spacing, not depending on whether or not the terminal needs a space on the screen to process them. It is the responsibility of the information provider to limit the display of pages to pages to fit the capability assumed to receive, without any modification, pages designed for lower grade terminals.

5.4.1.2 In addition to functions described in § 3.3, the following functions are defined.

active position addressing (APA, coded 1/15)

F: adressage de position active (APA)

S: direccionamiento de posición activa (APA)

This code is followed by two characters. If these both range from 3/0 to 3/9, they represent in decimal form respectively the tens and units of the row address of the first character to be displayed. This first character will be displayed on the first character position of the addressed row. If they both range from 4/0 to 7/14, they represent respectively the row address and the column address, in binary form with 6 useful bits, of the first characters to be displayed.

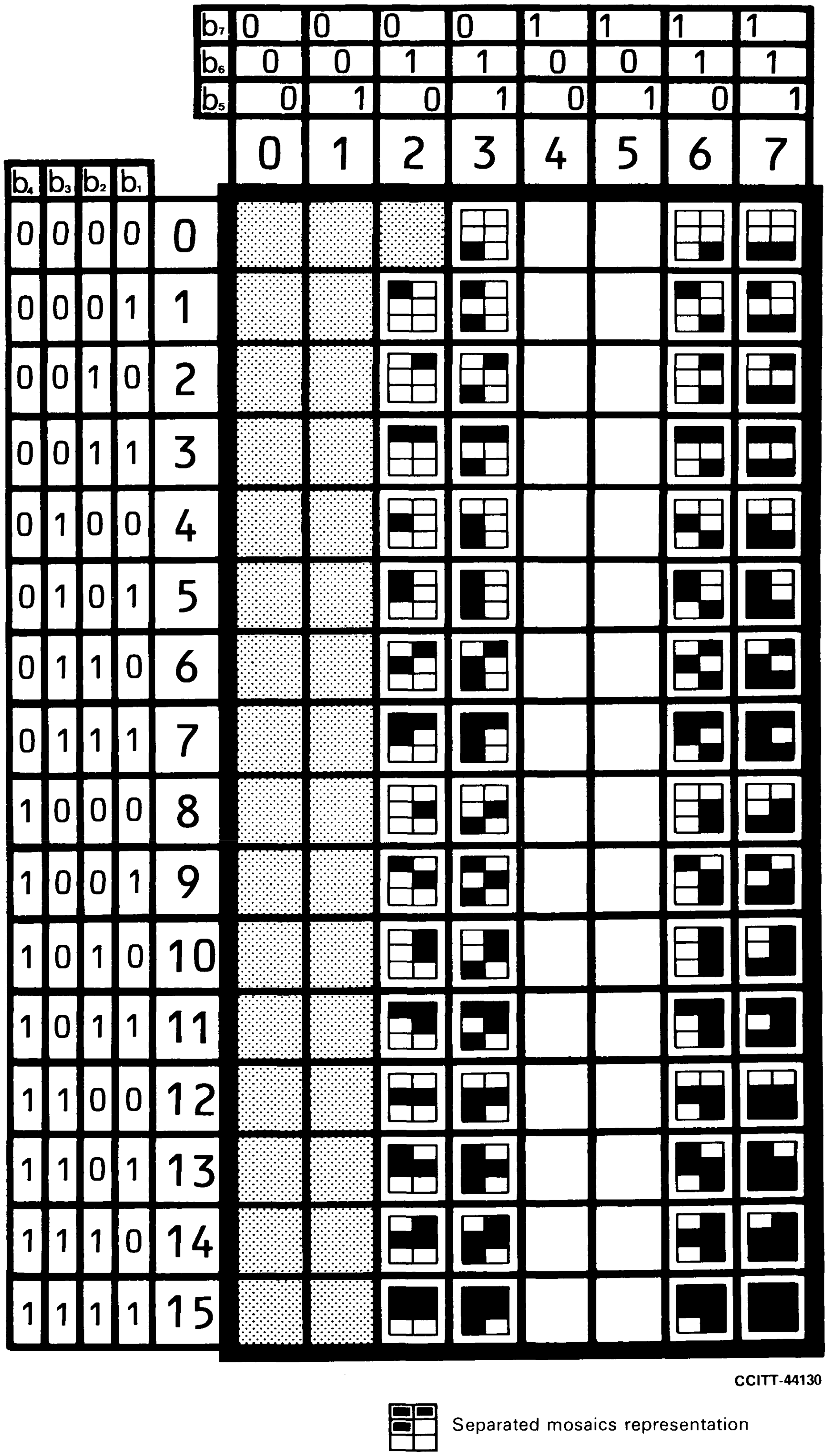


FIGURE 6/T.100

The mosaic set for the serial mode with blast-through characters in columns 4 and 5

TABLE 1/T.100

Display modes and control characters-serial mode

<table>
<tr><th colspan="2">Display mode</th><th>Set at</th><th>Set after
(see Note)</th><th colspan="2">Complementary display mode</th><th>Set at</th><th>Set after
(see Note)</th></tr>
<tr><td colspan="2">Alphanumerics</td><td>Row start</td><td>4/1 4/2 4/3
4/4 4/5 4/6 4/7</td><td colspan="2">Block mosaics</td><td>–</td><td>5/1 5/2 5/3
5/4 5/5 5/6 5/7</td></tr>
<tr><td colspan="2">Contiguous</td><td>Row start 5/9</td><td>5/9</td><td colspan="2">Separated</td><td>5/10</td><td>5/10</td></tr>
<tr><td rowspan="3">Fore-ground display colour</td><td>includes red</td><td>Row start</td><td>4/1 4/3 4/5 4/7
5/1 5/3 5/5 5/7</td><td rowspan="3">Fore-ground display colour</td><td>excludes red</td><td>–</td><td>4/2 4/4 4/6
5/2 5/4 5/6</td></tr>
<tr><td>includes green</td><td>Row start</td><td>4/2 4/3 4/6 4/7
5/2 5/3 5/6 5/7</td><td>excludes green</td><td>–</td><td>4/1 4/4 4/5
5/1 5/4 5/5</td></tr>
<tr><td>includes blue</td><td>Row start</td><td>4/4 4/5 4/6 4/7
5/4 5/5 5/6 5/7</td><td>excludes blue</td><td>–</td><td>4/1 4/2 4/3
5/1 5/2 5/3</td></tr>
<tr><td colspan="2">Black background</td><td>Row start 5/12</td><td>–</td><td colspan="2">New background colour</td><td>5/13</td><td>–</td></tr>
<tr><td colspan="2">Stop conceal</td><td>Row start</td><td>4/1 4/2 4/3
4/4 4/5 4/6 4/7
5/1 5/2 5/3
5/4 5/5 5/6 5/7</td><td colspan="2">Conceal</td><td>5/8</td><td>–</td></tr>
<tr><td colspan="2">Steady</td><td>Row start 4/9</td><td>–</td><td colspan="2">Flash</td><td>–</td><td>4/8</td></tr>
<tr><td colspan="2">Unboxed</td><td>Row start 4/10</td><td>4/10</td><td colspan="2">Boxed</td><td>4/11</td><td>4/11</td></tr>
<tr><td colspan="2">Normal height</td><td>Row start 4/12</td><td>–</td><td colspan="2">Double height</td><td>–</td><td>4/13</td></tr>
<tr><td colspan="2">Release</td><td>Row start</td><td>5/15</td><td colspan="2">Hold</td><td>5/14</td><td>–</td></tr>
</table>

Note – All attribute control characters are preceded by Escape ESC.

repeat (RPT, coded 1/2)

F: répétition (RPT)

S: repetición (RPT)

This code indicates that the preceding graphics character is to be repeated. The number of repetitions is indicated in binary form by the six least significant bits of the subsequent character chosen from columns 4 to 7. The character itself is not included in the count. This function does not apply to control characters.

5.4.1.3 A supplementary set of 32 controls, of which 31 have been allocated, are coded as a C1 set (see Figure 7/T.100). The attributes defined by such controls become a property of the active position and move with it under the action of format effectors or spacing display characters.

5.4.1.4 The mosaic repertoire is coded as a G1 set, of which several representations may be defined (see Figure 8/T.100).

5.4.2 *Display control functions*

5.4.2.1 The display control functions are of two kinds depending on the range of their action:

- *Defined display area attributes* apply to individual character locations. Their action is limited to zones separated by APA functions.
- *Full screen attributes* apply to the full screen area and are taken as default values for defined display area attributes.

The defined display area attributes are coded as functions from the supplementary set of control functions (see Figure 7/T.100), with two character escape sequences.

The full screen attribute is coded as a function from the supplementary set of control functions with four character escape sequences (see § 5.4.2.3).

5.4.2.2 Attributes for use in the defined display area are as follows.

5.4.2.2.1 $\left.\begin{array}{l}\text{Black foreground}\\\text{Red foreground}\\\text{Green foreground}\\\text{Yellow foreground}\\\text{Blue foreground}\\\text{Magenta foreground}\\\text{Cyan foreground}\\\text{White foreground}\end{array}\right\}$ Causes the following characters to be written in the colour indicated.

5.4.2.2.2 *Flashing*

This control function causes the characters following it to be displayed alternatively as they would otherwise be displayed, and as spaces, under the control of a timing device in the receiver.

5.4.2.2.3 *Steady*

This control function causes the action of *flashing* to be stopped.

5.4.2.2.4 *Start box*

This control function causes the characters following it to be inset or added to a television picture, when the receiver is in the user's control. (For further study.)

5.4.2.2.5 *End box*

This control function causes the action of *start box* to be stopped. (For further study.)

5.4.2.2.6 *Normal size*

This control function causes the characters following it to occupy one character position each.

b4	b3	b2	b1		0	1	2	3	4	5	6	7
				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
0	0	0	0	0					Black foreground	Black background		
0	0	0	1	1					Red F....	Red B....		
0	0	1	0	2					Green F....	Green B....		
0	0	1	1	3					Yellow F....	Yellow B....		
0	1	0	0	4					Blue F....	Blue B....		
0	1	0	1	5					Magenta F....	Magenta B....		
0	1	1	0	6					Cyan F....	Cyan B....		
0	1	1	1	7					White F....	White B....		
1	0	0	0	8					Flash	Conceal display		
1	0	0	1	9					Steady	Stop lining		
1	0	1	0	10					End box [1]	Start lining		
1	0	1	1	11					Start box [1]	[2]		
1	1	0	0	12					Normal size	Normal polarity		
1	1	0	1	13					Double height	Inverted polarity		
1	1	1	0	14					Double width	Transparent background		
1	1	1	1	15					Double size	Stop conceal		

1 For further study

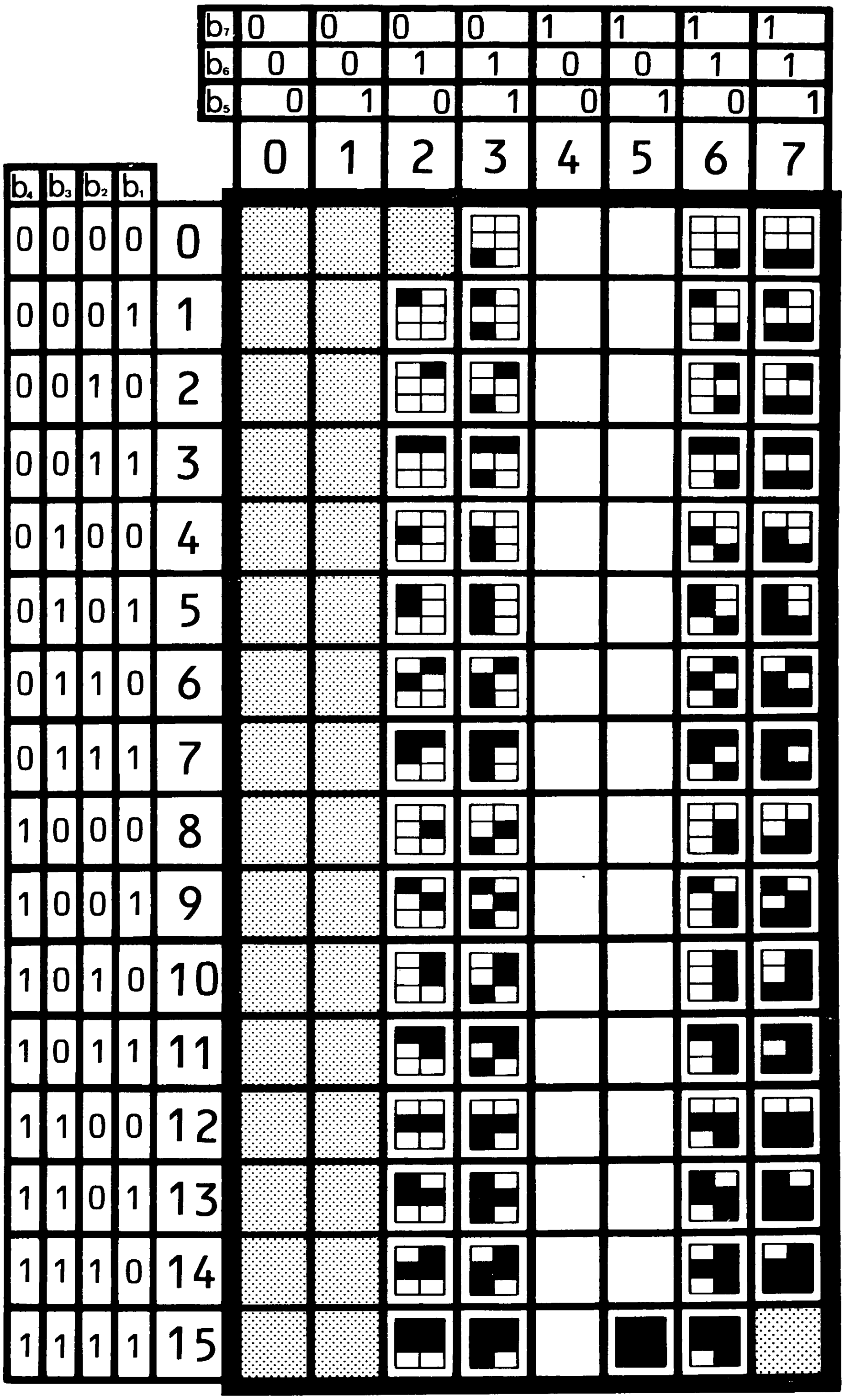
2 Reserved for CSI

CCITT-44140

Note — This coding represents the final bit combination of ESC F_e sequences in a 7-bit code.

FIGURE 7/T.100

Supplementary set of control functions — parallel mode



CCITT-44150

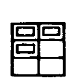
 « Lined » mosaic character

FIGURE 8/T.100

The mosaic character set — parallel mode

5.4.2.2.7 *Double height*

This control function causes the characters following it to occupy each its active position and the corresponding position on the previous row. (The origin of a character is the bottom left corner of the character position.)

5.4.2.2.8 *Double width*

This control function causes the characters following it to occupy two consecutive character positions on the same row, and the active position to be moved two positions forward with every character.

5.4.2.2.9 *Double size*

This control function causes the characters following it to occupy the active position, the next on the row and the two corresponding character positions on the previous row. The active position is moved two character positions forward with every character.

5.4.2.2.10 Black background Red background Green background Yellow background Blue background Magenta background Cyan background White background	Causes the following characters to be displayed in their foreground colour on a background of the colour indicated.

5.4.2.2.11 *Transparent background*

This control function causes the characters following it to be displayed with a transparent background. This means the area not occupied by the foreground colour takes the underlying background colour. This may be one of the eight colours or the video picture as defined by the off screen attributes.

5.4.2.2.12 *Conceal display*

This control function causes the characters following it, in the same unit although stored in the receiver, to be displayed as spaces until the user chooses to reveal them.

5.4.2.2.13 *Stop conceal*

This control function causes the action of *conceal display* to be stopped.

5.4.2.2.14 *Start lining*

This control function causes the characters following in the same unit to be lined. The shape of lining may be different depending on the character set used. In the case of the mosaic set, the lining causes the six cells to be separated with a background boundary.

5.4.2.2.15 *Stop lining*

This control function causes the action of *start lining* to be stopped.

5.4.2.2.16 *Normal polarity*

This control function causes the action of *inverted polarity* to be stopped.

5.4.2.2.17 *Inverted polarity*

This control function causes the characters following it, in the same unit, to be displayed as if the background and the foreground colour have been exchanged. In the flashing attribute, the polarity of the flashing clock is also inverted.

5.4.2.3 *Full-screen attributes*

5.4.2.3.1 Full-screen attributes apply for the total display period and include the border area. In addition, provisions are made for full-row attributes, applying for the entire row including the border area related to that row.

Full-screen attributes display controls are represented by four character *Escape* sequences of the form ESC 2/3 2/0 F_e where F_e is taken from Figure 7/T.100.

Full-row attributes display controls are represented by four-character *Escape* sequences of the form ESC 2/3 2/1 F_e.

5.4.2.3.2 The following full-screen attributes need precise definition:

- *Transparent background:* The full-screen area is occupied by a picture, which may not be part of the Videotex service (e.g. a television picture). Non-concealed characters appear on this picture. If they are also displayed with defined display area transparent background, only the foreground appears over the picture. Concealed characters are displayed as transparent spaces.
- *Conceal:* The defined display area is in the full-screen background colour until the user chooses to reveal it or until this attribute is stopped by full-screen stop conceal.
- *Full-screen stop conceal:* This has the same action as the action of the user on the reveal key.

5.4.2.3.3 For row-defined full-screen attributes, the following may also apply:

- lined;
- double width;
- double height.

5.4.3 *Coding of the mosaic repertoire*

5.4.3.1 The mosaic repertoire is designated as a G1 set invoked by the SO function. Two alternative fonts (contiguous and separated) are proposed. The separated font is obtained by applying the lining attribute applied to the mosaic set. The mosaic set code table is given in Figure 8/T.100 together with examples of the fonts.

5.4.4 *Default conditions*

5.4.4.1 *Default full-screen attributes*

At the beginning of a display frame (initiated by function CS) the default conditions for full-screen attributes are set at white foreground, black background, single size, unboxed, revealed, steady, non-lined.

5.4.4.2 *Default defined display area attributes*

After functions directly addressing a character location on the screen (APH or APA function) the defined display area attributes are reset to the value of the current full-screen attributes.

5.4.4.3 *Default full-row attributes*

The default condition of full-row attributes is the current value of full-screen attributes.

6 **Alphageometric option**

6.1 *General*

6.1.1 *Description*

6.1.1.1 In the alphageometric option, the display is composed of alphanumeric texts and pictorial drawings that are defined in terms of geometric primitives transmitted to the terminal as drawing commands.

6.1.1.2 One coding scheme for the alphageometric option for Videotex is described in § 6.

6.1.2 *Designation and invocation of geometric codes*

6.1.2.1 The designation and invocation of the alphageometric code is specified in § 2.3.

The occurrence of the control function SO invokes the geometric primitives in code table positions 2/0 to 7/15 inclusive. The occurrence of the code function SI re-establishes the G0 set and the *space* (2/0) and *delete* (7/15) functions.

6.1.3 *Geometric primitives*

6.1.3.1 The coding scheme for the G1 set together with the code positions 2/0 and 7/15 for the geometric model is based on geometric primitives. Each drawing primitive is specified in terms of Cartesian coordinates to describe the positions, end-points, or vertices of each drawing operation.

6.1.3.2 Geometric drawings are defined in terms of the drawing primitives: *point*, *line*, *arc*, *rectangle*, and *polygon*.

6.1.4 *Drawing position*

6.1.4.1 Drawings are positionally independent; therefore drawing primitives may overlay each other redefining the drawing at the position.

6.1.5 *Drawing space*

6.1.5.1 Space for geometric drawing operations consists of a rectangular area entirely visible on the display screen. Any area of the display screen outside of the valid drawing area is termed a *border area* and it is not possible to specify a coordinate position in a border area.

6.1.6 *Picture element*

6.1.6.1 The Cartesian coordinate grid is made up of square picture elements (pixels).

6.1.7 *Picture resolution*

6.1.7.1 Any number of picture elements may be implemented. Hence, picture resolution is at the discretion of terminal manufacturers.

6.1.8 *Coordinate system*

6.1.8.1 The coordinate specifications are defined based on a Cartesian 0 to 1 numbering scheme.

6.1.8.2 The numbering system is referenced to the visible valid drawing area and consists of coordinates ranging from 0 to 1 on both the X and Y axes, with coordinate values being specified as fractions of this range.

6.1.8.3 The coordinates are encoded in 2's complement notation and specified as signed numbers to a minimum accuracy of 9 bits, including the sign bit. Increased accuracy is obtained by additional increments of 3 bits. Unused least significant bits are truncated when the coordinates are defined to a greater accuracy than can be handled by the terminal.

6.1.8.4 Display screens with non-square visible areas map into the square drawing area number system so that the origin (0,0) remains in the lower left-hand corner. On a television-like display with a 4 : 3 aspect ratio, this corresponds to a range of 0 to 0.999 ... in the X axis and 0 to approximately 0.75 in the Y axis. Drawing commands addressing the entire square 0 to 1 grid are permissible, but only the circumscribed 4 : 3 area is visible.

6.2 *Drawing command*

6.2.1 *General*

6.2.1.1 Drawing commands consist of *operational codes* (opcodes) and their associated data parameters.

6.2.1.2 Opcodes describe the types of drawing operation.

6.2.1.3 Following the opcode byte are one or more blocks of additional bytes of data to describe one or more (X, Y) coordinate positions. Each block of data for the (X, Y) coordinates may contain 3 bytes (9 bits accuracy), 4 bytes (12 bits accuracy), etc., depending on the degree of resolution desired.

6.2.1.4 Figure 9/T.100 is the code table for the opcodes and data bytes or status sub-commands.

6.2.2 *Opcode byte*

6.2.2.1 The structure of the opcode byte is as shown in Figure 10/T.100.

6.2.3 *Opcode definitions*

6.2.3.1 *Point*

Sets the drawing beam to any position in the display space and optionally draws a point.

6.2.3.2 *Line*

Draws a line based on the two given end points.

6.2.3.3 *Arc*

Draws a circular arc based on three points, which are the start point, a point on the arc and the end point of the arc. A circle results when the start and end points are coincidental and the point on the arc defines the opposite end of the diameter. The arc may be either in outline or the area enclosed by the arc and the chord may be filled.

6.2.3.4 *Rectangle*

Draws a rectangle based on specified width and height. The rectangle may be in outline or a filled-in area.

6.2.3.5 *Polygon*

Draws a closed polygon of arbitrary shape specified by the vertices. The polygon may be in outline or a filled-in area. The maximum number of vertices is limited to 256.

6.2.3.6 *Spare*

An opcode available for future definition.

6.2.3.7 *Reserved*

An opcode reserved for a specific future application.

6.2.3.8 *Control*

Provides control over the modes or attributes of the drawing commands.

6.2.4 *Opcode facilities*

6.2.4.1 Each opcode has four variants; these are defined by the facility bits (b2 and b1) as shown in Figure 11/T.100. Facility field interpretations are as given below.

<table>
<tr><td></td><td></td><td></td><td></td><td>b7</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td></td><td></td><td></td><td></td><td>b6</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td></td><td></td><td></td><td></td><td>b5</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td><td></td><td rowspan="4">Spare</td><td rowspan="4">Rect 4</td><td rowspan="16" colspan="4">Numeric data (for opcodes) or status commands</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td></td><td></td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td><td></td><td></td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3</td><td></td><td></td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4</td><td></td><td></td><td rowspan="4">Point 1</td><td rowspan="4">Poly 5</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>5</td><td></td><td></td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>6</td><td></td><td></td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>7</td><td></td><td></td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>8</td><td></td><td></td><td rowspan="4">Line 2</td><td rowspan="4">Reserved 6</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>9</td><td></td><td></td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>10</td><td></td><td></td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>11</td><td></td><td></td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>12</td><td></td><td></td><td rowspan="4">Arc 3</td><td rowspan="4">Control 7</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>13</td><td></td><td></td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>14</td><td></td><td></td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>15</td><td></td><td></td></tr>
</table>

CCITT-44160

FIGURE 9/T.100

Operation code and data field assignments

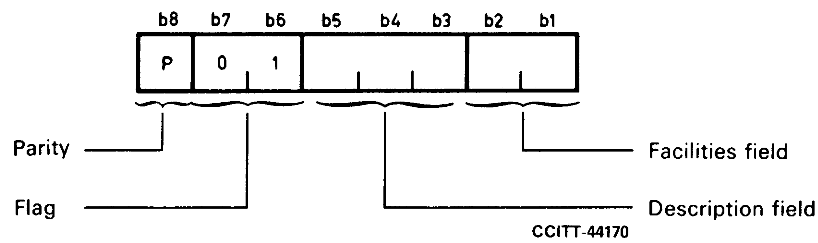


FIGURE 10/T.100

8-bit opcode byte

Opcode	Parity	Flag		Descriptor field			Facility field			
							b2		b1	
	b8	b7	b6	b5	b4	b3	0	1	0	1
Spare	P	0	1	0	0	0	–	–	–	–
Point	P	0	1	0	0	1	INVIS	VIS	ABS	REL
Line	P	0	1	0	1	0	JOIN	SET	ABS	REL
Arc	P	0	1	0	1	1	JOIN	SET	OUTLINE	FILL
Rectangle	P	0	1	1	0	0	JOIN	SET	OUTLINE	FILL
Polygon	P	0	1	1	0	1	JOIN	SET	OUTLINE	FILL
Reserved	P	0	1	1	1	0	–	–	–	–
Control	P	0	1	1	1	1				

INVIS Invisible
VIS Visible
ABS Absolute
REL Relative

FIGURE 11/T.100

Opcode facilities

6.2.4.2 *b2 is binary 1*

a) *Point* – A visible point is drawn on the display screen.

b) *Line, arc, rectangle, polygon* – The initial drawing position is specified within the data bytes as absolute (X, Y) coordinates, i.e. the initial point is *set.*

6.2.4.3 *b2 is binary 0*

a) *Point* – An invisible point is located on the display screen.

b) *Line, arc, rectangle, polygon* – The initial drawing position is the same point as the final drawing position of the previous opcode, i.e., the current drawing is joined to the previous drawing.

6.2.4.4 *b1 is binary 1*

a) *Point* – The (X, Y) coordinates are relative displacements to the preceding coordinate specifications.

b) *Line* – The (X, Y) coordinates for the final drawing position of a line segment are relative displacements from initial drawing position of that line segment.

c) *Arc, rectangle, polygon* – The areas established are filled or crosshatched.

6.2.4.5 *b1 is binary 0*

a) *Point* – The (X, Y) coordinates of the point are absolute values.

b) *Line* – The (X, Y) coordinates of the final drawing position of the line segment are absolute values.

c) *Arc, rectangle, polygon* – The drawings are outlined.

6.3 *Opcode numeric data*

6.3.1 The numerical data bytes associated with an opcode immediately follow the opcode byte and are recognized when the flag bit (b7) is binary 1. Any number of blocks of data bytes defining pairs of coordinates or drawing displacements may follow the drawing opcode until one of the following conditions occurs:

a) when another opcode is encountered;

b) when the *shift-in* code (SI) is encountered;

c) when the *shift-out* code (SO) is encountered;

d) when the *single-shift* codes (SS2 or SS3) are encountered;

e) when an *escape* (ESC) code is encountered.

6.3.2 The minimum number of data bytes that forms a block that defines a pair of X, Y coordinates is three. The structure of the data block is shown in Figure 12/T.100.

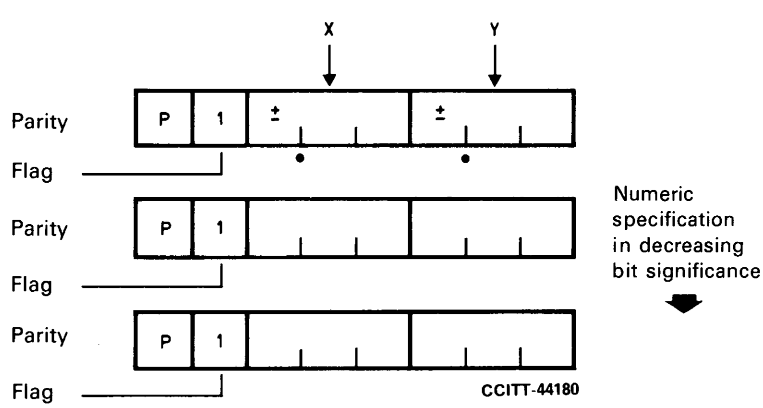


FIGURE 12/T.100

Structure of a block of 3 data bytes

6.4 *Repeated opcode operation*

6.4.1 For each of the *point, line* and *rectangle* opcodes, repeated drawing operations will automatically be effected if the numerical data field following the opcode byte contains more than one complete set of coordinate specifications. A complete set of coordinate specifications is defined as all the coordinates needed to define a *point, line* or *rectangle* drawing as a single drawing. That is, the repeated drawing feature allows concatenated drawings to be effected without having to repeat the opcode itself.

6.5 *Geometric control opcode*

6.5.1 *General*

6.5.1.1 The *control* opcodes control the drawing states of the terminal and the interpretation of the drawing opcode attributes. The sequence of *control* opcodes and their *status* sub-commands always precedes the opcodes for the geometric drawing primitives of *point, line, arc, rectangle,* or *polygon.* The controls also apply to text in *shift-in* (SI) mode. The four *control* opcodes, distinguished by the opcode facilities bits, (b2 and b1), are given in Figure 13/T.100.

Control (value)	P	0	1	1	1	1	0	0
Control (status)	P	0	1	1	1	1	0	1
Control (reserved)	P	0	1	1	1	1	1	0
Control (private)	P	0	1	1	1	1	1	1

CCITT-44190

FIGURE 13/T.100

Control codes

6.5.1.2 *Control (value)*

This control opcode defines the colour or grey scale accessed by subsequent drawing opcodes.

6.5.1.3 *Control (status)*

This control opcode provides extension to a field of sub-commands.

6.5.1.4 *Control (reserved)*

This control opcode is reserved for future control commands.

6.5.1.5 *Control (private)*

This control opcode is reserved for use by terminal manufacturers to implement proprietary non-standard functions.

6.5.2 *Attributes*

6.5.2.1 A number of drawing attributes may be applied to the drawing commands. Attributes are defined by appropriate coded sequences as described below. Once an attribute is defined, it remains valid until the attribute is redefined.

6.5.2.2 In the implementation of attributes, the level of sophistication and complexity is left to the discretion of the implementer.

6.5.2.3 For the different drawing attributes and their feature levels see Recommendation F.300.

6.5.3 *Control (value)*

6.5.3.1 This opcode specifies the colour attribute or grey scale value of the drawings (or text) that follow. Whether the *control (value)* opcode and its associated data bytes contain colour or grey scale information, is predetermined by the *tonal* status sub-command (see § 6.5.4). The number of data bytes is variable and the sequence is terminated on the appearance of another opcode. Less significant bits for colour or grey scale information are truncated where they are not used. The bit assignments of the data bytes are shown in Figure 14/T.100 (only the 6-bit data portion of the 8-bit byte is shown).

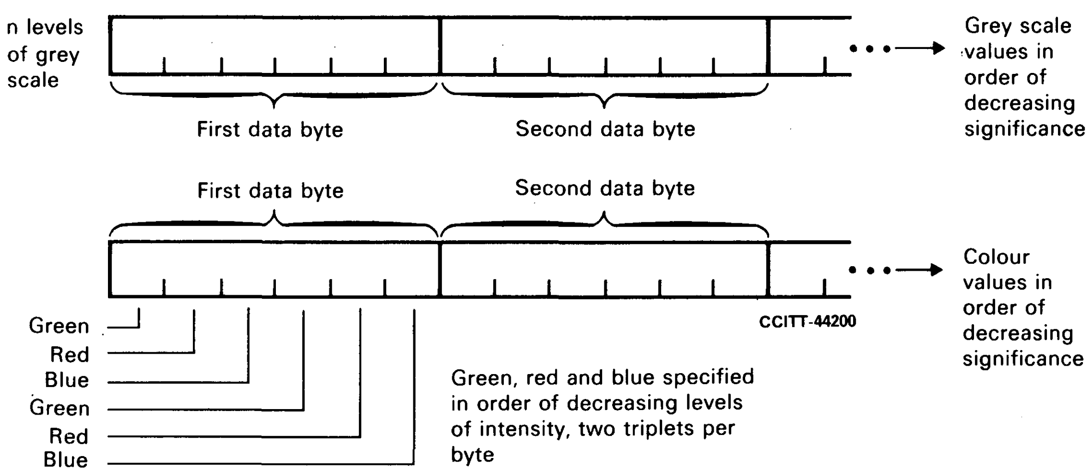


FIGURE 14/T.100

Bit assignments for grey scale or colour attributes

6.5.4 *Control (status) and status sub-commands*

6.5.4.1 The *control (status)* opcode accesses a field of *status sub-commands* (columns 4, 5, 6 and 7) which define in detail all the modes of drawing operation or attributes. The sequence is always *control (status)* followed by a *status sub-command,* which in turn may or may not be further followed by parameter data bytes. Figure 15/T.100 gives the codings of the *status sub-commands.* Detailed definitions of the *status sub-commands* are given below.

6.5.4.2 (4/0) *Clear-to-black*

This sub-command clears the entire display to black.

6.5.4.3 (4/1) *Clear-to-transparent*

This sub-command clears the entire display of the screen to transparent. By transparent is meant that conventional television pictures can be mixed with Videotex images or text.

6.5.4.4 (4/2) *Clear-to-black and initialize*

This sub-command clears the entire display to black and resets the terminal to the default mode.

6.5.4.5 (4/3) *Clear-to-current colour*

This sub-command clears the entire display to the colour currently specified by the *control (value)* opcode sequence.

6.5.4.6 (4/4) *Domain (3 bytes)*

The block of numerical data that follows an opcode contains 3 bytes. This is also the default condition.

6.5.4.7 (4/5) *Domain (4 bytes)*

The block of numerical data that follows an opcode contains 4 bytes.

6.5.4.8 (4/6) *Domain (5 bytes)*

The block of numerical data that follows an opcode contains 5 bytes.

6.5.4.9 (4/7) *Domain (6 bytes)*

The block of numerical data that follows an opcode contains 6 bytes.

6.5.4.10 (4/8) *Drawing (blink-off)*

Terminates the drawing (blink-on) status sub-command.

6.5.4.11 (4/9) *Reserved*

6.5.4.12 (4/10) *Drawing (blink-on) (or flashing)*

This sub-command causes the drawing (or text) that follows to flash in a repetitive manner for the purpose of drawing attention. In general, an object of any colour or grey scale may be blinked, but in some implementations, blinking may be restricted.

6.5.4.13 (4/11) *Reserved*

6.5.4.14 (4/12) *Tonal (colour)*

This sub-command designates that the *Control (value)* sequence carries colour information (see § 6.5.3).

6.5.4.15 (4/13) *Tonal (grey scale)*

This sub-command designates that the *Control (value)* sequence carries grey scale information (see § 6.5.3).

6.5.4.16 (4/14) *Reserved*

6.5.4.17 (4/15) *Reserved*

6.5.4.18 (5/0) *Line (solid)* (See Note)

This sub-command indicates that the drawing lines will be solid. This is also the default condition.

6.5.4.19 (5/1) *Line (dotted)* (See Note)

This sub-command indicates that the drawing lines will be dotted in texture.

				b_7	0	0	0	0	1	1	1	1
				b_6	0	0	1	1	0	0	1	1
				b_5	0	1	0	1	0	1	0	1
b_4	b_3	b_2	b_1		0	1	2	3	4	5	6	7
0	0	0	0	0					Clear (to black)	Line (solid)	Text format	
0	0	0	1	1					Clear (to trans-parent)	Line (dotted)		
0	0	1	0	2					Clear (to black and initialize)	Line (dashed)		
0	0	1	1	3					Clear (to current colour)	Line (dot-dashed)		
0	1	0	0	4					Domain (3 bytes)	Fill		
0	1	0	1	5					Domain (4 bytes)			
0	1	1	0	6					Domaih (5 bytes)	Fill (black highlight)		
0	1	1	1	7					Domain (6 bytes)			
1	0	0	0	8					Drawing (blink-off)			
1	0	0	1	9								
1	0	1	0	10					Drawing (blink-on)			
1	0	1	1	11								
1	1	0	0	12					Tonal control (colour)	Wait (timed)		
1	1	0	1	13				Control status	Tonal control (grey)	Wait (indef-inite)		
1	1	1	0	14								
1	1	1	1	15								

CCITT-44210

FIGURE 15/T.100

Control (status) and Status sub-commands assignment

6.5.4.20 (5/2) *Line (dashed)* (See Note)

This sub-command indicates that the drawing lines will be dashed in texture.

6.5.4.21 (5/3) *Line (dot-dashed)* (See Note)

This sub-command indicates that the drawing lines will be dot-dashed in texture.

Note – The line texture pattern is referenced to the absolute coordinate grid of the display screen so that the texture pattern aligns between drawing commands.

6.5.4.22 (5/4) *Fill*

This sub-command fills the enclosed area drawn in the colour specified by the current *Control (value)* sequence.

6.5.4.23 (5/5) *Reserved*

6.5.4.24 (5/6) *Fill (border highlight black)*

This sub-command fills enclosed area drawn as § 6.5.4.22 above and the circumscribing border is highlighted in black.

6.5.4.25 (5/7) *Reserved*

6.5.4.26 (5/8) *Reserved*

6.5.4.27 (5/9) *Reserved*

6.5.4.28 (5/10) *Reserved*

6.5.4.29 (5/11) *Reserved*

6.5.4.30 (5/12) *Wait (timed)*

This sub-command causes a delay of a specific time in processing and display. The length of wait is specified in tenths of a second, either by one associated parameter byte (6 bits for up to 6.3 s) or two parameter bites (12 bits for up to 6.8 m).

6.5.4.31 (5/13) *Wait (indefinite)*

This sub-command causes an indefinite wait. This may be achieved by the terminal responding with a *pause flow* control character (DC3 in C0 set) towards the computer. The wait is then terminated when the terminal sends a *resume data flow* character (DC1 in C0 set).

6.5.4.32 (5/14) *Reserved*

6.5.4.33 (5/15) *Reserved*

6.5.4.34 (6/0) *Text format*

This sub-command has an associated data byte, which defines the text formats as follows:

Bit b6 = 0: Free format, i.e. character strings are wrapped around on the right margin.

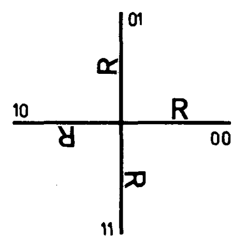
Bit b6 = 1: Annotation format, i.e. character strings are in fixed positions on the screen.

Bit b5 = 0: In free format, character strings are broken on a character boundary.

Bit b5 = 1: In free format, character strings are broken on a word boundary.

b4, b3: Defines character rotation as shown in Figure 16/T.100. Rotated strings of characters proceed in the direction of rotation. However, all other format controls on characters such as APB, APF, APD, APU and APR have their (unrotated) orientation meanings.

b2, b1 = 0.0: Vertical spacing = 1.0
b2, b1 = 0.1: Vertical spacing = 1.5
b2, b1 = 1.0: Vertical spacing = 2.0
b2, b1 = 1.1: Vertical spacing = 2.5



CCITT-44220

FIGURE 16/T.100
Character rotation

6.6 *Default conditions*

6.6.1 The default conditions of the attributes for the alphageometric coding scheme are summarized below:

			Reference
1)	Control (value):	White	§ 6.5.3
2)	Tonal control:	Tonal (colour)	§ 6.5.4.14
3)	Domain:	3 bytes (9 bits)	§ 6.5.4.6
4)	Drawing:	Blink-off	§ 6.5.4.10
5)	Line control:	Solid line	§ 6.5.4.18
6)	Fill:	Solid fill (no highlight)	§ 6.5.4.22
7)	Text format:	a) Free format b) Break on character boundary c) No rotation d) Vertical spacing = 1.0	§ 6.5.4.34 with bits 1 to 6 set to "0"

7 Alpha-dynamically redefinable character sets (DRCS) option

7.1 *General*

7.1.1 A DRCS is a set of characters whose shapes are sent from the data-base and down-loaded via the line. It may be used to represent alphabetic characters, special symbols, or picture element symbols for constructing fine graphics. Once loaded, the DRCS are regarded as members of a library that can be designated by appropriate ESC sequences as G0, G1, G2, G3 sets. Several schemes for the DRCS option are possible. One scheme is described in § 7 in the context of a general architecture. When used in its alphanumeric mode DRCS may be employed as a part of the alphabetic representations of any other Videotex option and in that case, the attributes associated with that option are to be used.

7.2 *General architecture for down-loading DRCS*

7.2.1 *Initiation*

The down-loading process is initiated by a designation and invocation sequence. This sequence is followed by one or more of the following functions.

7.2.2 *Identification of character set (ICS)*

This function must immediately follow the initiating sequence. It identifies the *escape* sequence used for the designation of the character set.

7.2.3 *Select coding method (SCM)*

This function defines the type of coding used to describe the DRCS character.

7.2.4 *Select dot composition (SDC)*

This function defines the number of bits horizontally and vertically in a character matrix, the number of bits per pixel, the number of grey scale levels and the number of colours accessible within a character position.

7.2.5 *Pattern transfer (PT)*

This is the active part of the down-loading process. It defines the code location of the first character and provides instructions and data to draw characters. It may also incorporate an error checking procedure.

7.2.6 *Down-loading termination procedure (DLT)*

The down-loading process is terminated by a specific procedure, which may include acknowledgement.

7.3 *A possible coding scheme for the DRCS option*

7.3.1 *Initiating sequence*

The initiating sequence is ESC F_s followed by x bytes indicating the length of the loading data block, where x is for further study.

7.3.2 *Termination procedure*

The down-loading process is terminated by means of counting the length of the loading data block. See § 7.3.1.

7.3.3 *Designation and invocation of loaded DRCS*

7.3.3.1 Once loaded into the terminal, the DRCS is placed into a library. This library is used in the context of ISO 2022 in the 7-bit environment as implemented in earlier sections. Before invoking the designated DRCS, it is required to designate a C1 set to be associated with it. For the scheme described herein any of the C1 sets (to be registered) that are defined in §§ 2.2 and 2.3 may be used.

7.3.3.2 The designation sequence will be of the form ESC I_1, 2/0, ($I_3 \dots I_n$) F. I_1 will be 2/8, 2/9, ... or 2/15. $I_3 \dots I_n$ are optional, and if present together with F, will identify the set. Means for associating the designating sequence with the process of defining the character shapes will be for further study.

8 **Alphaphotographic option**

8.1 The alphaphotographic option is used to render an image by the transmission and display of individual picture elements.

8.2 This option may include both continuous-tone images such as pictures of faces, etc., as well as pattern-oriented techniques for the display of pictures, including graphics, Latin and non-Latin characters for text, etc. The system features and attributes include colour and monochrome.

8.3 The detailed system proposals are for further study.

9 Service enhancements

9.1 *Introduction*

9.1.1 Many Administrations are offering or considering the introduction of a Videotex service, and it is recognized that this Recommendation may influence some of their decisions. While the other sections of this Recommendation contain details of those aspects of an international Videotex service that could be agreed upon, this § 9 identifies certain potential enhancements (features or attributes) that some Administrations believe need to be considered in future developments.

9.1.2 It is recognized that some of these potential enhancements may only exist on national Videotex services, while others may have international application. However, an enhancement that begins on a national service only could become international in the future. Therefore, it is considered desirable to have international coordination of future enhancements.

9.2 *General*

9.2.1 The growth of international Videotex services during the years following the publication of this Recommendation will be greatly affected by the specific specifications contained in the other parts of this Recommendation. However, some Administrations believe that experiments with and/or implementation of certain enhancements will allow the development of an international service that provides a range of capabilities that will maximize the desirability and utilization of Videotex service.

9.2.2 Some of the potential enhancements to Videotex service, national or international, are presented in the following. This is for the purpose of identifying to interested Administrations those enhancements that warrant serious consideration in the view of the CCITT, but which presently lack enough details to obtain the full agreement of all Administrations.

9.2.3 The enhancements have been grouped into three categories in order to assist the reader in understanding the application of each individual enhancement (which may be referred to by some Administrations as attributes or features or some other descriptive phrase) and to prompt an orderly investigation of them:

a) display-related enhancements;

b) transmission-oriented enhancements;

c) system level enhancements.

9.3 *Display related enhancements*

9.3.1 Most of the currently planned and/or offered services utilize images created with only eight colours, which are formed by the various combinations (on or off) of three primary colours – red, green and blue. Limiting Videotex to eight colours is an unnecessary restriction, since the electronic emission devices controlling the red, green and blue colours can be caused to have more than just the two states of on or off. For example, with just eight different states or levels, a potential of 512 colours exist. Additionally, for those services that use a matrix-oriented screen (e.g. a mosaic graphic mode), different colours could be identified for foreground symbols to those for background areas.

9.3.2 The ability to simulate motion (i.e. animation) is a potential enhancement that can be achieved by several means. These include:

a) alternating between slightly different display frames stored in the terminal;

b) dynamically altering the colour of portions of the display image, making them appear or disappear by redefining the colour table (an image disappears when its colour is set to the same colour as the surrounding area);

c) execution of a resident program to redefine the image at a controlled rate.

9.3.3 The flashing of symbols or areas of the display has typically been limited to changing the foreground symbol (in the case of a matrix-oriented screen) to the background colour, momentarily, or some other single-state change. An enhanced flashing capability could allow for different rates of change and for various conditions associated with each change (e.g. colour X to colour Y, rather than foreground colour to background colour or foreground colour to black).

9.3.4 Different pictorial (text and graphic) symbols may be developed that extend the repertoire of a Videotex service. This may be a fixed extension defined in the terminal memory, or can be a modification to the existing memory by downloading from the data base. The range of extended symbols includes different fonts of existing symbols, smoothed mosaic graphics, or other unique symbols.

9.4 *Transmission oriented enhancements*

9.4.1 The exchange of information directly between terminals, without communicating with a Videotex service may be permitted by some Administrations as an enhanced capability, and could be of value to the users of Videotex terminals. Such a capability would require the existence of control functions that might not, otherwise, be available in some terminals that utilize certain existing or planned national Videotex services, but this should not cause any incompatibilities with such services.

9.4.2 The optimization of the coded character stream for maximum data rate is a valuable enhancement. This might be accomplished by utilizing an 8-bit per word coding format rather than the 7-bit per word format currently planned by most Administrations, coupled with a related decision on the line or link level protocol selected. The selection of an 8-bit per word format could permit a more efficient transmission of data.

In addition, such techniques as run-length-encoding might be specified in the Recommendation to reduce the transmission of unnecessary or redundant data. The choice of higher speed modems/circuits is also considered by some Administrations as a way to optimize the transfer of data within or between Videotex services.

9.4.3 For some applications of a Videotex service, sophisticated error detection and correction schemes may be required and should be considered with other transmission-oriented enhancements on future Videotex services.

9.5 *System level enhancements*

9.5.1 An enhancement seriously considered by some Administrations is the provision of a Videotex service that provides visual information, augmented by audio information. This capability could permit access by a terminal to visual-only information in a data base, and to visual/audio information in the same or other data base. The audio information might be associated with the visual information, or treated separately, or even alternately, depending upon the implementation. The audio information might be analogue or digitally encoded or handled as a composite signal.

9.5.2 The provisioning of peripheral input/output devices associated with the Videotex terminal is an important enhancement for future services. These could include magnetic storage devices for recording visual/audio information as received by the terminal, or recorded locally by the terminal for subsequent transmission to a data base or other terminal. Various hard copy printing devices could also be provided, with their design based upon the specific visual capabilities of the terminal, e.g. degree of resolution and colour of the image on the display screen.

10 Line and end-to-end protocols

10.1 The purpose of § 10 is to describe the protocols needed for international Videotex transactions. Section 10 contains an introduction only. Detailed consideration is left for further study.

10.2 The transfer of information from a data base of one service to a user of another service may be split up into two parts:

a) the information transfer from one service to another;

b) the information transfer from the service to the user.

10.3 *Line protocols*

10.3.1 *Line protocols between services*

10.3.1.1 The international line between national data base computers must be able to transmit transparent coding schemes identified in this Recommendation and accept the protocols of § 10.4.

10.3.2 *Line protocols between service and user*

10.3.2.1 The following protocol functions should be studied:

PF1: Start of coded data starts a sequence of data to be understood as textual information (could be coded as STX).

PF2: Start of prefix causes the following bytes to be understood as a prefix containing framing information including codes for error check and/or correction (could be coded as SOH).

PF3: End of coded data ends a sequence of data to be understood as textual information (could be coded as ETX).

PF4: End of frame. Ends a frame of data and requests for reverse transmission and give an answer (could be coded as ETB).

PF5: Answer given in case of error free reception or when error correction is possible (could be coded as ACK).

PF6: Answer given in case of errors when no error correction is possible (could be coded as NAK).

10.3.2.2 It is noted that TC1 to TC10 (SOH to ETB of Recommendation T.50) are intended to control the transmission of information over transmission networks. The use of these functions may therefore not be used as part of the information stream from one service to another.

10.3.2.3 The use of protocol functions is for further study.

10.4 *Protocols for communication between services on the application level*

10.4.1 *General*

10.4.1.1 International exchange of information between national Videotex services may be sent in blocks, here called messages. For efficient use of networks and communication equipment it is important to design the messages to minimize the capacity needed for applications that are frequently used in Videotex services.

10.4.2 *Types of message elements*

10.4.2.1 A complete message is composed of message elements. Each element contains an element identifier, a data field and an indication of element length (explicit or implicit).

10.4.2.2 *Transmit a standardized function*

Codes for functions may be different from the character sequences, sent by the user.

10.4.2.3 *Transmit a service message*

A service message is a frame that is transmitted to the subscriber, without erasing the screen, moving the active position of the cursor, or changing the contents of the previous display.

10.4.2.4 *Transmit a service message code*

The proper service message is generated by the receiving system and transmitted to the subscriber.

10.4.2.5 *Transmit a frame*

Billing and other additional information is to be transmitted together with the frame.

10.4.2.6 *Transmit data block*

By data is meant all types of data that are not listed under separate items, e.g. software. It is necessary to transmit block length when transmitting transparent data.

10.4.2.7 *Transmit field description*

A field description is a list of positions on the screen, where an application program expects additional information to be filled in, either by the user or by the application program itself. It includes also format and type of information which allows simple syntax control in the host computer.

Three formats are recognized; strings, which means any combination of graphical characters including space, integers (0-9), and free format.

A field may be of input and/or output type. An input field is a field where the information is user originated. An output field is a field in which the information is filled in by the application program.

10.4.2.8 *Transmit a user message to an application*

A user message is the data that is filled in by the user according to a field description. It is sent to the external computer. The transmission is initiated either by a send-function if it is available, or when all input fields are filled. The use of a delimiter causes the rest of the field to be filled with spaces. If a delimiter is used in the first position of a combined input and output field, the contents remain unchanged.

10.4.2.9 *Transmit an application message*

An application message is a block of data to be filled into the output fields, defined by a field description. It may be sent either in the same message as the field description, or after.

10.4.2.10 *Request information on terminal capability*

(For further study.)

10.4.2.11 *Transmit information on terminal capability*

(For further study.)

10.4.2.12 *Error condition element*

The detection of contradicting information in a system will result in an error condition message to the other system, e.g. data with a format different to the corresponding field description. The entire message causing the error will be ignored, and it is the responsibility of its transmitting system to handle the error properly.

10.5 *User to data base protocol*

10.5.1 In order to use Videotex service, a user must be able to generate a set of functions which enables him to access and use different applications. A set of user functions is listed in Recommendation F.300.

10.5.2 The minimum set of characters to code these functions contains the digits 0-9 and two other symbols. For some applications however, the generation of alphanumeric as well as pictorial and attribute information and other control characters may be needed.

10.5.3 Although it is desirable that all Videotex services employ the same keying sequences and visual identifiers for these functions, there are historical reasons why there will be different manners of coding the same user functions.

10.5.4 Accessing the national service of another country using an international connection between services is possible, if the user obeys the function coding rules of the service of the other country. It is, however, possible that the local data bank may be able to translate the local keying sequence into the appropriate command in a national service level (see § 10.4.2.4). This subject is left for further study.

11 Interworking with other services

11.1 *Telex-Videotex*

11.1.1 Telex is a message transfer service and therefore, interworking between telex and Videotex should be limited to the exchange of alphanumeric text between terminal equipments.

11.1.2 Only the graphic characters of the Videotex graphic character repertoire corresponding to International Telegraph Alphabet No. 2 should be used to compose messages.

11.1.3 The message format will be limited by the Videotex page format.

11.1.4 Telex can only display alphanumeric information without the capability of displaying the other attributes of Videotex.

11.2 *Teletex-Videotex*

11.2.1 *Graphic character repertoire*

11.2.1.1 The Teletex and Videotex graphic repertoires are largely identical. The following fallback representations of Videotex characters (see Table 2/T.100), may be transcoded at a Videotex-Teletex interworking facility.

TABLE 2/T.100

Identifier	Videotex character	Fallback representation	
SM30	←	<	SA03
SM31	→	>	SA05
SM32	↑	¡	SP03
SM33	↓	!	SP02
SP19	‘	'	SP05
SP20	’	'	SP05
SP21	“	"	SP04
SP22	”	"	SP04
SM12	—	-	SP10
MG01 to MG63	Block graphics	/	SP12

11.2.1.2 For Teletex terminals having the ability to present the Videotex character repertoire in its entirety, the need for this transcoding disappears. Therefore, on initial call establishment, a determination of the terminal display/printing capabilities must be made by handshaking.

11.2.2 *Control functions*

11.2.2.1 Transcoding of the Videotex attribute control functions is for further study.

11.2.3 *Format*

11.2.3.1 Interworking between Videotex and Teletex will be limited to the Videotex display frame format.

11.3 *Videotex-facsimile*

(For further study.)

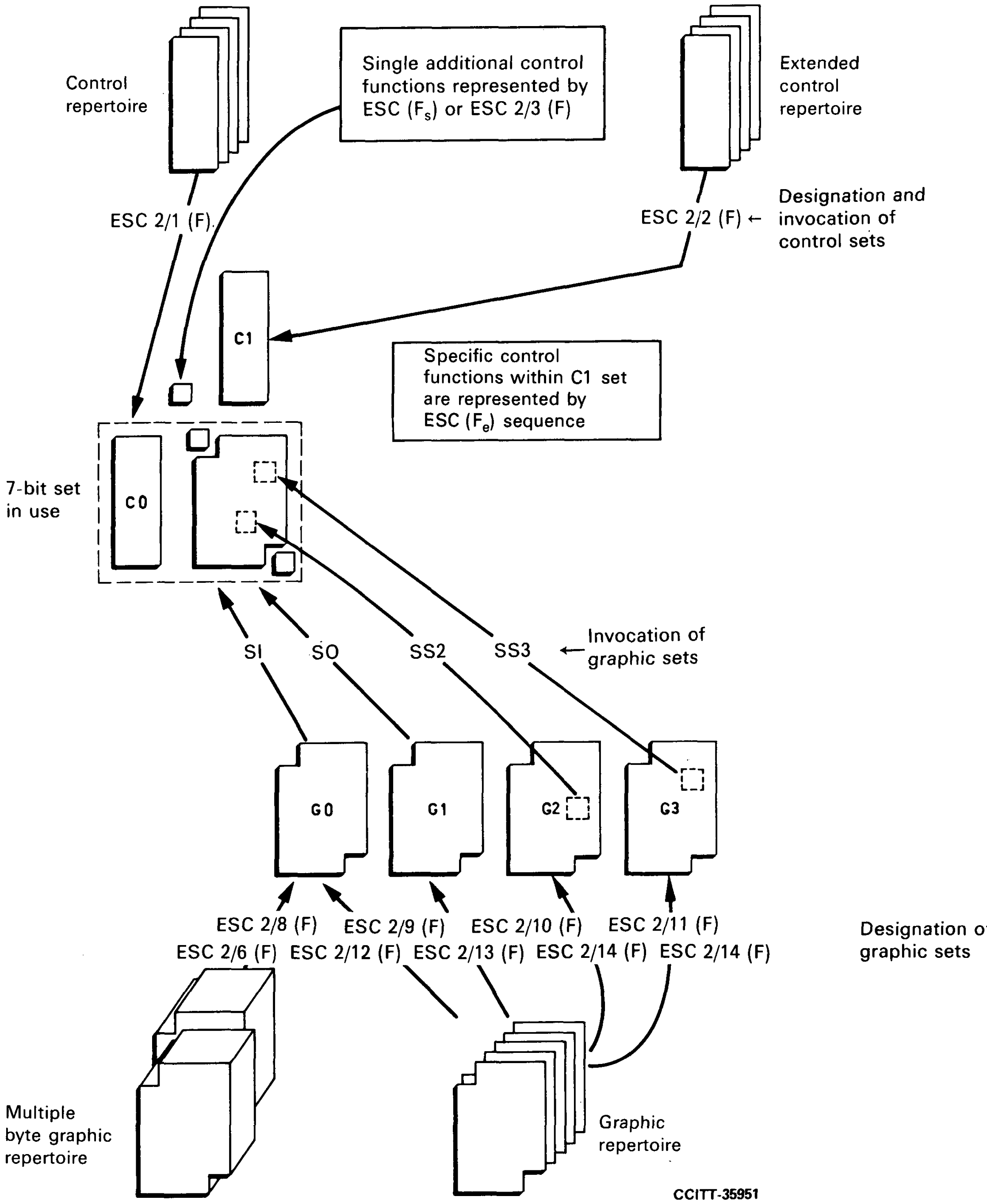
11.4 *Videotex-Teletex*

(For further study.)

ANNEX A

(to Recommendation T.100)

Part of the code extension scheme of ISO 2022



CCITT-35951

ANNEX B

(to Recommendation T.100)

Repertoire of graphic characters

B.1 *General*

B.1.1 This annex defines the basic graphic repertoire of the international Videotex service. This repertoire consists of the total range of non-pictorial symbols, which may be communicated between Videotex services and terminals by means of coded character sets for Latin-alphabet based languages.

B.1.2 The repertoire of graphic characters defined in this part of the Recommendation consists of:

a) Latin alphabetic characters, listed in § B.2, which comprise:
   - the 52 small and capital letters of the basic Latin alphabet,
   - combinations of basic Latin letters and diacritical marks,
   - special alphabetic characters, which are neither basic Latin letters nor combinations of basic Latin letters and diacritical marks,

b) non-alphabetic characters, listed in § B.3, which comprise decimal digits, currency signs, punctuation marks, arithmetic signs and miscellaneous symbols that have individual special meanings.

B.1.3 A diacritical mark has no meaning as an individual character but is used only in combination with a basic Latin letter to form an accented letter or an umlaut.

B.1.4 The repertoire of graphic characters defined in this part of the Recommendation contains a limited set of accented letters and umlauts.

B.2 *Latin alphabetic characters*

B.2.1 The repertoire of Latin alphabetic characters is identical to that specified in § 3.2.2 of Recommendation T.61 (for the Teletex basic repertoire of graphic characters).

B.3 *Non-alphabetic characters*

B.3.1 Decimal digits (0 to 9), currency signs, arithmetic signs, subscripts and superscripts and fractions are as specified in §§ 3.2.3.1, 3.2.3.2, 3.2.3.4, 3.2.3.5 and 3.2.3.6 of Recommendation T.61.

B.3.2 Punctuation marks are as specified in § 3.2.3.3 of Recommendation T.61, with the exclusion of SP09 (low line) and the addition of SP19 to SP22, which are as shown in Table B-1/T.100.

B.3.3 Miscellaneous symbols are as shown in Table B-2/T.100.

B.3.4 The lists in Tables B-1/T.100 and B-2/T.100 are composed as described in the following.

The first column contains the identifier of each character, assigned in accordance with the identification system explained in Annex C of Recommendation T.61.

The second column presents the graphical representation of the character.

The third column specifies the name or the description of the character.

TABLE B-1/T.100

Punctuation marks

Identifier	Graphic	Name or description
SP19	‘	Single quotation mark left
SP20	’	Single quotation mark right
SP21	“	Double quotation mark left
SP22	”	Double quotation mark right

Note – In Videotex (and Teletex), *quotation mark, apostrophe* and *comma* are independent characters that cannot have the meaning of diacritical marks.

TABLE B-2/T.100

Miscellaneous symbols

Identifier	Graphic	Name or description
SM01	#	Number sign
SM02	%	Percent sign
SM03	&	Ampersand
SM04	*	Asterisk
SM05	@	Commercial at
SM12	—	Horizontal bar
SM13	|	Vertical line
SM17	µ	Micro sign
SM18	Ω	Ohm sign
SM19	°	Degree sign
SM20	º	Ordinal indicator, masculine
SM21	ª	Ordinal indicator, feminine
SM24	§	Section sign
SM25	¶	Paragraph sign, pilcrow
SM26	·	Middle dot
SM30	←	Leftward arrow
SM31	→	Rightward arrow
SM32	↑	Upward arrow
SM33	↓	Downward arrow

INTERNATIONAL INTERWORKING FOR VIDEOTEX SERVICES

(Malaga-Torremolinos, 1984; amended at Melbourne, 1988)

CONTENTS

Preamble

The CCITT,

considering

(a) that Videotex services have been implemented in different countries/regions using different data syntaxes referred to as data syntax I, data syntax II and data syntax III, which have an equal status;

(b) that the CCIR is studying standards for broadcast Teletext services for general reception and has expressed a view that it is desirable that terminal equipment compatibility should exist between broadcast Teletext systems for general reception and public network-based data base systems;

(c) that different countries/regions are entitled to use their existing systems;

(d) that interworking between Videotex services in different countries may require transcoding and/or conversion;

(e) that the interworking between Videotex services may be provided by using different types of networks such as the public switched telephone network (PSTN), packet switched public data network (PSPDN), circuit switched public data network (CSPDN), integrated services digital network (ISDN), etc.;

(f) that Videotex interworking protocols should offer a large degree of compatibility with protocols used in other telematic services,

recommends

that the following technical provisions be applied for international interworking for Videotex services.

[1)] *Note* – Annexes B, C and D will not be published in Fascicle VII.5 (T-Series Recommendations) but will be issued as a separate publication.

1 Purpose and scope of the Recommendation

1.1 *Purpose*

The purpose of this Recommendation is:

a) to facilitate the interworking of different Videotex services;

b) to identify parameters needed to communicate with Videotex terminals;

c) to provide technical recommendations desirable for potential interworking of other telematic services with Videotex services.

1.2 *Scope*

1.2.1 This Recommendation describes the characteristics of coded information that is exchanged between countries participating in the international interactive Videotex service.

1.2.2 Videotex systems are text communication systems having in addition the capability of a given level of pictorial representation and a repertoire of display attributes. The text and the pictures obtained are intended to be displayed using the current television (TV) raster standards of the different countries.

1.2.3 Different data syntaxes are offered as a choice for the Administrations to implement their national services. Substantial degrees of compatibility exist between these options, but some transcoding and/or conversion may be necessary to facilitate interworking.

1.2.4 For the purpose of the international service, different data syntaxes have been identified:

a) interworking data syntax;

b) data syntax I;

c) data syntax II;

d) data syntax III;

e) other syntaxes are for further study.

2 Interworking between Videotex services – General

2.1 It is the possibility of Administrations to decide in which network(s) the Videotex service(s) are to be provided.

2.2 Serveral possibilities are considered below:

2.2.1 Videotex service operated on the PSTN; the communication between a Videotex terminal and a Videotex host computer is established over the PSTN.

2.2.2 Videotex service operated on the PSTN and a public data network (PDN) (generally a PSPDN); the communication between a Videotex terminal connected to the PSTN and a Videotex host computer connected to a PDN is established via a Videotex access prior or a Videotex service center interfacing between both networks.

2.2.3 Other possibilities (CSPDN, ISDN, etc.) could also be considered.

2.3 International interworking between Videotex services via gateways and connected to any network (PSTN, PSPDN, CSPDN, ISDN, etc.) may be possible. Such interworking allows a Videotex terminal pertaining to a Videotex service to access a Videotex host computer pertaining to another Videotex service. International interworking between Videotex terminal in one country and a Videotex host in another country may also be possible. All international data exchange should comply with the specifications contained in this Recommendation. (See Recommendation F.300 for the service description).

3 International interworking of Videotex service

3.1 Videotex interworking allows a Videotex terminal in a given country to interact in real time with Videotex application located in a different country.

3.2 International interworking between Videotex services should use those functions that are defined in the data syntaxes implemented by the Administrations concerned: data syntaxes I, II and III defined in Annexes.

3.3 *International interworking configurations*

The various configurations for international interworking are defined in Recommendation F.300. The two major classes of interworking are defined below.

3.3.1 *Gateway to gateway interworking*

This class of interworking involves communication between gateways located in each country and where all the data handling processes involved by the interworking are performed. The protocols and data syntaxes for this class of interworking are specified in § 4.

3.3.2 *Terminal to host interworking*

This class of interworking involves communication between a terminal and a host located in different countries, either directly or through a conversion unit situated in the country where the terminal is located. Several cases have been identified. The protocols and data syntaxes for the various cases of this class of interworking are specified in § 5.

4 International interworking between gateways

The international interworking between gateways allows a Videotex terminal located in country A to access the Videotex services located in country B via a Videotex service of country A. The configuration for the international interworking between gateways is described by Figure 1/T.101:

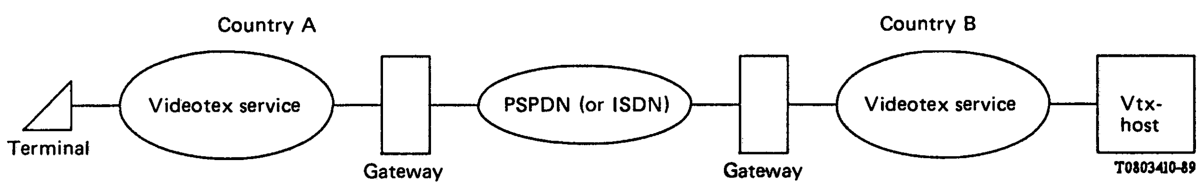


FIGURE 1/T.101

4.1 *International interworking at network level*

4.1.1 International interworking between Videotex services should preferably take place between networks of the same type when these networks are provided by both Administrations involved (PSPDN, CSPDN, ISDN and leased lines).

4.1.2 The network service definition of open systems interconnection for CCITT application is defined in Recommendation X.213.

4.1.3 When the interworking takes place between Videotex services operated on different types of network, Recommendation X.75 should apply. Interworking with ISDN should be in accordance with Recommendation T.90.

4.2 *Transport layer*

The transport layer service of open systems interconnection for CCITT applications is defined in Recommendation X.214.

The transport protocol of open systems interconnection for CCITT applications is specified in Recommendation X.224.

Both classes 0 (corresponding to Recommendation T.70) and 2 may be used.

When class 0 is selected, then the protocol used is fully compatible with CCITT Recommendation T.70. When class 2 is selected, explicit flow control is to be used.

4.3 *Session layer*

This session layer service of open systems interconnection for CCITT applications is defined in Recommendation X.215. The session protocol of open systems interconnection for CCITT applications is specified in Recommendation X.225.

The use of the session protocols by Videotex interworking is defined in Recommendation T.523.

4.4 *Presentation layer*

4.4.1 *Presentation protocol*

The presentation layer service of open systems interconnection for CCITT applications is defined in Recommendation X.216. Presentation protocol of open systems interconnection for CCITT applications is specified in Recommendation X.226.

The use of the presentation protocols by Videotex interworking is defined in Recommendation T.523.

4.4.2 *Coding of Videotex information*

Coding of the contents of the display-data element

The Videotex content conforms to one of the several different data syntaxes. A data syntax, referred to as interworking data syntax, is described in Annex A. There are three existing data syntaxes, based on Recommendation T.50 and referred to as data syntax I, data syntax II and data syntax III. They are described in Annex B, Annex C and Annex D respectively. All the four annexes form an integral part of this Recommendation.

Different Administrations implementing a Videotex service may use one of the three above data syntaxes.

If two countries implement the same data syntax, then Videotex interworking between the two countries can use that same data syntax.

If one country implements one data syntax and another country implements a different data syntax, then Videotex interworking between the two countries can either:

i) use the interworking data syntax as the intermediary syntax. Transcoding/conversion into and from the interworking data syntax by the two countries will be required; or

ii) use one of the two data syntaxes with transcoding/conversion performed either at the originating or at the destination country.

For identification of the particular in-use data syntax (I or II or III), the designation and invocation of the "complete code" escape sequence may be used:

ESC 2/5 4/3 for data syntax I

ESC 2/5 4/4 for data syntax II

ESC 2/5 4/1 for data syntax III

The "complete code" environment will be terminated either by the sequence:

ESC 2/5 4/0

or by the designation and invocation of any other complete code.

4.5 *Application layer*

The association control service element (ACSE) of open systems interconnection for CCITT applications is defined in Recommendation X.217. The association control service element (ACSE) protocol of open systems interconnection for CCITT applications is specified in Recommendation X.227.

The application layer for Videotex interworking makes use of the following Recommendations:

– Recommendation T.400: Introduction to document architecture, transfer and manipulation
– Recommendation T.411: Open document architecture (ODA) and interchange format; Introduction and general principles
– Recommendation T.412: Open document architecture (ODA) and interchange format; Document structures
– Recommendation T.414: Open document architecture (ODA) and interchange format; Document profile
– Recommendation T.415: Open document architecture (ODA) and and interchange format; Document interchange format (ODIF).

The application layer for Videotex interworking makes use of DTAM (document transfer and manipulation) service and protocol described in Recommendations T.431, T.432 and T.433.

The application layer for Videotex interworking makes use of operational structures described in Recommendation T.441.

Recommendation T.564 describes the Videotex interworking application profile and the gateway characteristics.

Recommendation T.504 describes the document application profile for Videotex interworking.

Recommendation T.523 describes the communication application profile for Videotex interworking.

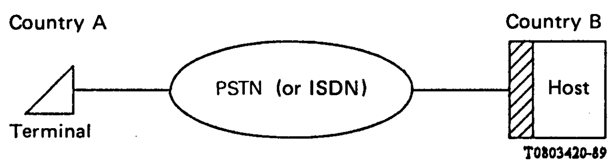
Recommendation T.541 describes the operational application profile for Videotex interworking.

4.6 *Relation with DTAM/ODA*

The relations with the document architecture (Recommendation T.412) and the document interchange format (see Recommendation T.415) are expressed through the content architecture class attributes, and the content portion attributes are described in §§ 6 and 7.

5 International interworking between a terminal and a host

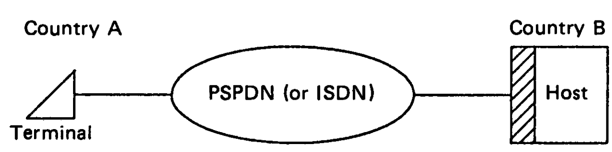
5.1 *Access via PSTN or ISDN bearer service*



In this configuration, the terminal uses the international PSTN (respectively the ISDN bearer services) to reach the host. On the international link, the following protocols should be used:

layers 1 to 3 via PSTN:	the protocols defined by the host;
layers 1 to 3 via ISDN bearer service[2]:	Recommendation T.90;
layers 4 to 7:	the protocols (if any) defined by the host located in country B;
data syntax:	data syntax defined by the host;
dialogue/service functions:	functions defined by the host.

5.2 *Access via PSPDN or ISDN bearer service*

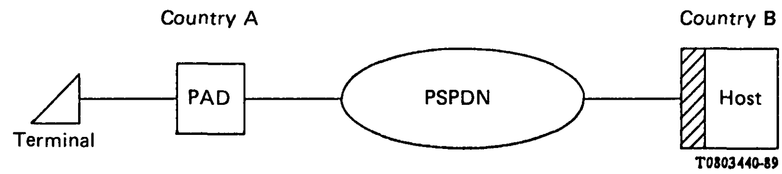


In this configuration, the terminal uses the international PSPDN (respectively the ISDN bearer services) to reach the host. On the international link, the following protocols should be used:

layers 1 to 3 via PSPDN:	Recommendation X.75;
layers 1 to 3 via ISDN bearer service[2]:	Recommendation T.90;
layers 4 to 7:	the protocols (if any) defined by the host located in country B;
data syntax:	data syntax defined by the host;
dialogue/service functions:	functions defined by the host.

[2] The protocols to be used in the ISDN Videotex teleservice are for further study.

5.3 *Access via PSPDN/PAD*



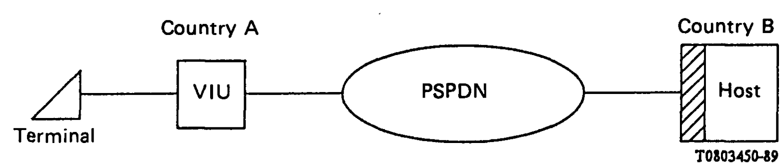
In this configuration, the terminal is connected to a PAD which gives access to the international PSPDN; both terminal and PAD are located in country A. The type of connection between the terminal and the PAD is a national matter (generally the PSTN or a leased line).

The host of country B may be accessed through the international PSPDN. The type of connection between the host and the national PSPDN is a national matter (generally a leased line).

On the international link, the following protocols should be used:

layers (1 to 3):	Recommendation X.75;
above layer 3:	Recommendation X.29 + Recommendation X.3;
data syntax:	data syntax defined by the host located in country B;
dialogue/service functions:	functions defined by the host.

5.4 *Access via PSPDN through a VIU*



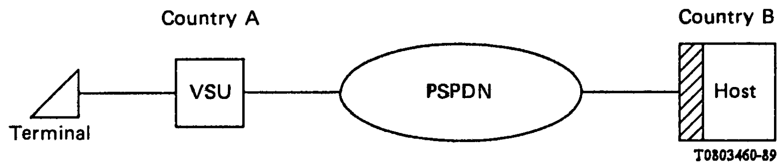
In this configuration, the terminal is connected to a VIU (Videotex interface unit) which gives access to the international PSPDN; both the terminal and the VIU are located in country A. The type of connection between the terminal and the VIU is a national matter (generally the PSTN or a leased line). The VIU performs two functions: it supports terminals and converts data syntaxes. It is up to the Administration of country A to decide how a VIU is implemented: it may be realized as a separate system or integrated with an existing equipment (PAD or Videotex access point for example).

The host country B may be accessed through the international PSPDN. The type of connection between the host and the national PSPDN is a national matter (generally a leased line).

On the international link, the following protocols should be used:

layers (1 to 3):	Recommendation X.75;
above layer 3:	Recommendation X.29 + Recommendation X.3. Alternatively Recommendation X.200 based protocols can be used. For this case, application profiles will need to be defined in T.500 Series of Recommendation. This is for further study.
data syntax:	the data syntax defined by the host located in country B;
dialogue/service functions:	those defined by the host.

5.5 *Access via PSPDN through a VSU*



In this configuration, the terminal is connected to a VSU (Videotex service unit) which gives access to the international PSPDN; both the terminal and the VSU are located in country A. A VSU is a VIU which is also in charge of handling application charge and accounting. It is up to Administration of country A to decide to set up or not a VSU and to decide how a VSU, if any, is to be implemented: it may be realized as a separate system or integrated with an existing equipment (PAD, Videotex access point or Videotex service center).

The host in country B may be reached through the international PSPDN. The type of connection between the host and the national PSPDN is a national matter (generally a leased line).

On the international link, the following protocols should be used:

layers (1 to 3):	Recommendation X.75,
above layer 3:	Recommendation X.200 based protocols. For this case applications profiles need to be defined in the T.500 Series. This is for further study.
	Alternatively Recommendation X.29 plus Recommendation X.3 may be used. Extensions (Application rules) to Recommendation X.29 are necessary (see § 5.6);
data syntax:	the data syntax defined by the host located in country B;
dialogue/service functions:	those defined by the host.

5.6 *Application rules for X.29 to support administrative functions*

When an international communication is established via a VSU using Recommendation X.29, X.29 Videotex commands may be used to allow application charges (if any) to be passed from the host to the VSU.

Videotex commands should be sent in complete packet sequences with the Q bit set to one.

Videotex commands use a T(ype)-L(ength)-V(alue) encoding. Fixed length commands do not require any length indicator. When used, length indicator is coded on two bytes and defines the total length in bytes of the V field.

In order to distinguish Videotex commands from PAD commands as currently defined by Recommendation X.29, the type value of a Videotex command is defined with the most significant bit set to one.

The following values are proposed to support the exchange of charging of international connections:

– Administration (99H) L2 charging (82H), L2 charging-parameter.

  The charging parameter may take either the value service-operation (80H) or the value application-operation (81H); both values may be present within the same charging parameter:

– service-operation (80H) L1 service-parameter

– application-operation (81H) L1 application parameter.

The service parameter is time dependent and may correspond either to the amount period (80H) or to the duration of the period (81H) or to a combination of them:

- amount (80H) L1 value
- period (81H) L1 value.

The application parameter may be frame dependent (80H), time dependent (81H) or transaction dependent (82H) or a combination of them:

- frame (80H) L1 value
- time (81H) L1 charging-on-time-parameter
- transaction (82H) L1 value.

The charging-on-time parameter is organized as the service parameter.

L1 is an acronym for a length coded on one byte.

L2 is an acronym for a length indicator coded on two bytes.

The following depicts the coding mechanism:

```
Administration L2 Charging L2 (Service) (Application)
    (99H)          (82H)

(Service) ::= Service L1 (Amount L1 value) (Period L1 value)
               (80H)       (80H)             (81H)

(Application) ::= Application L1 (Frame) (Time) (Transaction)
                     (81H)

(Frame) ::= Frame L1 value
            (80H)

(Time) ::= Time L1 (Amount L1 value) (Period L1 value)
           (81H)    (80H)             (81H)

(Transaction) ::= Transaction L1 value
                     (82H)
```

6 Content architecture class attributes

6.1 *Content architecture class*

The value of the attribute "content architecture class" of a basic component description that conforms to this Recommendation T.101 is a ASN1 object identifier with the value.

{0 1 8 16 3}

6.2 *Content type*

The content architecture class attribute "content type" cannot be used to specify the content architecture class defined in this Recommendation.

7 Content portion attributes

7.1 *Type of coding*

Classification	Defaultable
Applicability	Videotex content architecture class
Structure	ASN1 object identifier
Permissible values	ASN1 object identifier
{0 1 8 16 4}	for "IDS encoding"
{0 1 8 16 5}	for "Data syntax I encoding"
{0 1 8 16 6}	for "Data syntax II encoding"
{0 1 8 16 7}	for "Data syntax III encoding"
Default value:	"IDS encoding"
Definition:	for Videotex interworking, the possible values correspond to the data syntaxes described in Annexes A, B, C, D of this Recommendation.

7.2 *Specific coding attributes*

These attributes provide additional information required for encoding/decoding the content information, as well as other information intrinsic to the content portion and type of coding.

7.2.1 *Subset*

Classification	Defaultable
Applicability	Videotex content architecture class Type of coding "IDS encoding"
Values	Integer [0, 1 to 5, 81 to 92]
Default value	0
Definition	This attribute identifies the subset (rank or profile) used within the IDS. Value 0 is used when no subset is specified.

7.2.2 *Rank*

Classification	Defaultable
Applicability	Videotex content architecture class Type of coding "Data syntax I encoding"
Values	Integer [0, 1 to 5]
Default value	0
Definition	This attribute identifies the rank used within Data syntax I. Value 0 is used when the rank is not specified.

7.2.3 *Profile*

Classification	Defaultable
Applicability	Videotex content architecture class Type of encoding "Data Syntax II encoding"
Values	Integer [0, 81 to 92]
Default value	0
Definition	This attribute identifies the profile used within Data Syntex II. Value 0 is used when the profile is not specified.

8 Formal definition of Videotex dependent data type

8.1 *Introduction*

This section contains formal definition in ASN.1 notation (defined in Recommendation X.208) of the data type corresponding to attributes applicable to Videotex.

This data type is:

- the data type to represent specific coding attributes.

8.2 *Representation of specific coding attributes*

```
Videotex-coding-attributes ::= CHOICE {
        subset  [0] IMPLICIT subset OPTIONAL
        rank    [1] IMPLICIT rank OPTIONAL
        profile [2] IMPLICIT profile OPTIONAL }
Subset a)  ::= INTEGER { undefined      (0)
                         rank 1         (1)
                         rank 2         (2)
                         rank 3         (3)
                         rank 4         (4)
                         rank 5         (5)
                         profile 1      (81)
                         profile 2      (82)
                         profile 3      (83)
                         profile 4      (84)
                         profile X1-1   (85) b)
                         profile X1-2   (86)
                         profile X1-3   (87)
                         profile X1-4   (88)
                         profile X2-1   (89)
                         profile X2-2   (90)
                         profile X2-3   (91)
                         profile X2-4   (92) }
Rank       ::= INTEGER { undefined      (0)
                         rank 1         (1)
                         rank 2         (2)
                         rank 3         (3)
                         rank 4         (4)
                         rank 5         (5) }
Profile    ::= INTEGER { undefined      (0)
                         profile 1      (81)
                         profile 2      (82)
                         profile 3      (83)
                         profile 4      (84)
                         profile X1-1   (85)
                         profile X1-2   (86)
                         profile X1-3   (87)
                         profile X1-4   (88)
                         profile X2-1   (89)
                         profile X2-2   (90)
                         profile X2-3   (91)
                         profile X2-4   (92) }
```

a) Use of subset IDS is for further study.

b) Profil $X_{i\text{-}j}$: geometric profile X_i together with alphamosaic profile j.

8.3 *Summary of ASN.1 object identifiers* (see Table 1/T.101)

TABLE 1/T.101

Videotex document application profile	0 1 8 16 0
DM1 communication application profile	0 1 8 16 1
Videotex operational application profile	0 1 8 16 2
T.101 Content architecture class	0 1 8 16 3
Type of coding	
IDS	0 1 8 16 4
Data syntax I	0 1 8 16 5
Data syntax II	0 1 8 16 6
Data syntax III	0 1 8 16 7
Application context	0 1 8 16 8

ANNEX A

(To Recommendation T.101)

Interworking data syntax (IDS)
described in ASN.1 (Recommendation X.208)

Preamble

For Videotex interchange:

a) If two countries implement the same data syntax, then interworking can use the same data syntax (DS I, or DS II, or DS III).

b) If two countries implement two different data syntaxes, then interworking can use either:

i) the interworking data syntax (IDS) as defined herein, or

ii) any one of the three data syntaxes and convert directly between DS I/DS II/DS III. The data syntaxes may be identified by the ESC 2/5 F mechanism described in § 4.4.2 of the main body of Recommendation T.101.

If the IDS is used, then the Administration in which country the data base is located will be responsible to convert into the IDS and the Administration in which country the user terminal is located will be responsible to convert from the IDS.

If the direct conversion method is used instead of using the IDS, the IDS would serve as a technical guide in designing the conversion process.

The IDS is not intended to be used in terminal to host communications.

A.1 *Videotex page*

A Videotex page in the interworking data syntax (IDS) is a sequence of presentation commands expressed in a manner independent of any of the terminal data syntaxes. This formulation of the presentation information which composes a Videotex page is intended to aid interworking between basically different terminal data syntaxes. It does this by isolating the unique and common elements between each of the data syntaxes. The interworking data syntax is not meant to be used as a terminal data syntax in its own right. One encoding of the interworking data syntax is that defined in Recommendation X.209. Other types of encoding are for further study.

```
Videotex-Page           ::= SEQUENCE OF Presentation-Commands

Presentation-Commands   ::= SEQUENCE { State-Vector, Function-&-Parameters }
```

A.2 *State vector*

A state victor is defined along with each presentation command to establish the relationship of that presentation command to each other presentation command. Although the information explicitly contained in the state vector is also implicitly contained within each presentation command, it would require the conversion apparatus to fully understand each of the three terminal oriented data syntaxes to uncover this information. Therefore a state vector is included with each presentation command in which a global state is affected or in which a boundary value is encountered, so that the conversion process might operate on a general level.

```
State-Vector            ::= CHOICE { [1] Vector-Definition
                                     [2] Reset-State-Vector,
                                     [3] NULL }
```

A.2.1 *Vector definition*

```
Vector-Definition       ::= SEQUENCE { Global-State-Affected-Indicator,
                                       Terminal-Model-Precedence,
                                       Boundary-Condition-Definition }
```

-- Only that information which changes between state vectors need be communicated. If there is no change in a particular component of the state vector, then that component need not be communicated. This means that the state vector is not communicated often and does not introduce significant overhead.

A.2.1.1 *Global state affected indicator*

The global state affected indicator carries information relating to the global states of the presentation data syntax. Global state variables are variables representing those states of the presentation data syntax which are established by presentation commands and which carry on to affect the results of subsequent presentation commands. By declaring the global state variables explicitly, it is not necessary for the conversion process to understand the interrelationship between presentation commands. This means that the conversion process does not have to simulate a terminal of the source data syntax in order to handle the conversion of its elements.

The global state affected indicator does not carry information about the value to which a global state has been set. That information is carried within the 'Function and Parameter' section of the IDS. The indicator merely identifies which states have changed. This is of great importance in situations where it is necessary for the conversion process to sort the presentation commands to account for differences in the terminal model used in the source and destination of the interchange. If the order of presentation commands is altered, the conversion process must establish the appropriate global variables before each command in the altered sequence. By referring to the global state affected indicator, the conversion process can determine which global states must be re-established. For example, if a sorting of presentation commands is necessary to convert from a multi-plane to a single plane terminal mode, and colour control commands have been used, then the global state affected indicator will indicate that the appropriate colour state must be established ahead of each portion of the sorted data.

In some of the terminal data syntaxes attributes have global effects while in other data syntaxes the effects are localized according to the particular display primitive type. For example, in Data Syntax III a colour command remains in effect for all primitives, until the next colour command, whereas in Data Syntax II there are various colour states which apply independently to different primitives, such as LINE colour, FILLED AREA colour, etc. The global state indicator carries a reference to a number of independent 'attribute state vectors' which define the attribute context. For those data syntaxes which make use of only global parameters, only one 'attribute state vector' need be referenced. For other data syntaxes which make use of multiple localized attributes, several 'attribute state vectors' may be referenced.

```
Global-State-Affected-Indicator ::= SEQUENCE {
attribute-state-vector-reference           INTEGER,
attribute-affected-indicators              SEQUENCE OF {
                                                   INTEGER {
                                              current-text-position              (1),
                                              current-foreground-colour          (2),
                                              current-auxiliary-colour           (3),
                                              lining-state                       (4),
                                              flash-blink-state                  (5),
                                              basic-char-size-state              (6),
                                              conceal-state                      (7),
                                              char-invert-box-state              (8),
                                              char-marking-state                 (9),
                                              screen-protection-state            (10),
                                              display-control-state              (11),
                                              device-control-state               (12),
                                              cursor-control-state               (13),
                                              geometric-control-1-state          (14),
                                              geometric-control-2-state          (15),
                                              wait-state                         (16),
                                              general-text-state                 (17),
                                              p-text-state                       (18),
                                              geometric-text-state               (19),
                                              DRCS-definition-state              (20),
                                              macro-definition-state             (21),
                                              texture-pattern-state              (22),
                                              music-part-memory-state            (23),
                                              animation-configuration-state      (24),
                                              workstation-configuration-state    (25)
                                           }}}
```

-- The Global State Affected Indicator consists of a set of indicator flags which identify particular global states or in some cases categories of global states which may be altered by presentation and control commands. Some states, such as all the forms of Flashing and Blink processes, are grouped together for simplicity.

A.2.1.2 *Terminal model*

The terminal model differs significantly between the various terminal data syntaxes. For the presentation of static images this manifests itself in the manner by which presentation commands overlay each other. A picture developed for a multi-plane terminal model can be represented on a terminal using a single-plane terminal model or a multi-plane terminal model with a different order of precedence for the planes, by sorting the presentation commands so that they build up an equivalent picture. The sorting operation is necessary since otherwise the buildup order might conflict with the precedence order in the new environment. The terminal model precedence indicator is simply a numeric indicator of the overlay precedence for presentation commands intended by the source terminal data syntax. The conversion process is independent of the terminal model or a particular data syntax and simply sorts presentation commands based on this indicator. Note that certain commands such as resets which have an effect in more than one plane of a terminal model might have to be repeated in different parts of the presentation sequence after the sort. The terminal model precedence indicator consists of a sequence of numbers to indicate the effect of a command across the terminal model.

Terminal-Model-Precedence ::= INTEGER

-- The terminal model precedence indicator is a number which indicates the order of precedence of the identified presentation information. The number '1' indicates that the identified information is of highest precedence and should be placed in front of any other information currently displayed. The number '2' indicates that the identified information is of the second level of precedence and should be placed behind any level '1' information but ahead of any other level information. For example, Data Syntax II Text and Mosaic data is level '1' information, whereas geometric information may be level '1' or level '2'. For Data Syntax III, all of the information is at level '1' since the precedence order by which it is displayed is determined only by the order in which the information is communicated. For Data Syntax I, the order is not fixed since certain 'planes' of memory may be changed in precedence by the ASSIGN FRAME command.

-- The value '0' has special meaning for the Terminal Model Precedence Indicator. It indicates that the identified information requires special interpretation. Such special information includes partial reset commands which affect more than one layer of the terminal model, as well as commands having a time dependent effect, specifically WAIT, the BEL character, and REVEAL.

A.2.1.3 *Boundary conditions*

Boundary condition variables represent the limits within which the particular presentation command has been defined. Each presentation command takes on its normal interpretation only within a certain range of values. For example, the number of characters which may be displayed on the screen varies between each of the source terminal data syntaxes, and therefore the operation of presenting a single character cannot be considered to be the same in each terminal data syntax. To factor out the commonality, the boundary condition of encountering the edge of the display area is identified separately from the presentation of a character. This aids conversion since it means that the boundary conditions applying to each presentation command are given explicitly. The conversion process is therefore independent of the internal boundary conditions within each of the source terminal data syntaxes.

```
Boundary-Condition-Definition ::= SET { [1] Screen-Dimensions,
                                        [2] Colour-Map-Limit,
                                        [3] Presentation-Sub-Area,
                                        [4] Char-Mode-Constraints,
                                        [5] Coordinate-Limit-Polygon,
                                        [6] Coordinate-Limit-Spline,
                                        [7] Presentation-Resolution,
                                        [8] Macro-Seg-Memory-Limit,
                                        [9] DRCS-Memory-Limit,
                                        [10] Direct-Colours-Limit }
```

A.2.1.3.1 *Screen dimensions*

Screen-Dimensions ::= SEQUENCE { INTEGER, INTEGER }

-- Screen-Dimensions indicates the aspect ratio of the display screen expressed in terms of a fraction of the Y and a fraction of the X unit dimensions, where the INTEGER number represents a binary fraction with an implied binary point before the most significant bit. Note that a dimension of (1,1) implies no geometric constraint. A character mode service could use (1,1) to imply no constraint.

A.2.1.3.2 *Colour map limit*

Colour-Map-Limit ::= INTEGER

-- The colour map limit indicates the maximum number of colours which may be stored in a single colour map, or the combined total for multiple colour maps, and represents the maximum number of colour states which may be encountered in a particular presentation page. In the case where no colour map is used, the integer specifies the number of fixed colours.

A.2.1.3.3 *Presentation sub-area*

Presentation-Sub-Area ::= SEQUENCE { Abs-Coord, Rel-Coord, INTEGER, INTEGER }

-- The two coordinates give the boundary dimensions of a sub-area of the display screen both in terms of the dimensions of the sub-area and the number of characters per row and the number of columns. The absolute coordinate specifies the origin of the sub-area, the relative coordinate the size of the sub-area and the INTEGER coordinates the limit on characters per row and rows respectively.

A.2.1.3.4 *Char mode constraints*

Char-Mode-Constraints ::= SEQUENCE { INTEGER, INTEGER }

-- The two parameters give the limit to the number of characters per row and the number of rows of text which may be presented on the display screen; that is, the the boundaries at which character (or word) wrap and scroll will occur.

A.2.1.3.5 *Coordinate limit polygon*

Coordinate-Limit-Polygon ::= INTEGER

-- The polygon coordinate limit specifies the maximum number of coordinates which may be specified for a filled polygon.

A.2.1.3.6 *Coordinate limit spline*

Coordinate-Limit-Spline ::= INTEGER

-- The spline coordinate limit specifies the maximum number of coordinates which may be specified.

A.2.1.3.7 *Presentation resolution*

```
Presentation-Resolution    ::= SEQUENCE { INTEGER, INTEGER }
```

-- The presentation resolution specifies the nominal resolution of the display screen which was used by the information source.

A.2.1.3.8 *Macro seg memory limit*

```
Macro-Seg-Memory-Limit ::= INTEGER
```

-- The macro memory limit specifies the upper bound on the amount of memory which is available for the storage of Macros or Segments. The INTEGER parameter represents available memory expressed in bytes.

A.2.1.3.9 *DRCS memory limit*

```
DRCS-Memory-Limit      ::= INTEGER
```

-- The DRCS memory limit specifies the upper bound on the amount of memory which is available for the storage of DRCS. The INTEGER parameter represents available memory expressed in bytes.

A.2.1.4 *Data syntax identifier (SID)*

```
SID ::= IMPLICIT INTEGER { data-syntax-  I    (1),
                           data-syntax- II    (2),
                           data-syntax-III    (3) }
```

-- SID is an identifier which is referenced in a number of primitive commands and which identifies the source data syntax of the command.

A.2.2 *Reset state vector*

```
Reset-State-Vector ::= SEQUENCE { SID, Vector-Definition }
```

-- The Reset State Vector command is used to establish the initial state for the Interworking Data Syntax. The default state may be selected from the table corresponding to the source terminal data syntax (or profile) given in Appendix II. Alternate parameters may be specified by use of explicit state vector and function and parameter definitions.

A.2.3 *NULL*

NULL implies that the state vector is unchanged from the previous presentation command.

A.3 *Functions and parameters*

The functions and parameters which make up the presentation commands are grouped into categories which depend upon their commonality between the various terminal data syntaxes. Those functions which are compatible, such as the basic repertoire of alphanumeric characters defined in Recommendation T.51, define separate groups. Those functions which are unique, such as certain specific special characters, also establish separate groups so that they may be converted or otherwise handled in a special manner. Functions such as DRCS and graphics drawing commands, which differ in fundamental ways between the various terminal data syntaxes, are organized so that those underlying capabilities which are common may be exploited in the necessary conversion process.

```
Functions-&-Parameters ::= CHOICE { [0] Alpha-Char-String,
                                    [1] Special-Char-String,
                                    [2] Kana-Char-String,
                                    [3] Kanji-Char-String,
                                    [4] Block-Mosaic-String,
                                    [5] Smooth-Mosaic-String,
                                    [6] Special-Mosaic-String,
                                    [7] Format-Effector-C0-Chars,
                                    [8] Special-Format-C0-Characters,
                                    [9] General-Control-Characters,
                                    [10] Geometric-String,
                                    [11] Animation-Control-String,
                                    [12] Segment-Control-String,
                                    [13] Colour-Control-String,
                                    [14] Text-Control-String,
                                    [15] Photo-Graphic-String-Syntetic-Image,
                                    [16] Photo-Graphic-String-Natural-Image,
                                    [17] MACRO-String,
                                    [18] DRCS-String,
                                    [19] Fill Pattern-Control-String,
                                    [20] Music-String,
                                    [21] Tele-Software-String,
                                    [22] Audio-Data-String,
                                    [23] Greek-Char-String }
```

The first six categories of functions and the last one are various text or mosaic characters. None of the terminal data syntaxes defined in Recommendation T.101 encompasses all of these characters. There are different unique characters in each of the terminal data syntaxes. However, a large portion of the repertoire is common between the different terminal data syntaxes, although the characters may be coded differently. Since coding is irrelevant here, and the use of particular tables could in fact cause serious confusion, characters extracted from the different character repertoires will be distinguished by the identifier name codes for each character as defined in Recommendation T.51. Since all of the terminal-oriented data syntaxes in Recommendation T.101 do not explicitly make use of these name codes in the body of the Recommendation, the entire character repertoire, together with the name codes for each character are included here as an appendix.

A.3.0 *Alpha char string*

Alpha-Char-String ::= GRAPHICSTRING

-- Characters (LA01 to LZ30, ND01 to ND09 and ND10, SC01 to SC05, SP01 to SP22, SA01 to SA07, NS01 to NS03, NF01 to NF21, SM01 to SM44 and SM47 to SM49, and SD11 to SD43) taken from Repertoire 1 which are the characters from the primary and supplementary character sets of Recommendation T.51 together with the SPACE character (SP01) and DELETE character (SM34).

-- The coding of characters within an Alpha Character String will be taken from the IRV primary character code table (ISO Registration Number 2 under ISO 2375) and the secondary code table for use with IRV from ISO 6937/2 (ISO Registration Number 90).

-- *Note* – The coding for the character $ "Dollar Sign" (SC02) will be taken from the supplementary character set.

-- *Note* – The coding for the character # "number sign" (SM01) will be taken from the primary character set.

-- *Note* – The coding for the character "general currency sign" (SC01) will be taken from the primary character set.

A.3.1 *Special char string*

```
Special-Char-String ::= INTEGER { non-spacing-vector-overbar   (1),
                                  non-spacing-slant            (2),
                                  left-vertical-bar-jointive   (3),
                                  right-vertical-bar-jointive  (4) }
```

-- Non-Spacing-Vector-Overbar is a character (SM50) from Repertoire 2.

-- Non-Spacing-Slant is a character (SM51) from Repertoire 2.

-- Left-Vertical-Bar-Jointive is a character (SM45) from Repertoire 2.

-- Right-Vertical-Bar-Jointive is a character (SM46) from Repertoire 2.

A.3.2 *Kana char string*

Kana-Char-String ::= GRAPHICSTRING

-- Characters (JA01 to JA63) taken from Repertoire 3.

-- The coding of characters within a Kana Character String will be taken from the Kana character code table (ISO Registration Number 56 under ISO 2375).

A.3.3 *Kanji char string*

Kanji-Char-String ::= GRAPHICSTRING

-- Characters (JK01 to JK2980, HK01 to HK83, and JS01 to JS366) from Repertoire 4.

-- The coding of characters within a Kanji Character String will be taken from the two byte Kanji character code table (ISO Registration Number 87 under ISO 2375).

-- *Note* – The characters in this two byte code table which overlap other defined videotex character set are not considered to be part of Repertoire 4, and therefore are communicated as characters from Repertoire 1, Repertoire 3 or Repertoire 8 where appropriate. Specifically this involves the Latin alphanumeric characters (LA01 to LZ30), and non-alphabetic characters (ND01 to ND09 and ND00, SC01 to SC05, SP01, SP02, SP04 to SP15, SP17 to SP22, SA01 to SA07, NS02 to NS03, NF01 to NF05, SM01 to SM14, SM19, SM24 to SM34, SM38, SM43, SM44, SM47, SM48, and SD11 to SD43) from Repertoire 1, the Kana characters (JA01 to JA63) from Repertoire 3, the drawing characters (DG01 to DG04, DG13 to DG24, and DG32) from Repertoire 8, which have an alternate coding within the two byte code, but which are included in other Repertoires.

A.3.4 *Block mosaic string*

Block-Mosaic-String ::= GRAPHICSTRING

-- Block Mosaic characters (MG01 to MG63) taken from Repertoire 7.

-- The coding of characters for the Block Mosaic sub-Repertoire is identical between the three terminal data syntaxes defined in CCITT Recommendation T.101. The set is registered with ISO Number 129 under ISO 2375.

A.3.5 *Smooth mosaic string*

```
Smooth-Mosaic-String ::= CHOICE { [1] Sub-Cell-Aligned-Smooth-Mosaics,
                                  [2] General-Smooth-Mosaics }
```

A.3.5.1 *Sub-call aligned smooth mosaics*

Sub-Cell-Aligned-Smooth-Mosaics ::= GRAPHICSTRING

-- Smooth Mosaic characters (SG01 to SG56) taken from Repertoire 8.

-- The coding of characters for the Sub Cell Aligned Smooth Mosaic sub-Repertoire is identical between the two terminal data syntaxes defined in Recommendation T.101 which makes available these characters. These are registered as Numbers 71 and 72 under ISO 2375.

A.3.5.2 *General smooth mosaics*

General-Smooth-Mosaics ::= GRAPHICSTRING

-- Smooth Mosaic characters (MS01 to MS28) taken from Repertoire 8.

-- The coding of characters for the General Smooth Mosaic sub-Repertoire is taken from the terminal data syntax in CCITT Recommendation T.101 which makes available these characters. This code table is registered under ISO 2375 as Registration Number 137.

A.3.6 *Special mosaic string*

```
Special-Mosaic-String ::= CHOICE { [1] Drawing-Characters,
                                   [2] Other-Special-Mosaics }
```

A.3.6.1 *Drawing characters*

Drawing-Characters ::= GRAPHICSTRING

-- Drawing characters (DG01 to DG50) taken from Repertoire 10.

A.3.6.2 *Other special mosaics*

```
Other-Special-Mosaics ::= INTEGER { open-left-half-oval      (1),
                                    open-right-half-oval     (2),
                                    filled-left-half-oval    (3),
                                    filled-right-half-oval   (4),
                                    reverse-left-half-oval   (5),
                                    reverse-right-half-oval  (6) }
```

-- Open-Left-Half-Oval is a Special Mosaic characters (MS13) from Repertoire 11.

-- Open-Right-Half-Oval is a Special Mosaic characters (MS14) from Repertoire 11.

-- Filled-Left-Half-Oval is a Special Mosaic characters (MS30) from Repertoire 11.

-- Filled-Right-Half-Oval is a Special Mosaic characters (MS29) from Repertoire 11.

-- Reverse-Left-Half-Oval is a Special Mosaic characters (MS15) from Repertoire 11.

-- Reverse-Right-Half-Oval is a Special Mosaic characters (MS31) from Repertoire 11.

The function and parameter categories 7 and 8 contain basic control characters which are used to control the state of presentation of alphanumeric text and mosaic characters (including DRCS). These control characters can be broken down into two categories, format effector control characters and special format control characters. The format effector control characters have basically the same meaning in each of the three terminal data syntaxes. The only difference is how the functions invoked by these control characters interact with the terminal model and display environment of the various terminal data syntaxes; for example, they may apply to only one plane of display in a multi-plane terminal model or to all planes of display. The coding of the format effector characters is also compatible between the terminal data syntaxes.

The special format control characters in category 8, in general have a special meaning which is not shared by all of the data syntaxes. These functions must be specially converted during interworking, even between data syntaxes which appear to assign the same meaning to a particular control function. This is because the terminal model and display environment of the various terminal data syntaxes are quite different. The Bell character is included in this category because it requires special handling due to the timing of presentation. If a sorting of presentation commands is required in the interworking conversion process to accommodate differences in a terminal model, such as the handling of data intended for a multi-plane terminal on a single plane terminal, then the time of presentation of the Bell character must be altered.

A.3.7 *Format effector C0-char*

Format-Effector-C0-Char ::= GRAPHICSTRING

-- Format Effector C0 characters (APB, APF, APD, APU, CS, APR, APH) taken from CCITT Recommendation T.101 DS I, II, III; (C0 code table positions 0/8 to 0/13 and 1/14 respectively)

-- APB – Active Position Back – analogous to ISO 646 (FE_0 BS)

-- APF – Active Position Forward – (FE_1 HT)

-- APD – Active Position Down – (FE_2 LF)

-- APU – Active Position Up – (FE_3 VT)

-- CS – Clear Screen – (FE_4 FF)

-- APR – Active Position Return – (FE_5 CR)

-- APH – Active Position Home

A.3.8 *Special format-C0-char*

```
Special-Format-C0-Char ::= CHOICE { [1] Bell-Character,
                                    [2] Position-Set,
                                    [3] Cancel-Macro,
                                    [4] Non-Selective-Reset,
                                    [5] Cancel-Row }
```

A.3.8.1 *Bell character*

```
Bell-Character      ::= GRAPHICSTRING
```

-- Special C0 character (BEL) from Recommendation T.101 DS I, III (C0 set positions 0/7).

-- *Note* – This function provides an audio signal to the user of the terminal device. This function is not available in all of the terminal data syntaxes, and cannot be simulated in a reasonable manner.

A.3.8.2 *Position set*

```
Position-set        ::= SEQUENCE { INTEGER, INTEGER }
```

-- This function provides the equivalent capability of both the Active Position Set command (APS) from Recommendation T.101 DS I, III and the positioning portion of the Active Position Address command (APA) from DS II.

-- The parameters to establish the new screen active position as a count of "current size" character cells from the "home" upper left position.

A.3.8.3 *Cancel macro*

```
Cancel-Macro        ::= GRAPHICSTRING
```

-- Special C0 character [CAN (Macro)] from Recommendation T.101 DS I, III (C0 set positions 1/8).

A.3.8.4 *Non-selective reset*

```
Non-Selective-Reset ::= SEQUENCE { [1] NSR-Code,
                                   [2] Position-Set OPTIONAL }

NSR-Code            ::= GRAPHICSTRING
```

-- Special C0 character (NSR) from Recommendation T.101 DS I, III (C0 set positions 0/15). The positioning parameter sequence is optional.

A.3.8.5 *Cancel row*

```
Cancel-Row          ::= GRAPHICSTRING
```

-- Special C0 characters [CAN (Row)] from Recommendation T.101 DS II (C0 set positions 1/8).

A.3.9 *General control characters*

The function and parameter category 9 contains general control functions which are used to control the general state of presentation. The meaning of these control characters is very dependent upon the terminal model and display environment of the terminal data syntax in which they are used. Transcoding and conversion is required for each of the functions invoked by these control characters. These control characters have been organized into a number of sub categories which correspond to the area of functionality being addressed.

```
General-Control-Characters ::= CHOICE { [1] Other-Format-Effectors,
                                        [2] Lining-Control,
                                        [3] Character-Size-Control,
                                        [4] Flash-Control,
                                        [5] Conceal-Control,
                                        [6] Invert-Control,
                                        [7] Window/Box-Control,
                                        [8] Marking-Control,
                                        [9] Protection-Control,
                                        [10] Display-Control,
                                        [11] Device-Control,
                                        [12] Cursor-Control,
                                        [13] Reset-Control }
```

This subsection addresses the additional format effector characters which must be specially handled in conversion between the various data syntaxes.

A repeat function is available in all of the data syntaxes; however, the side effects of the function differ between the data syntaxes. Terminal data syntax DS I provides a function which allows the immediately preceding G-set character, or pair of characters in the case of a composite coded graphic character and non spacing accent character, to be repeated. Both terminal data syntaxes DS II and DS III restrict the character to a graphic character (i.e. an alphanumeric text character or mosaic character from the repertoire, or a DRCS character). These limitations must be considered in establishing the conversion process. Here the repeat function will be considered to repeat any preceding G-set character and testing must be performed in the interpretation of the IDS to eliminate any erroneous cases.

The functions hold mosaic and release mosaic occur in only one data syntax and require special interpretation since analogous functions do not exist directly in any of the terminal data syntaxes.

A.3.9.1 *Other format effectors*

```
Other-Format-Effectors ::= CHOICE { [1] Repeat-N,
                                    [2] Repeat-EOL,
                                    [3] Hold-Mosaic,
                                    [4] Release-Mosaic }
```

A.3.9.1.1 *Repeat-N*

Repeat-N ::= SEQUENCE { SID, RPT-Par }

-- Special character indicating the REPEAT Function from Recommendation T.101 DS I [C1 set position 5/8, (9/8)], DS II [C0 set position 1/2), DS III [C1 set position 4/6, (8/6)].

RPT-Par ::= INTEGER

-- Count of repetitions.

A.3.9.1.2 *Hold mosaic*

Repeat-EOL ::= SID

-- Special character indicating the REPEAT TO End Of Line Function from Recommendation T.101 DS I [C1 set position 5/8, (9/8)] with parameter 0), DS III [C1 set position 4/7, (8/7)].

A.3.9.1.3 *Hold mosaic*

Hold-Mosaic ::= SID

-- Special character indicating the Hold-Mosaic function (HMS) from Recommendation T.101 DS II [C1 set (serial) position 5/14, (9/14)].

A.3.9.1.4 *Release mosaic*

Release-Mosaic ::= SID

-- Special character indicating the Release-Mosaic (RMS) function from Recommendation T.101 DS II [C1 set (serial) position 5/15, (9/15)].

A.3.9.2 *Lining control*

The lining function permits an underline to be displayed as part of the graphics character shape for alphanumeric characters from Repertoire 1. This underline is considered as a part of the character cell image before any rotation operation is applied. In the special case of the display of mosaic characters, the lining function establishes a "separated mosaic" font. The capability to handle separated mosaics is available in all of the three terminal data syntaxes; however, the level of capability differs. In terminal data syntax DS II, only one size of separation for separated mosaics is directly available. In terminal data syntaxes DS I and DS III the amount of separation is defined by the line width (drawing point size) parameter (logical pel) in the geometric drawing commands. Basic separated mosaics may be converted directly between each of the data syntaxes. Since the variation in separation cannot be achieved directly in one of the terminal data syntax it must be simulated by the use of DRCS. Of course simulation of separated mosaics in this manner would consume limited DRCS resources and therefore must consider the boundary condition specification.

```
Lining-Control ::= INTEGER { start-lining   (1),
                             stop-lining    (2) }
```

-- Start-Lining is a function from Recommendation T.101 DS I and II [C1 set position 5/10, (9/10)] and (UNDERLINE START) from DS III [C1 set position 5/9, (9/9)].

-- Stop-Lining is a function from Recommendation T.101 DS I and II [C1 set position 5/9, (9/9)] and (UNDERLINE STOP) from DS III [C1 set position 5/10, (9/10)].

A.3.9.3 *Character size control*

The various terminal data syntaxes provide the capability to establish a wide range of character sizes for basic alphanumeric text, mosaics and DRCS characters. In addition, terminal data syntax DS II provides the capability to separately define completely variable character sizes for text defined as part of the geometric part of DS II. Since this "geometric text" data is used only for the annotation of geometric pictures in the optional geometric part of DS II, it is not necessary to consider it as part of the translation of basic alphanumeric text. DS III, on the other hand, provides only one form of text. Therefore it is necessary to handle operations such as dynamic text sizes and rotations as part of the conversion between data syntaxes.

Since the capability to scale text, mosaics and DRCS to arbitrary sizes is not available in all data syntaxes, there will be some degradation of the displayed image when converted from one data syntax to another. It is undesirable to lose any textual information in the conversion process, since this textual information might be of principal importance to the understanding of the videotex page. Also it is not desirable to arbitrarily wrap or scroll textual information since this would corrupt mosaic information. In certain situations the conversion process must automatically choose a smaller size of character cell in order to avoid the loss of information. The commands for character size control indicate the size of the character cell intended in the terminal data syntax used to represent the source data. The resultant character cell in the converted form may be smaller depending upon the capabilities of the target terminal data syntax and the boundary condition in effect.

Two separate functions exist to define characters of double height. This is due to a difference in the definition of the relationship of the double height character cell to the location of the baseline in part of one of the source data syntaxes. Since data syntax DS II provides a capability to define double height characters which both extend up a double height above the baseline, and which extend down below the baseline, two functions are provided here. Since the other two terminal data syntaxes provide only a single double height capability, a conversion involving a repositioning of the baseline is required.

```
Character-Size-Control ::= CHOICE { [1] Normal-Size,
                                    [2] Double-Size-Up,
                                    [3] Double-Width,
                                    [4] Double-Height-Up,
                                    [5] Double-Height-Down,
                                    [6] Small-Size,
                                    [7] Medium-Size,
                                    [8] Double-Size-Down }
```

A.3.9.3.1 *Normal size*

```
Normal-Size ::= SID
```

-- A function from Recommendation T.101 DS I [C1 set position 4/10, (8/10)], from DS II [C1 set position 4/12, (8/12)] and (NORMAL TEXT) from DS III [C1 set position 4/12, (8/12)].

-- *Note* – The "Normal-Size" of text is defined by the boundary conditions of each of the terminal data syntaxes and is not the same in any of the terminal data syntaxes. Although the width of the "normal" character cell size is 1/40 of the screen width in DS II and DS III, the screen width is not exactly the same. The "normal" character cell size in DS I is by default 1/31 of the width; however, it may be redefined by the DS I P-TEXT command. Similarly the vertical height of a character cell differs between the various terminal data syntaxes. This command indicates that the source terminal data syntax intended to use "normal" size implicit in that terminal data syntax. This will require conversion to the normal size implicit in that resultant terminal data syntax. The value of "normal size" should be communicated explicitly in the state vector associated with this command.

A.3.9.3.2 *Double size up*

```
Double-Size-Up ::= SID
```

-- A function from Recommendation T.101 DS II [C1 set position 4/15, (8/15)], (DOUBLE SIZE) from DS III [C1 set position 4/15, (8/15)], and (DBS 4/5) from DS I [C1 set position 4/11, (8/11) followed by parameter 4/5].

-- *Note* – The character cell width and height are twice that defined by the control command "Normal-Size".

A.3.9.3.3 *Double width*

Double-Width ::= SID

-- A function from Recommendation T.101 DS II [C1 set position 4/14, (8/14)], and (DBW 4/4) from DS I [C1 set position 4/11, (8/11) followed by parameter 4/4].

-- *Note* – The character cell width is twice that defined by the control command "Normal-Size".

A.3.9.3.4 *Double height up*

Double-Height-Up ::= SID

-- A function from Recommendation T.101 DS II [C1 set position 4/13, (8/13)], (DOUBLE HEIGHT) from DS III [C1 set position 4/13, (8/13)], and (DBH 4/1) from DS I [C1 set position 4/11, (8/11) followed by parameter 4/1].

-- *Note* – The character cell height is twice that defined by the control command "Normal-Size" and extends up two character cell heights above the baseline.

A.3.9.3.5 *Double height down*

Double-Height-Down ::= SID

-- A function from Recommendation T.101 DS II [C1 set position 4/13, (8/13)].

-- *Note* – The character cell height is twice that defined by the control command "Normal-Size" and the double height character extends one cell height above the baseline and one cell height below the baseline.

A.3.9.3.6 *Small size*

Small-Size ::= SID

-- A function from Recommendation T.101 DS I [C1 set position 4/8, (8/8)] and (SMALL TEXT) from DS III [C1 set position 4/10, (8/10)].

-- *Note* – The character cell width and height are half that defined by the control command "Normal-Size".

A.3.9.3.7 *Medium size*

Medium-Size ::= SID

-- A function from Recommendation T.101 DS I [C1 set position 4/9, (8/9)] and (MEDIUM TEXT) from DS III [C1 set position 4/11, (8/11)].

-- *Note* – The character cell size is defined to be an intermediate size. This intermediate size is defined by the boundary conditions of each of the source data syntaxes which use this control function. In data originating from data syntax DS III, medium size is defined to be 1/32 the normalized width of the display area and 3/64 the height of the normalized unit area. In data from data syntax DS I, medium text becomes half the character cell height and the full width defined by the control command "Normal-Size".

A.3.9.3.8 *Double size down*

Double-Size-Down ::= SID

-- A function from Recommendation T.101 DS II [C1 set position 4/15, (8/15)].

A.3.9.4 *Flash control*

The operation of the flash capability is dependent on the terminal model of the particular source data syntax. In a "multi-plane" terminal configuration, character cells may have an implicit foreground and background which alternate during blinking. In a "single-plane" terminal configuration, the flash capability is achieved by use of colour mapping operations. It is possible to convert between these two variants on flashing. In addition to a basic flash capability driven by control characters, each of the terminal data syntaxes also provides the capability to establish complex dynamic important to reference the boundary condition imposed by the number of colours in the colour map and the terminal model plane structure.

```
Flash-Control ::= SEQUENCE { Flash-Rate, Flash-Mode }

Flash-Rate    ::= CHOICE { [1] Flash,
                           [2] Steady,
                           [3] Phase1-Flash,
                           [4] Phase2-Flash,
                           [5] Phase3-Flash,
                           [6] Increment-Flash,
                           [7] Decrement-Flash,
                           [8] Blink-Stop }

Flash-Mode    ::= CHOICE { [1] Normal,
                           [2] Inverted-Flash,
                           [3] Reduced-Intensity-Flash }
```

A.3.9.4.1 *Flash*

Flash ::= SID

-- A function from Recommendation T.101 DS II [C1 set position 4/8, (8/8)], (FLC 4/0) from DS I [C1 set position 5/1, (9/1)] followed by parameter 4/0] and (BLINK START) from DS III [C1 set position 4/14, (8/14)].

-- *Note* – Establish a 50% cycle flash either from the foreground to the background or between two colour map addresses chosen implicitly to produce the equivalent effect of foreground/background flashing. Although the Flash function is similar in the three source data syntaxes, the rate of flashing is not necessarily the same.

A.3.9.4.2 *Steady*

Steady ::= SID

-- A function from Recommendation T.101 DS II [C1 set position 4/9, (8/9)], (FLC 4/15) from DS I [C1 set position 5/1, (9/1)] followed by parameter 4/15].

-- *Note* – Cancel the application of any flashing attribute.

A.3.9.4.3 *Invested flash*

Inverted-Flash ::= SID

-- A function from Recommendation T.101 DS II (C1 set position CSI 3/0 4/1), (FLC 4/7) from DS I [C1 set position 5/1, (9/1)] followed by parameter 4/7].

-- *Note* – Establish an inverted phase 50% cycle flash from the foreground to the background.

A.3.9.4.4 *Reduced intensity flash*

Reduced-Intensity-Flash ::= SID

-- A function from Recommendation T.101 DS II (C1 set position CSI 3/1 4/1), (FLC 4/7) from DS I [C1 set position 5/1, (9/1)] followed by parameter 4/7].

-- *Note* – Establish a reduced intensity flash between colour map addresses.

A.3.9.4.5 *Phase 1-flash*

Phase 1-Flash ::= SID

-- A function from Recommendation T.101 DS II (C1 set position CSI 3/2 4/1), (FLC 4/4) from DS I [C1 set position 5/1, (9/1)] followed by parameter 4/4].

-- *Note* – Establish a 33% cycle flash from the foreground to the background beginning on phase 1.

A.3.9.4.6 *Phase 2-flash*

Phase 2-Flash ::= SID

-- A function from Recommendation T.101 DS II (C1 set position CSI 3/3 4/1), (FLC 4/2) from DS I [C1 set position 5/1, (9/1)] followed by parameter 4/2].

-- *Note* – Establish a 33% cycle flash from the foreground to the background beginning on phase 2.

A.3.9.4.7 *Phase 3-flash*

Phase 3-Flash ::= SID

-- A function from Recommendation T.101 DS II (C1 set position CSI 3/4 4/1), (FLC 4/1) from DS I [C1 set position 5/1, (9/1)] followed by parameter 4/1].

-- *Note* – Establish a 33% cycle flash from the foreground to the background beginning on phase 3.

A.3.9.4.8 *Increment flash*

Increment-Flash ::= SID

-- A function from Recommendation T.101 DS II (C1 set position CSI 3/5 4/1).

-- *Note* – Establish a 33% cycle flash from the foreground to the background incrementing the phase reference.

A.3.9.4.9 *Decrement flash*

Decrement-Flash ::= SID

-- A function from Recommendation T.101 DS II (C1 set position CSI 3/6 4/1).

-- *Note* – Establish a 33% cycle flash from the foreground to the background incrementing the phase reference.

A.3.9.4.10 *Blink stop*

Blink-Stop ::= SID

-- A function from Recommendation T.101 DS III [C1 set position 5/14, (9/14)].

-- *Note* – Stop all blink processes.

A.3.9.5 *Conceal control*

The conceal display function is intended for operation on a terminal model which supports multiple independent planes. Data stored in character cells may be marked as concealed, in which case the background of the character cell will display in the same colour as the background of the cell. A local reveal command would cause the foreground to be displayed in the originally defined colours. A conversion is necessary to handle this function on a single plane terminal. The capability may be simulated either by the use of a key activated macro which contains a definition of the foreground of the concealed character cells or it may be simulated by the colour map. The definition of the key activated macro sequence must be established during a sorting process in the conversion procedure and is limited by the availability of macro memory. Use of the colour map for the simulation of this function consumes colour map resources very quickly. Therefore the handling of the conceal function should be the lowest priority in using colour map resources. The conceal and stop conceal control functions are included here so that they may be handled in the most effective manner by the conversion process.

```
Conceal-Control ::= CHOICE { [1] Conceal-Display,
                             [2] Stop-Conceal-Display }
```

A.3.9.5.1 *Conceal display*

Conceal-Display ::= SID

-- A function from Recommendation T.101 DS II [C1 set position 5/8, (9/8)] and DS I [C1 set position 5/2, (9/2)] followed by parameter 4/0].

-- *Note* – Establish a Conceal state attribute.

A.3.9.5.2 *Stop conceal display*

Stop-Conceal-Display ::= SID

-- A function from Recommendation T.101 DS II (C1 set position CSI 4/2) and DS I [C1 set position 5/2, (9/2) followed by parameter 4/15].

-- *Note* – Stop applying Conceal state attribute.

A.3.9.6 *Invert control*

Invert-Control ::= CHOICE { [1] Invert-Polarity,
[2] Normal-Polarity }

-- *Note* – Invert the application of the foreground and background colour attributes in a multi-plane terminal model environment and invert the overlaying (foreground) and underlaying (background) colours in a single plane terminal environment. These commands have essentially the same effect when generating a presentation in each of the identified terminal model environments; however, there is a great difference in the effect when this command is used to change the attributes of an already displayed graphic character. This must be handled in the conversion by the process which converts the effects of different planes of the terminal model.

A.3.9.6.1 *Invert polarity*

Invert-Polarity ::= SID

-- A function from Recommendation T.101 DS II [C1 set position 5/12, (9/12)] and (REVERSE VIDEO) DS III [C1 set position 4/8, (8/8)].

-- *Note* – Establishes Invert Polarity attribute.

A.3.9.6.2 *Normal polarity*

Normal-Polarity ::= SID

-- A function from Recommendation T.101 DS II [C1 set position 5/13, (9/13)] and (NORMAL VIDEO) DS III [C1 set position 4/9, (8/9)].

-- *Note* – Establishes Normal Polarity attribute.

A.3.9.7 *Window/box control*

The window/box capability establishes a special background colour for a character cell which is transparent to a video image which may underlay the display. This capability is provided directly by two control commands in one of the source terminal data syntaxes. The same capability is provided in a more complex manner in all of the data syntaxes by the establishment of a special transparent colour which may be used together with other presentation commands.

```
Window/Box-Control ::= INTEGER { start-box          (1),
                                 end-box            (2) }
```

-- Start-Box is a function from Recommendation T.101 DS II [C1 set position 4/10, (8/10)].

-- *Note* – Establish the Boxing attribute.

-- End-Box is a function from Recommendation T.101 DS II [C1 set position 4/11, (8/11)].

-- *Note* – Stop applying the Boxing attribute.

A.3.9.8 *Marking control*

The marking control capability marks character cell locations for further action. This function depends upon the availability of a character cell-oriented memory in the terminal model. It cannot be converted to other data syntaxes.

```
Marking-Control    ::= INTEGER { marked-mode-start   (1),
                                 marked-mode-stop    (2) }
```

-- Marked-Mode-Start is a function from Recommendation T.101 DS II (C1 set position CSI 3/0 5/3, CSI 3/1 5/3 or CSI 3/2 5/3).

-- *Note* – Apply the Marking attribute.

-- Marked-Mode-Stop is a function from Recommendation T.101 DS II (C1 set position CSI 3/0 5/4, CSI 3/1 5/4 or CSI 3/2 5/4).

-- *Note* – Stop applying Marking attribute.

A.3.9.9 *Protection control*

The manner in which selective input control is handled in the three source terminal data syntaxes differs greatly. Not only are the procedures different but the input processes are bounded by different boundary conditions. For example, in one case input is associated with the character cell memory of the multi-plane terminal model, whereas in another case, such input data is bounded by a storage limit on the number and cumulative size of such input fields. Since such input processes are fundamentally different, the commands which control them are included here separately. This will permit the conversion process to simulate one set of functions in a different terminal environment.

```
Protection-Control ::= INTEGER { unprotect-field        (1),
                                 protect-field          (2),
                                 protect-mode-start     (3),
                                 protect-mode-cancel    (4),
                                 protect-mode-idle      (5),
                                 unprotect-block        (6),
                                 protect-block          (7) }
```

-- Unprotect-Field is a function from Recommendation T.101 DS III [C1 set position 5/15, (9/15)].

-- *Note* – Unprotect a given area of the display screen, defined by the FIELD geometric command, to allow the input of characters into the unprotected field buffer when the cursor is in the unprotected area.

-- Protect-Field is a function from Recommendation T.101 DS III [C1 set position 5/0, (9/0)].

-- *Note* – Protect a given area of the display screen to prevent the input of characters into the unprotected field buffer when the cursor is in the unprotected area. The entire screen area is protected by default.

-- Protect-Mode-Start is a function from Recommendation T.101 DS II (C1 set position CSI 3/0 5/0, CSI 3/1 5/0 or CSI 3/2 5/0).

-- *Note* – Apply the protected attribute to character cell positions preventing overwriting.

-- Protect-Mode-Cancel is a function from Recommendation T.101 DS II (C1 set position CSI 3/0 5/1, CSI 3/1 5/1 or CSI 3/2 5/1).

-- *Note* – Cancel the protected attribute to character cell positions allowing overwriting.

-- Protect-Mode-Idle is a function from Recommendation T.101 DS II (C1 set position CSI 3/2 5/2).

-- *Note* – Stop the application of the protect mode attribute.

-- Unprotect-Block is a function from Recommendation T.101 DS I [C1 set position 5/14, (9/14)].

-- *Note* – Remove protection of character cell positions against alteration.

-- Protect-Block is a function from Recommendation T.101 DS I [C1 set position 5/15, (9/15)].

-- *Note* – Protect character cell positions against alteration.

A.3.9.10 *Display control*

The display control subcategory of commands contains functions which affect the manner in which the display device presents information. This includes configuration of the display memory available in a particular terminal model, includes whether the contents of that display memory is to be scrolled, and includes the overwriting of information in the display memory.

```
Display-Control ::= CHOICE { [1] Plane-Configuration-Control,
                             [2] Scroll-Control,
                             [3] Overwrite-Mode }
```

The terminal models used in each of the three terminal data syntaxes differ significantly from one another. In two of the cases the terminal model structures are fixed. In the case of data syntax DS I the terminal model structure can be altered dynamically. The amount of display memory assigned to each display plane and the presentation (overlay) order of the planes may be altered. These functions are highly dependent upon the display hardware used to realize the particular display model which underlies data syntax DS I and the dynamic effects which may be generated using these functions cannot be converted to either of the other data syntaxes. However these comamnds must be interpreted by the conversion process in order to establish the criteria for sorting other display information to achieve the mapping from the data syntax DS I multi plane terminal model to the different data syntax DS II multi-plane terminal model or to the data syntax DS III single plane terminal model.

A.3.9.10.1 *Plane configuration control*

```
Plane-Configuration-Control ::= CHOICE { [1] Frame-Area,
                                         [2] Set-Frame,
                                         [3] Assign-Frame,
                                         [4] Header-Area,
                                         [5] Body-Area }
```

A.3.9.10.1.1 *Frame area*

Frame-Area ::= SEQUENCE { Area-Origin, Area-Dimensions }

-- The Frame Area Function is from Recommendation T.101 DS I [Display Control command set position 2/5, (10/5)].

-- *Note* – The Display Control Command G Set has final character 3/8 within DS I.

Area-Origin ::= SEQUENCE { REAL, REAL }

-- Specification of the origin of the Frame Area.

Area-Dimensions ::= SEQUENCE { REAL, REAL }

-- Specification of the dimensions of the Frame Area.

-- Coordinates are specified as normalized fractions of the unit screen area represented in a signed integer field with an implied binary point in the most significant place.

A.3.9.10.1.2 *Set frame*

```
Set-Frame ::= SEQUENCE OF { Set-Frame-Index,
                            Set-Frame-Memory-Assignment }
```

-- The Frame Area Function is from Recommendation T.101 DS I [Display Control command set position 2/6, (10/6)].

Set-Frame-Index ::= INTEGER

-- Frame Area Index.

Set-Frame-Dimensions ::= INTEGER

-- Number of bits of raster memory allocated to the frame.

A.3.9.10.1.3 *Assign frame*

Assign-Frame ::= INTEGER

-- A function from Recommendation T.101 DS I [Display Control command set position 2/7, (10/7)].

A.3.9.10.1.4 *Header area*

Some of the terminal oriented Videotex data syntaxes provide a capability to present information in a special message area as well as in the main display area. This message area would contain service oriented messages. The content of these messages would doubtless change in international interworking between Videotex systems. Data syntax DS I provides special commands which control this message header. In DS I the raster and header raster commands control the display of presentation information in the main display area or the header message area. The raster commands also establish the initial colour values in data syntax DS I. These commands are included here so that header information can be identified and properly converted.

Header-Area ::= SEQUENCE { Raster-Colour-Value }

-- A function from Recommendation T.101 DS I [Display Control command set position 3/9, (11/9)].

-- *Note* – The Display Control Command G Set has final character 3/8 within DS I.

Raster-Colour-Values ::= SEQUENCE { INTEGER, INTEGER, INTEGER }

-- Specification of the initial raster header colour for Green, Red, and Blue respectively.

-- Colour values are specified as normalized fractions of the unit range of colours represented in a signed integer field with an implied binary point in the most significant place.

A.3.9.10.1.5 *Body area*

Body-Area ::= SEQUENCE { Body-Opcode, Raster-Colour-Values }

-- A function from Recommendation T.101 DS I [Display Control command set position 3/8, (11/8)].

A.3.9.10.2 *Scroll control*

Scrolling may occur on a whole screen basis or on a partial screen basis. There is a major difference between scrolling on a multi-plane terminal model and on a single plane terminal model. As well the assignment of functions to the various planes in a multi-plane terminal model also makes a tremendous difference to the result of scrolling. In some cases the underlying graphics information moves with the scrolling characters and in other cases it remains in place. In DS I the multi-plane motion capability permits dynamic motions and plane assignments which greatly affect how scrolling operates. In general it is not possible to convert all dynamic operations such as scrolling between terminal data syntaxes; however, the results of scrolling affects the final presentation. The conversion process must buffer data and post-process it so that the final image is correct. Since the scroll operations in each of the three terminal data syntaxes are fundamentally different, they are all included here so that the conversion process can handle them.

```
Scroll-Control ::= CHOICE { scroll-on                       [1] NULL,
                            scroll-off                      [2] NULL,
                            scroll-up                       [3] NULL,
                            scroll-down                     [4] NULL,
                            activate-implicit-scrolling     [5] NULL,
                            deactivate-implicit-scrolling   [6] NULL,
                            create-scroll-area              [7] Create-Scroll-Area,
                            delete-scroll-area              [8] Delete-Scroll-Area,
                            scroll-display-mode-on          [9] NULL,
                            scroll-display-mode-off         [10] NULL }
```

-- Scroll-On is a function from Recommendation T.101 DS III [C1 set position 5/7, (9/7)].

-- *Note* – Enable single plane scroll within an active display Field.

-- Scroll-Off is a function from Recommendation T.101 DS III [C1 set position 5/8, (9/8)].

-- *Note* – Disable single plane scroll.

-- Scroll-Up is a function from Recommendation T.101 DS II (C1 set position CSI 3/0 6/0).

-- *Note* – Cause the scrolling area to scroll up.

-- Scroll-Down is a function from Recommendation T.101 DS II (C1 set position CSI 3/1 6/0).

-- *Note* – Cause the scrolling area to scroll down.

-- Activate-Implicit-Scrolling is a function from Recommendation T.101 DS II (C1 set position CSI 3/2 6/0).

-- *Note* – Cause the scrolling area to scroll implicitly on encountering scroll area boundary.

-- Deactivate-Implicit-Scrolling is a function from Recommendation T.101 DS II (C1 set position CSI 3/3 6/0).

-- *Note* – Cause the scrolling area not to scroll implicitly.

-- Scroll-Display-Mode-On is a function from Recommendation T.101 DS I (Display Control Command G Set position 2/4 with parameter b6 = 1).

-- *Note* – The Display Control Command G Set has final character 3/8 within DS I.

-- *Note* – Establish the Scroll attribute of the Display Mode.

-- Scroll-Display-Mode-Off is a function from Recommendation T.101 DS I (Display Control Command G Set position 2/4 with parameter b6 = 0).

-- *Note* – Disable the Scroll attribute of the Display Mode.

A.3.9.10.2.1 *Create scroll area*

Create-Scroll-Area ::= SEQUENCE { Upper-Par, Lower-Par }

-- A function from Recommendation T.101 DS II (5/5).

-- *Note* – Create a scrolling area.

Upper-Par ::= SEQUENCE { INTEGER, INTEGER, INTEGER }

-- Parameters <URH> <URT> <URU> defining the upper boundary row of the scrolling area.

Lower-Par ::= SEQUENCE { INTEGER, INTEGER, INTEGER }

-- Parameters <LRH> <LRT> <LRU> defining the lower boundary row of the scrolling area.

A.3.9.10.2.2 *Delete scroll area*

Delete-Scroll-Area ::= SEQUENCE { Upper-Par, Lower-Par }

-- A function from Recommendation T.101 DS II (5/6).

-- *Note* – Delete a scrolling area.

A.3.9.10.3 *Overwrite mode*

In conjunction with control over the terminal model memory configuration, one of the terminal data syntaxes provides a unique capability of controlling how data builds up in a particular display plane. Data syntax DS I allows the overwriting of memory to be dependent upon the current contents of memory. The new data may either replace the old contents of memory, or perform a logical "OR", logical "AND", or logical "XOR" (eXclusive OR) with the old contents of the memory before replacing it. This function is extremely difficult to simulate in either of the other two data syntaxes in the general case since it requires operations at the bit level within a particular terminal model dependent memory. It is included here so that the conversion process can perform the best simulation possible.

Overwrite-Mode ::= SEQUENCE { Overwrite-Par }

-- A function from Recommendation T.101 DS I (Display Control Command G Set position 2/4).

-- *Note* – The Display Control Command G Set has final character 3/8 within DS I.

```
Overwrite-Par    ::= INTEGER { replace        (1),
                               or             (2),
                               and            (3),
                               xor            (4) }
```

A.3.9.11 *Device control*

Except for the display device on or off commands, the device-control commands control other than presentation display functions and are outside the scope of the interworking data syntax.

```
Device-Control ::= INTEGER { display-device-on     (1),
                             display-device-off    (2) }
```

-- Display-Device-On is a function from Recommendation T.101 DS II (Control Sequence ESC 3/12).

-- Display-Device-Off is a function from Recommendation T.101 DS II (Control Sequence ESC 3/13).

A.3.9.12 *Cursor control*

The display cursor is controlled explicitly in each of the terminal data syntaxes as well as implicitly in one. In addition the explicit cursor control commands do not have the same coding in any of the terminal data syntaxes. In the implicit case of cursor control, the display cursor is controlled by the unprotected field protect mode control in terminal data syntax DS III. Conversion is required between each of these control functions.

```
Cursor-Control ::= CHOICE { Cursor-On          (1),
                            Cursor-Flash       (2),
                            Cursor-Off         (3) }
```

A.3.9.12.1 *Cursor-on*

Cursor-On ::= SID

-- A function from Recommendation T.101 DS II (C0 set position 1/1), DS I [C1 set position 4/14, (8/14)] and (CURSOR STEADY) from DS III [C1 set position 5/12, (9/12)].

A.3.9.12.2 *Cursor flash*

Cursor-Flash ::= SID

-- A function from Recommendation T.101 DS III [C1 set position 5/11, (9/11)].

A.3.9.12.3 *Cursor-off*

Cursor-Off ::= SID

-- A function from Recommendation T.101 DS II (C0 set position 1/14), DS I [C1 set position 4/15, (8/15)] and (CURSOR OFF) from DS III [C1 set position 5/13, (9/13)].

A.3.9.13 *Reset control*

Each of the source Videotex data syntaxes provides the capability to reset the states of the display environment supporting that particular data syntax to a predefined set of values. The parameters which may be altered by the various reset functions in the different source data syntaxes are quite different. Some of the reset functions provide the capability to reset particular parameters selectively while others reset a data syntax dependent predefined list of parameters. The interworking data syntax must support reset functions in two different ways. Firstly an indication of the particular reset command must be communicated as a syntactic element within the IDS. The various reset functions are included here so that the presentation affect of the reset function may be effected in the conversion. Secondly a reset function greatly effects the global presentation states. These states are kept track of in the conversion process so that the conversion process need not understand the interrelationship between presentation commands. This means that the conversion process does not have to simulate a terminal of the source data syntax in order to handle the conversion of its elements. Therefore along with an IDS reset control command it is necessary to include a special form of the state vector which re-establishes the global variables.

```
Reset-Control ::= CHOICE { [1] Reset-Type-I,
                           [2] Reset-Type-II,
                           [3] Reset-Type-III }
```

A.3.9.13.1 *Reset type-I*

Reset-Type-I ::= SEQUENCE { P-Reset-Par OPTIONAL }

-- A function from Recommendation T.101 DS I [Display Control Command G Set position 2/1, (10/1)].

-- *Note* – The Display Control Command G Set has final character 3/8 within DS I.

A.3.9.13.1.1 *P-reset par*

```
P-Reset-Par    ::= SEQUENCE { macro-reset BOOLEAN,
                              blink-reset BOOLEAN,
                              lut-reset BOOLEAN,
                              screen-reset BOOLEAN }
```

-- Selectively reset the identified parameters.

-- *Note* – Data Syntax DS I also includes the NSR reset function which is identified separately above.

A.3.9.13.2 *Reset type-II*

```
Reset-Type-II ::= SEQUENCE { US-Reset-Operation,
                             US-Reset-Parameter }
```

-- A function from Recommendation T.101 DS II (C0 set position 1/15) followed by fixed character 2/15.

A.3.9.13.2.1 *US-reset operation*

```
US-Reset-Operation ::= CHOICE { us-reset-mosaic-1          [1] NULL,
                                us-reset-mosaic-2          [2] NULL,
                                us-reset-mosaic-1-limited  [3] NULL,
                                us-reset-mosaic-2-limited  [4] NULL,
                                us-reset-service-break     [5] US-Reset-Service-Break,
                                us-reset-to-previous-state [6] NULL }
```

-- US-Reset-Mosaic-1 is represented by US Reset Identifier Character (4/1), and resets to defaults and invokes serial C1 set.

-- US-Reset-Mosaic-2 is represented by US Reset Identifier Character (4/2), and resets to defaults and invokes parallel C1 set.

-- US-Reset-Mosaic-1-Limited is represented by US Reset Identifier Character (4/3), and resets to limited defaults and invokes parallel C1 set.

-- US-Reset-Mosaic-2-Limited is represented by US Reset Identifier Character (4/4), and resets to limited defaults and invokes parallel C1 set.

-- US-Reset-to-Previous-State is represented by US Reset Identifier Character (4/15), and resets to previous state after a reset to service break.

A.3.9.13.2.2 *US-reset service break*

```
US-Reset-Service-Break ::= SEQUENCE { INTEGER { break-to-row-serial (1),
                                      break-to-row-parallel (2) }, row-designator }
```

-- Break-to-Row-Serial is represented by US Reset Identifier Character (4/0), and service breaks to row serial C1 set.

-- Break-to-Row-Parallel is represented by US Reset Identifier Character (4/5), and service breaks to row parallel C1 set.

-- Row-Designator is represented by US Reset Row Designator Parameter Character, where the designated row is coded from columns 4 to 7 of the code table. The row number is indicated by the binary value of the 6 least significants bits.

A.3.9.13.3 *Reset type-III*

```
Reset-Type-III ::= SEQUENCE { [1] Reset-Par1 OPTIONAL,
                              [2] Reset-Par2 OPTIONAL }
```

-- A function from Recommendation T.101 DS III [PD1 G Set position 2/0, (10/0)].

```
Reset-Par1     ::= SEQUENCE { INTEGER {
                                   colour-mode-1                                       (1),
                                   colour-mode-2                                       (2),
                                   colour-mode-3                                       (3), }
                              INTEGER {
                                   display-to-nominal-black                            (1),
                                   display-to-current-colour                           (2),
                                   border-to-nominal-black                             (3),
                                   border-to-current-colour                            (4),
                                   display-and-border-to-current-colour                (5),
                                   display-to-current-colour-and-border-to-nominal-black (6),
                                   display-and-border-to-nominal-black                 (7) },
                              domain BOOLEAN }

Reset-Par2     ::= SEQUENCE {
                              drcs-reset BOOLEAN,
                              macro-pdi-reset BOOLEAN,
                              texture-reset BOOLEAN,
                              unprotected-field-reset BOOLEAN,
                              blink-pdi-reset BOOLEAN,
                              text-pdi-reset BOOLEAN }
```

-- Selectively reset the identified parameters.

-- *Note* – Data Syntax DS III also includes the NSR reset function which is identified separately above.

A.3.10 *Geometric string*

All of the source terminal data syntaxes provide a geometric capability, however the capabilities available in each of the geometric systems is quite different. The interworking data syntax groups the common geometric commands together. Recommendation F.300 has identified the various categories into which geometric functions may be organized. These categories are used below. The IDS uses normalized coordinates for all geometric commands. Also the IDS uses relative coordinate specifications for all lists of coordinates, except for the set-position and marker-point commands which are absolute. However, in certain cases there may be a choice of either absolute or relative coordinates. All other forms of coordinates used within any of the terminal oriented data syntaxes, such as the general use of absolute or incremental, will be converted to the forms indicated above.

```
Geometric-String           ::= CHOICE { [1] Geometric-Drawing-Command,
                                        [2] Geometric-Control-Command }
```

A.3.10.1 *Geometric drawing command*

```
Geometric-Drawing-Command ::= CHOICE { [1] Marker-Point,
                                       [2] Line,
                                       [3] Arc-Circle,
                                       [4] Rectangle,
                                       [5] Polygon,
                                       [6] Spline,
                                       [7] Pixel-Array }
```

Some of the source terminal data syntaxes provide a method of optionally carrying on from one primitive to another in a relative manner. This provides a level of efficiency in certain situations, however the multiplicity of equivalent formats would make the interworking data syntax more complex. Therefore the interworking data syntax requires the specification of the initial position of a drawing command as part of the string of parameters for each command. In some situations in converting from data syntaxes which permit the relative association of commands it will be necessary for the conversion process to calculate the current position in effect at the beginning of a command and include that data as part of the parameter string. There is no direct equivalent in the IDS of the data syntaxes I and III set position command. This information is carried as the initial parameter of each of the other drawing commands.

A.3.10.1.1 *Marker point*

The various terminal data syntaxes differ in their capability to present a marker shape at a point. Data syntaxes DS I and DS III provide only the capability to draw a dot, whereas data syntax DS II also provides the capability to draw a marker shape at a specific point. The conversion process can easily simulate the marker point functionality in converting to data syntax DS I or DS III by the use of more than one presentation function, possibly included in a MACRO command for efficiency. The dot-point or shape-point command is identified by the context tag in the CHOICE statement. The shape of the shape point (marker) is defined by a geometric control command.

```
Marker-Point  ::= CHOICE { [1] Dot-Point,
                           [2] Shape-Point }

Dot-Point     ::= SEQUENCE OF { Abs-Coord }
```

-- This command carries the functionality of the Data Syntax I, and III SET POINT command and of the Data Syntax II POLYMARKER command, with the marker the shape of a dot.

```
Shape-Point   ::= SEQUENCE OF { Abs-Coord }
```

-- This command carries the functionality of the Data Syntax II POLYMARKER command with a general marker shape.

A.3.10.1.2 *Line*

All of the terminal data syntaxes provide the capability to draw a single or a series of lines. Minor differences exist with respect to the manner in which boundary conditions are handled, however in general a direct conversion is possible.

```
Line ::= SEQUENCE OF { Abs-Coord, SEQUENCE OF { Rel-Coord }
```

-- This command carries the functionality of the Data Syntax I, and III LINE command and of the Data Syntax II POLYLINE command.

A.3.10.1.3 *Arc-circle*

The capability to draw an arc or a circle differs somewhat between the various data syntaxes. In each of the various data syntaxes the circle/arc function has been optimized to such an extent that it provides an efficient manner of communicating arc or circle information in the context of that data syntax. The interworking data syntax is less concerned about efficiency that it is about carrying sufficient information to permit conversion to take place. Therefore alternate ways of carrying the same parameters will not be addressed by the IDS, however the IDS will include all the functions available in the circle-arc capability in the various data syntaxes.

```
Arc-Circle ::= CHOICE { [1] Circle,
                        [2] Arc-3-Point,
                        [3] Arc-3-Point-Chord,
                        [4] Arc-3-Point-Pie,
                        [5] Ellipse,
                        [6] Elliptic-Arc,
                        [7] Elliptic-Arc-Chord,
                        [8] Elliptic-Arc-Pie,
                        [9] Arc-Centre-Cord,
                        [10] Arc-Centre-Pie }
```

A.3.10.1.3.1 *Circle*

```
Circle          ::= SEQUENCE { Abs-Coord, Coord }
```

-- This command carries the functionality of the Data Syntax I, and III ARC command (circle form) and of the Data Syntax II GDP (circle) command.

-- The absolute coordinate defines the initial position of the circle. The other coordinate defines the diameter of the circle by specifying a point on the opposite side.

A.3.10.1.3.2 *Arc-3 point*

```
Arc-3-Point     ::= SEQUENCE { Abs-Coord, Coord, Coord }
```

-- This command carries the functionality of the Data Syntax I, and III ARC command (outline form) and of the Data Syntax II GDP (circular arc 3 point) command.

-- The absolute coordinate defines the initial position of the arc. The two other coordinate parameters define a point on the arc and the final position of the arc respectively.

A.3.10.1.3.3 *Arc-3 point chord*

```
Arc-3-Point-Chord ::= SEQUENCE { Abs-Coord, Coord, Coord }
```

-- This command carries the functionality of the Data Syntax I, and III ARC command (chord fill form) and of the Data Syntax II GDP (circular arc 3 point chord) command.

-- The absolute coordinate defines the initial position of the arc. The two other coordinate parameters define a point on the arc and the final position of the arc respectively. A Chord is drawn from the initial to the final position of the arc.

A.3.10.1.3.4 *Arc-3 point pie*

Arc-3-Point-Pie ::= SEQUENCE { Abs-Coord, Coord, Coord }

-- This command carries the functionality of the Data Syntax II GDP (circular arc 3 point pie) command.

-- The absolute coordinate defines the initial position of the arc. The two other coordinate parameters define a point on the arc and the final position of the arc respectively. Two lines are drawn from the initial to the geometric centre of the arc and then to the final position of the arc to form a pie shape. Although a pie filled arc is not directly available in data syntax DS I or DS III the conversion process can simulate the function by the use of an arc and two lines.

A.3.10.1.3.5 *Ellipse*

Ellipse ::= SEQUENCE { Abs-Coord, Coord, Coord, Coord }

-- This command carries the functionality of the Data Syntax II GDP (ellipse) command.

-- The absolute coordinate defines the initial position of the ellipse. A second coordinate parameters defines a point on the opposite side of the arc which establishes the major axis diameter. The third and fourth parameters define the minor axis diameter. Although an ellipse or elliptic arc are not directly available in data syntax DS I or DS III the conversion process can simulate the function in a piecewise manner or by fitting a spline curve.

A.3.10.1.3.6 *Elliptic arc*

Elliptic-Arc ::= SEQUENCE { Abs-Coord, Coord, Coord, Coord }

-- This command carries the functionality of the Data Syntax II GDP (elliptic arc) command.

-- The absolute coordinate defines the initial position of the arc. A second coordinate parameters defines a point on the opposite side of the arc which establishes the major axis diameter. A third parameter defines the minor axis diameter. A fourth parameter defines the final position of the arc.

A.3.10.1.3.7 *Elliptic arc chord*

Elliptic-Arc-Chord ::= SEQUENCE { Abs-Coord, Coord, Coord, Coord }

-- This command carries the functionality of the Data Syntax II GDP (elliptic arc chord) command.

-- The absolute coordinate defines the initial position of the arc. A second coordinate parameters defines a point on the opposite side of the arc which establishes the major axis diameter. A third parameter defines the minor axis diameter. A fourth parameter defines the final position of the arc. A Chord is drawn from the initial to the final position of the arc.

A.3.10.1.3.8 *Elliptic arc pie*

Elliptic-Arc-Pie ::= SEQUENCE { Abs-Coord, Coord, Coord, Coord }

-- This command carries the functionality of the Data Syntax II GDP (elliptic arc pie) command.

-- The absolute coordinate defines the initial position of the arc. A second coordinate parameters defines a point on the opposite side of the arc which establishes the major axis diameter. A third parameter defines the minor axis diameter. A fourth parameter defines the final position of the arc. Two lines are drawn from the initial to the geometric centre of the arc and then to the final position of the arc to form a pie shape.

A.3.10.1.3.9 *Arc centre cord*

Arc-Centre-Chord ::= SEQUENCE { Abs-Coord, Coord, Coord }

-- This command carries the functionality of the Data Syntax II GDP (arc-centre-chord) command.

-- The absolute coordinate defines the initial position of the arc. The other coordinate parameters define the start and end points of the arc.

A.3.10.1.3.10 *Arc centre pie*

Arc-Centre-Pie ::= SEQUENCE { Abs-Coord, Coord, Coord }

-- This command carries the functionality of the Data Syntax II GDP (arc-centre-pie) command.

-- The absolute coordinate defines the centre of the arc. The other coordinate parameters define the start and end points of the arc.

A.3.10.1.4 *Rectangle*

Rectangle ::= SEQUENCE { Abs-Coord, Rel-Coord }

-- This command carries the functionality of the Data Syntax I and III RECTANGLE command and the Data Syntax II GDP (rectangle) command.

-- The absolute coordinate defines the initial position of the rectangle. A relative coordinate parameters defines a point on the diagonally opposite side of the rectangle which establishes the size of the rectangle.

A.3.10.1.5 *Polygon*

Polygon ::= SEQUENCE { Abs-Coord, SEQUENCE OF { Rel-Coord } }

-- This command carries the functionality of the Data Syntax I and III POLYGON (filled) command and the Data Syntax II FILL AREA command. Data Syntax I and III also provide a POLYGON (outline) command which can be carried through the IDS by a LINE command with a repetition of the initial points as the final point.

-- The absolute coordinate defines the initial position of the polygon. The sequence of relative coordinate define the vertices of the polygon. A polygon is always closed and the final position is the same as the initial position.

A.3.10.1.6 *Spline*

Spline ::= SEQUENCE { Abs-Coord, SEQUENCE OF { Rel-Coord } }

-- This command carries the functionality of the Data Syntax I and III ARC (spline) command and the Data Syntax II GDP (spline) command.

-- The absolute coordinate defines the initial position of the poly curve. The sequence of relative coordinates (greater than 3) define the curve.

-- *Note* – The various terminal data syntaxes do not use exactly the same definition of the type and/or parameters for the spline generating function, however all of the source terminal data syntaxes tend to use a spline function of some type. Although potentially this could cause significant diferences in the resultant picture after conversion, it is still the closest result than can be generated in a reasonable manner.

A.3.10.1.7 *Pixel array*

```
Pixel-Array      ::= SEQUENCE {
                              first-point Abs-Coord,
                              second-point Abs-Coord,
                              third-point Rel-Coord,
```

-- these 3 points define the pixel area which in general could be a parallelogram. The first two points are the end points of a diagonal.

```
                              cells-first-direction INTEGER,
                              cells-second-direction INTEGER,
```

-- These values divide the pixel area in a grid with equal dimensions, to represent the intended (logical) resolution. The first direction is considered from the first to the third point. The second direction is from the first point to the unspecified point. These values can easily be derived, e.g. from the logical pel in case of INCREMENTAL POINT.

```
                              Pixel-Array-Data }
```

```
Pixel-Array-Data ::= CHOICE { [1] IMPLICIT SEQUENCE OF Basic-Colour-Selection,
                              [2] IMPLICIT SEQUENCE OF Direct-Colour-Selection,
                              [3] IMPLICIT SEQUENCE OF Indexed-Colour-Selection }
```

-- The colour list is defined according to the 'Colour-Control-String'. Auxiliary colour selection is not meaningful for this definition. The first colour is mapped to the cell associated with the first point. The colour elements are mapped within rows running from the first to the third point, and with rows incrementing in order from the third to the second point.

The various source terminal Videotex data syntaxes contain commands to efficiently code line and polygon data in an incremental fashion to achieve greater efficiency. The incremental capability differs greatly between the different data syntaxes, and no intermediate format could be developed which would be suitable in all the different environments. Since efficiency is of secondary importance, incremental lines and polygons should be communicated in terms of the general line and polygon functions above.

A.3.10.2 *Geometric control commands*

A large number of control commands are available in each of the terminal data syntaxes to control the geometric drawing functions. Although many of the geometric control commands defined in each of the data syntaxes may appear to be the same, they differ in side effect. For this reason all of the geometric control commands which appear in the various data syntaxes are included here. Only where the control commands are identical, such as a number of the geometric control commands in data syntaxes DS I and DS III, is a common control command definition used below.

```
Geometric-Control-Command    ::= CHOICE { [1] Geo-Control-Command-1,
                                          [2] Geo-Control-Command-2 }
```

-- Two types of geometric control commands are included in the IDS in order to accommodate the two different approaches taken in Data Syntax II and in Data Syntax I, III. These commands are grouped separately since they would never be received in combination.

A.3.10.2.1 *Geo control command-1*

```
Geometric-Control-Command-1 ::= CHOICE { [1] Numeric-Precision,
                                         [2] Drawing-Point-Size,
                                         [3] Line-Style,
                                         [4] Highlight,
                                         [5] Fill,
                                         [6] Field,
                                         [7] Blink-Process,
                                         [8] Wait }
```

-- Geometric control commands analogous to those in Data Syntax I and III.

A.3.10.2.1.1 *Numeric precision*

```
Numeric-Precision    ::= SEQUENCE { REAL, REAL }
```

-- This command carries the functionality of the Data Syntax I, and III DOMAIN geometric control command.

-- Define the nominal numeric precision in use by the source data syntax. Since the ASN.1 encoding rules permit any precision of data to be communicated, this control command does not affect the precision of data communicated. It is used to inform the conversion process of the nominal precision being used by the source data syntax. The first parameter carries the precision, expressed as a number of significant bits, for single-value operands. Similarly the second parameter carries the number of significant bits for multi-valued (2d and 3d) operands.

A.3.10.2.1.2 *Drawing point size*

```
Drawing-Point-Size  ::= Rel-Coord
```

-- This command carries the functionality of the Data Syntax I, and III DOMAIN (logical pel size) geometric control command.

-- This geometric control function establishes the size of the logical drawing point (LOGICAL PEL) as a fraction of the unit screen dimensions. The special case of zero is interpreted as being the smallest size possible on a given presentation device.

A.3.10.2.1.3 *Line style*

```
Line-Style          ::= INTEGER { solid          (1),
                                  dotted         (2),
                                  dashed         (3),
                                  dot-dashed     (4) }
```

-- Establish the style for presenting lines from a fixed set of line styles.

-- This command carries the functionality of the Data Syntax I, and III TEXTURE (line texture) geometric control command.

A.3.10.2.1.4 *Highlight*

Highlight ::= BOOLEAN

-- Establish whether filled areas are drawn in highlight mode, in which the perimeter is drawn in BLACK or a contrasting colour to the fill.

-- This command carries the functionality of the Data Syntax I, and III TEXTURE (highlight) geometric control command.

A.3.10.2.1.5 *Fill*

Fill ::= BOOLEAN

-- Establish whether polygons, closed arcs, ellipses or rectangles are to be filled. For efficiency this control is codes as part of the opcode identifying the drawing primitive in some of the source terminal data syntaxes. This function has been separated here in order to ease conversion between data syntaxes.

-- This command carries the functionality of the Data Syntax I, and III TEXTURE (fill texture pattern) geometric control command.

A.3.10.2.1.6 *Field*

Field ::= Rel-Coord

-- Define the dimensions of the active area on the dispaly screen. The field command establishes boundaries which "contain" text; that is boundaries for scroll areas, and to which the format effector characters operate. The initial position is defined by the current geometric drawing position. The relative coordinate parameters define a point on the diagonally opposite side of the field which establishes the size of the field rectangular area.

-- This command carries the functionality of the Data Syntax I, and III FIELD geometric control command.

A.3.10.2.1.7 *Blink process*

```
Blink-Process ::= SEQUENCE { [1] INTEGER,
                             [2] INTEGER OPTIONAL,
                             [3] INTEGER OPTIONAL,
                             [4] INTEGER OPTIONAL }
```

-- Establish a Blink process in which the colour map is dynamically altered for a specified interval and phase. The first integer represents the colour map address of the Blink To colour, then the On interval, the Off interval and the Phase Delay in 1/10 of a second respectively. The capability to handle Blink processes is very terminal model dependent. In general Blink processes can be used to simulate any other blink capability available in any data syntax, within the limits of the available memory assigned for such operations, as specified in boundary value conditions. However Blink processes cannot easily be simulated in display environments which do not present sufficient capabilities.

-- This command carries the functionality of the Data Syntax I, and III BLINK geometric control command.

A.3.10.2.1.8 *Wait*

```
Wait          ::= INTEGER
```

-- Establish a time delay in processing presentation data for the time specified in units of 1/10 of a second. Although the Wait command is very simple, it provides very great problems in conversion. This is because the wait command is a dynamic control command. Presentation dynamics cannot be guaranteed in conversion because the order of presentation commands may have to be altered to accommodate for differences in the terminal model between two data syntaxes. Conversion of the wait command should only be attempted when the source and target presentation processes are in synchronization, i.e. when no sorting of presentation commands is necessary in the conversion, or at the end of a unit (page) of data.

A.3.10.2.2 *Geo control command-2*

```
Geo-Control-Command-2 ::= CHOICE { [1] Display-Element-Attributes,
                                   [2] Control-Element-Attributes }
```

-- Geometric control commands analogous to those in Data Syntax II.

-- Display Element Attributes pertain to the output display primitives. Some of this primitives may be similar to those in Geo-Control-Command-1 section, however the side effects are different for these commands.

-- Control Element Attributes establish the display transformation, clipping and work station control functions which are unique to the display environment associated with Data Syntax II.

-- The use of bundle facilities is for further study.

A.3.10.2.2.1 *Display element attributes*

```
Display-Element-Attributes ::= CHOICE {
                                   [1] IMPLICIT Line-Attributes,
                                   [2] IMPLICIT Marker-Attributes,
                                   [3] IMPLICIT Fill-Area-Attributes }

Line-Attributes            ::= SET {
                                   [1] IMPLICIT Line-Type OPTIONAL,
                                   [2] IMPLICIT Line-Width-Scale-Factor OPTIONAL,
                                   [3] IMPLICIT Polyline-Colour-Index OPTIONAL }

Line-Type                  ::= INTEGER {
                                   solid                              (0),
                                   dashed                             (1),
                                   dotted                             (2),
                                   dashed-dotted                      (3),
                                   implementation dependent           (4) }

Line-Width-Scale-Factor    ::= REAL

Polyline-Colour-Index      ::= Colour-Index

Marker-Attributes          ::= SET {
                                   [1] IMPLICIT Marker-Type OPTIONAL,
                                   [2] IMPLICIT Marker-Size-Scale-Factor OPTIONAL,
                                   [3] IMPLICIT Polymarker-Colour-Index OPTIONAL }

Marker-Type                ::= INTEGER {
                                   dot                                (0),
                                   plus                               (1),
                                   asterisk                           (2),
                                   circle                             (3),
                                   diagonal-cross                     (4) }

Marker-Size-Scale-Factor   ::= REAL

Polymarker-Colour-Index    ::= Colour-Index

Fill-Area-Attributes       ::= SET {
                                   [1] IMPLICIT Fill-Area-Interior-Style OPTIONAL,
                                   [2] IMPLICIT Fill-Area-Colour-Style OPTIONAL,
                                   [3] IMPLICIT Fill-Area-Style-Index OPTIONAL,
                                   [4] IMPLICIT Pattern-Reference-Point OPTIONAL,
                                   [5] IMPLICIT Pattern-Vectors OPTIONAL }

Fill-Area-Interior-Style   ::= INTEGER {
                                   hollow                             (0),
                                   solid                              (1),
                                   pattern                            (2),
                                   hatch                              (3) }

Fill-Area-Colour-Index     ::= Colour-Index

Fill-Area-Style-Index      ::= INTEGER {
-- For interior style pattern the fill area style index selects a pattern defined by "Fill pattern control string".
-- For interior style hatch the following styles are selected:
                                   vertical-lines                     (0),
                                   horizontal-lines                   (1),
                                   slope-45-degree-lines              (2),
                                   slope-45-degree-lines              (3),
                                   crossed-lines-vertical-and-horizontal-lines (4),
                                   crossed-lines-45-and-45-degrees    (5) }

Pattern-Reference-Point    ::= Abs-Coord

Pattern-Vectors            ::= SEQUENCE { Abs-Coord, Abs-Coord }
-- The origin of the NDC space an the first point defines the pattern height vector. The origin of the NDC
   space an the second point defines the pattern widht vector.

Colour-Index               ::= CHOICE {
                                   [1] IMPLICIT Basic-Colour-Selection,
                                   [2] IMPLICIT Indexed-Colour-Selection }
```

A.3.10.2.2.2 *Control element attributes*

```
Control-Element-Attributes  ::= CHOICE {
                                        [1] WS-Management-Primitives,
                                        [2] Transformation-Primitives }

WS-Management-Primitives ::= CHOICE {
open-workstation                        [1] IMPLICIT INTEGER,
                                         --WS Identifier
close-workstation                       [2] IMPLICIT INTEGER,
                                         --WS Identifier
activate-workstation                    [3] IMPLICIT INTEGER,
                                         --WS Identifier
deactivate-workstation                  [4] IMPLICIT INTEGER,
                                         --WS Identifier
clear-workstation                       [5] IMPLICIT INTEGER,
                                         --WS Identifier
set-defaults                            [6] IMPLICIT NULL,
update-workstation                      [7] IMPLICIT Update-WS,
deferral-state                          [8] IMPLICIT Deferral-State }

Update-WS                   ::= SEQUENCE {
workstation-identifier                      INTEGER,
regeneration-flag                           INTEGER { perform      (0),
                                                      postpone     (1) } }

Deferral-State              ::= SEQUENCE {
workstation-identifier                      INTEGER,
deferral-mode                               INTEGER { asap         (0),
                                                      bnil         (1),
                                                      bnig         (2),
                                                      asti         (3) }
implicit-regeneration                       INTEGER { suppressed   (0),
                                                      allowed      (1) } }

Transformation-Primitives   ::= SET {
                                        [1] IMPLICIT WS-Window OPTIONAL,
                                        [2] IMPLICIT WS-Viewport OPTIONAL,
                                        [3] IMPLICIT Clipping-Rectangle OPTIONAL }

WS-Window                   ::= SEQUENCE {
workstation-Identifier                  INTEGER,
first-point                             Abs-Coord,
second-point                            Abs-Coord }

WS-Viewport                 ::= SEQUENCE {
workstation-identifier                  INTEGER,
xmin                                    REAL,
xmax                                    REAL,
ymin                                    REAL,
ymax                                    REAL }

Clipping-Rectangle          ::= SEQUENCE {
first-point                             Abs-Coord,
second-point                            Abs-Coord }
```

A.3.10.3 *Geometric coordinates*

Coordinate data for geometric operations is stored in terms of normalized display coordinates in all three source data syntaxes. However, the exact details of the number format differ significantly between the approach taken in data syntaxes DS I and DS III and that taken in data syntax DS II. Since the purpose of the IDS is for interworking, differences with respect to the number format should be avoided. Therefore, within the IDS a simple numbering scheme based on the ASN.1 signed REAL data type is used. ASN.1 REAL numbers are self-delimiting and of arbitrary length, so there is no difficulty with precision and no need to assign special bit fields to determine the length of the number. A coordinate can therefore be represented as a pair of numbers. The mapping of a real data field to a numeric data field in any of the data syntaxes is dependent upon that particular data syntax. For the case of data syntaxes DS I and DS III the normalized unit display area is mapped to the fractional part (i.e. mantissa part) of the real number field. For DS II both the mantissa and exponent of the real number are used.

Since three dimensional coordinate specifications are optionally available in all of the data syntaxes, an integer triplet is optionally provided below. Because three dimensional operation is optional, the projection to two dimensions must be defined so that three dimensional information may be viewed in a two dimensional environment through interworking. A plane projection which assumes Z = 0 is used.

```
Coord       ::= IMPLICIT CHOICE { Abs-Coord, Rel-Coord }

Abs-Coord   ::= CHOICE { [1] X-Y,
                         [2] X-Y-Z }

X-Y         ::= SEQUENCE { REAL, REAL }

--  Absolute X, Y Coordinates

X-Y-Z       ::= SEQUENCE { REAL, REAL, REAL }

--  Absolute X, Y, Z Coordinates

Rel-Coord   ::= CHOICE {
                         [3] DX-DY,
                         [4] DX-DY-DZ }

DX-DY       ::= SEQUENCE { REAL, REAL }

--  Relative DX, DY Coordinates

DX-DY-DZ    ::= SEQUENCE { REAL, REAL, REAL }

--  Relative DX, DY, DZ Coordinates
```

A.3.11 *Animation control string*

The capability to achieve dynamic or animated effects on the presentation device is highly dependent on the terminal model and display environment. Several of the terminal data syntaxes provide some specialized capabilities to achieve dynamic effects. For example, data syntaxes DS I and DS III include three phase flash (blink) capability and data syntax DS III includes a colour map phased blink function. The dynamic effects generated by these special functions will not in general be preserved in conversion. This is especially true since the order of the display of presentation entities may be altered by the conversion process to account for differences in the terminal model. Except for flash (blink) it is necessary for the conversion process to take into account dynamic effects even though it cannot convert them faithfully, since they may significantly alter the final resultant picture.

A sophisticated terminal model dependent animation capability is available in data syntax DS I. This capability makes use of a multi-plane terminal model in which the order and relative position of the various planes may be altered. The effects which may be generated by this capability are unique to the environment in which they were defined. The animation control commands from data syntax DS I must, however, be included in the interworking data syntax, since they affect the final result of the display. The conversion process must generate the correct final resultant picture.

```
Animation-Control-String  ::= CHOICE { mvi-start          [1] NULL,
                                       mvi-stop           [2] NULL,
                                       mvi-repeat-start   [3] MVI-Repeat-Start,
                                       mvi-repeat-end     [4] NULL,
                                       mvi-move           [5] MVI-Move }
```

-- MVI-Start is a function from Recommendation T.101 DS I (MVI Code set position 2/0)

-- MVI-Stop is a function from Recommendation T.101 DS I (MVI Code set position 2/1)

A.3.11.1 *MVI-repeat start*

MVI-Repeat-Start ::= SEQUENCE { GRAPHICSTRING, INTEGER }

-- General character (REPEAT START) from Recommendation T.101 DS I (MVI Code set position 3/12 or 11/12), followed by a count of the number of repetitions

-- MVI-Repeat-End is a function from Recommendation T.101 DS I (MVI Code set position 3/13 or 11/13)

A.3.11.2 *MVI-move*

MVI-Move ::= SEQUENCE { Move-Origin, Move-Termination, Move-Time }

-- MVI-Move is a function from Recommendation T.101 DS I (MVI Code set position 3/10 or 11/10)

Move-Origin ::= Abs-Coord

-- X, Y Parameters codes as packed binary fractions

Move-Termination ::= OCTETSTRING

-- X, Y Parameters codes as packed binary fractions

Move-Time ::= INTEGER

-- Numeric count of the time period for the move operation in units of 1/10 of a second

A.3.12 *Segment control string*

Data syntax II provides an optional segment storage and editing capability. One or two storage memories for display segments are retained. Editing commands may produce dynamic effects by altering the stored display segment and causing the redisplay of the picture. A display segment may contain any geometric string data as well as the special segment attributes as described below.

Segment control is similar to animation control in that it provides functions which control special display environment dependent capabilities. Since analogous functions are not available in either data syntax I or III, these functions must be handled in the conversion process. For the conversion of information from data syntax II into data syntax I or III only one "Workstation" (or display screen) is used.

```
Segment-Control-String    ::= CHOICE { [1] Work-Station-Dependent,
                                       [2] Work-Station-Independent }

Work-Station-Dependent    ::= CHOICE { [1] W-Create,
                                       [2] W-Close,
                                       [3] W-Rename,
                                       [4] W-Delete-1,
                                       [5] W-Delete-2,
                                       [6] W-Redraw,
                                       [7] W-Set-Highlight,
                                       [8] W-Set-Visibility,
                                       [9] W-Set-Seg-Transparent,
                                       [10] W-Set-Priority }
```

A.3.12.1.1 *W-create*

```
W-Create      ::= INTEGER

--    Open the identified segment.
```

A.3.12.1.2 *W-close*

```
W-Close       ::= INTEGER

--    Close the identified segment.
```

A.3.12.1.3 *W-rename*

```
W-Rename      ::= SEQUENCE {
                          old-segment-number  [1] INTEGER,
                          new-segment-number  [2] INTEGER }

--    Rename old segment number to new segment number.
```

A.3.12.1.4 *W-delete-1*

```
W-Delete-1    ::= SEQUENCE {
                           work-station-id  [1] INTEGER,
                           segment-number [2] INTEGER }
```

-- Delete identified segment from workstation.

A.3.12.1.5 *W-delete-2*

```
W-Delete-2    ::= INTEGER
```

-- Delete the identified segment from all workstations.

A.3.12.1.6 *W-redraw*

```
W-Redraw      ::= INTEGER
```

-- Redraw the identified workstation.

A.3.12.1.7 *W-set highlight*

```
W-Set-Highlight:= SEQUENCE {
                           highlight-segment-number  [1] INTEGER,
                           highlight-attribute       [2] INTEGER }
```

-- Set highlight attribute of identified segment.

A.3.12.1.8 *W-set visibility*

```
W-Set-Visibility::= SEQUENCE {
                           visibility-segment-number  [1] INTEGER,
                           visibility-attribute       [2] INTEGER }
```

-- Set visibility attribute of identified segment.

A.3.12.1.9 *W-set segment transparent*

```
W-Set-Seg-Transparent     ::= SEQUENCE { transparent-segment-number [1] INTEGER,
                                         transform-matrix           [2] MAT }
```

-- Set transformation matrix attributes for the identified segment.

```
MAT                       ::= SET { matrix-element-11               [11] REAL,
                                    matrix-element-12               [12] REAL,
                                    matrix-element-13               [13] REAL,
                                    matrix-element-21               [21] REAL,
                                    matrix-element-22               [22] REAL,
                                    matrix-element-23               [23] REAL }
```

-- Transform Matrix Definition.

A.3.12.1.10 *W-set priority*

```
W-Set-Priority            ::= SEQUENCE { priority-segment-number    [1] INTEGER,
                                         priority-value             [2] REAL }
```

-- Set segment priority attribute for the identified segment. This is analogous to display order priority.

A.3.12.2 *Work station independent*

```
Work-Station-Independent ::= CHOICE { [1] W-Associated,
                                      [2] W-Copy,
                                      [3] W-Insert }
```

A.3.12.2.1 *W-associated*

```
W-Associated              ::= SEQUENCE { associated-w-station-id    [1] INTEGER,
                                         associated-segment-number  [2] INTEGER }
```

-- Associate the identified segment with the identified work station.

A.3.12.2.2 *W-copy*

```
W-Copy                ::= SEQUENCE { copy-w-station-id        [1] INTEGER,
                                     copy-segment-number      [2] INTEGER }
```

-- Copy the primitives of the identified work station.

A.3.12.2.3 *W-insert*

```
W-Insert              ::= SEQUENCE { insert-segment-number        [1] INTEGER,
                                     insert-transform-matrix-ref  [2] MAT }
```

-- Transform and display segment.

A.3.13 *Colour control string*

All of the source terminal data syntaxes provide the capability to define colour and have available, at least optionally, a colour map capability. However the colour model which is used by each of the source terminal data syntaxes differs significantly. In order to provide a neutral basis for colour, the colour model developed for ISO 8613 Text and Offices Systems – Office Document Architecture is used here.

The basic colour model used in ISO 8613 is a colour cube in terms of the three Red, Green, Blue basic vectors. A colour in the colour model is represented by a three tuple of RGB components. Logically these colours are normalized from 0 (minimum) to 1 (maximum). Therefore 'Black' is the three tuple <0,0,0> and "White" is the three tuple <1,1,1>. Since all of the videotex terminal data syntaxes support colour models which are different from this, the mapping of the specific colour models to the basic RGB colour cube must be understood by the conversion process.

Two colour indexing modes are available: Direct and Indexed. In direct colour selection, the colour is defined by providing a three tuple of discrete values for the RGB components. In the indexed colour selection mode, the colour is defined by an index into a single colour table of discrete colour values. The number of colours which may be defined in the colour table is terminal model dependent. The limit assumed in the definition of a particular set of data is specified in the Boundary Value Definition section. If a receiving system cannot image the range of colour values specified by a direct colour value or the colour value indexed by a colour index, then a 'closest match' is assumed according to the criteria stated in ISO 8613. A variant of the indexed colour mode called 'auxiliary colour mode' is used to define a colour for the background of a text or mosaic character cell.

```
Colour-Control-String  ::= CHOICE { [1] Basic-Colour-Selection,
                                    [2] Direct-Colour-Selection,
                                    [3] Indexed-Colour-Selection,
                                    [4] Auxiliary-Colour-Selection,
                                    [5] Colour-Index-Setup }
```

A.3.13.1 *Basic colour selection*

```
Basic-Colour-Selection ::= INTEGER { black              (0),
                                     red                (1),
                                     green              (2),
                                     yellow             (3),
                                     blue               (4),
                                     magenta            (5),
                                     cyan               (6),
                                     white              (7),
                                     auxiliary-black    (8),
                                     auxiliary-red      (9),
                                     auxiliary-green    (10),
                                     auxiliary-yellow   (11),
                                     auxiliary-blue     (12),
                                     auxiliary-magenta  (13),
                                     auxiliary-cyan     (14),
                                     auxiliary-white    (15),
                                     auxiliary-foreground (16) }
```

-- Several of the terminal data syntaxes provide a simplified way to access the basic primary colours by the use of C1 set codes. The various C1 control sets differ in a fundamental manner with respect to how the colour command interacts with other attributes. In order to avoid the difficulty in interworking, only the basic colour commands themselves are identified here. The range of a colour specification is terminal model dependent. This tendency has been avoided by defining all the colour selection commands in terms of the colour model specified in ISO 8613. Colour range dependencies (serial row attributes or parallel cell attributes) should be expressed in terms of the abstract colour by the conversion process. That is, all the rules inherent in the serial attribute method of specifying basic colours, or in the parallel attribute method, should be resolved by the conversion process which creates the IDS colour commands.

-- The auxiliary colour commands specify the background colour for text and mosaics. The command 'auxiliary foreground' specifies that the background colour should be set to the current foreground colour.

A.3.13.2 *Direct colour selection*

Direct-Colour-Selection ::= SEQUENCE { REAL, REAL, REAL }

-- Direct colour selection permits colours to be specified in terms of the Red Green Blue components of the colour model. The ASN.1 REAL data type is use since this form of number is self-delimiting and of arbitrary length. The real number parameters are relative to the maximum colour value for each component. The parameters are Red, Green and Blue respectively.

A.3.13.3 *Indexed colour selection*

Indexed-Colour-Selection ::= INTEGER

-- Indexed colour selection permits colours to be specified as an index into an indirect colour map, which contains actual Red, Green and Blue colour specifications for each colour. The length of the colour map and the number of colour maps available is terminal model dependent. The INTEGER parameter is interpreted with respect to the current size of the colour map specified in Boundary Value Definition. In order to accommodate the rules for accommodating differences in the colour value extent, as specified in ISO 8613, the INTEGER parameter is interpreted as a normalized fraction of the specified map lenght. Some terminal data syntaxes provide the capability of multiple colour maps. Multiple maps are logically equivalent to one large map encompassing a number of submaps. In the IDS, the use of several colour maps is handled by arbitrarily partitioning the single IDS colour map.

A.3.13.4 *Auxiliary colour selection*

Auxiliary-Colour-Selection ::= INTEGER

-- Auxiliary colour selection permits colours to be specified for the background of Text or Mosaics character cells. The operation of this command is similar to the Indexed Colour selection above, except that the current backgroun colour is established.

A.3.13.5 *Colour index setup*

Colour-Index-Setup ::= SEQUENCE { INTEGER, REAL, REAL, REAL }

-- The Colour Index setup command defines the contents of the colour map. The first parameter takes indexes into the colour map in a similar manner to the Indexed Colour Selection command. The remaining three parameters define the Red, Green and Blue colour values in a manner similar to the Direct Colour Specification command.

A.3.14 *Text colour string*

The manner in which text is presented and the specialized attributes and constraints which pertain to the presentation of texts differs between each of the terminal data syntaxes.

```
Text-Control-String    ::= CHOICE { [1] General-Text-Control,
                                    [2] Word-Wrap-Control }
```

A.3.14.1 *General text control*

```
General-Text-Control  ::= SEQUENCE { [1] General-Text-Control-Code,
                                     [2] G-Text-Par1 OPTIONAL,
                                     [3] G-Text-Par2 OPTIONAL,
                                     [4] Rel-Coord OPTIONAL,
                                     [5] Abs-Coord OPTIONAL }

General-Text-Control-Code ::= GRAPHICSTRING
```

-- General control function from Recommendation T.101 DS III [PDI G Set position 2/2, (10/2)].

-- *Note* – PDI G Set has final character 5/7 within DS III.

```
G-Text-Par1          ::= SET { [1] Char-Rotation OPTIONAL,
                               [2] IMPLICIT Char-Path OPTIONAL,
                               [3] Char-Spacing OPTIONAL,
                               [4] IMPLICIT Text-Precision OPTIONAL,
                               [5] IMPLICIT Char-Expansion-Factor OPTIONAL,
                               [6] Text-Colour-Index OPTIONAL,
                               [7] IMPLICIT Text-Alignment OPTIONAL }

Char-Rotation        ::= CHOICE { predefined [1] IMPLICIT INTEGER {
                                                 char-rotation-0        (0),
                                                 char-rotation-90       (1),
                                                 char-rotation-180      (2),
                                                 char-rotation-270      (3) }

                          continuous [2] IMPLICIT SEQUENCE {
                                                 height-vector Abs-Coord,
                                                 width-vector Abs-Coord } }
```

```
Char-Path              ::= INTEGER { char-path-right   (0),
                                     char-path-left    (1),
                                     char-path-up      (2),
                                     char-path-down    (3) }

Char-Spacing           ::= CHOICE { predefined [1] IMPLICIT INTEGER {
                                                  char-spacing-1        (0),
                                                  char-spacing-5/4      (1),
                                                  char-spacing-3/2      (2) }
                                    continuous [2] IMPLICIT REAL }

Text-Precision         ::= INTEGER { string          (0),
                                     char            (1),
                                     stroke          (2) }

Char-Expansion-Factor ::= REAL

Text-Control-Index     ::= CHOICE { [1] IMPLICIT Basic-Colour-Selection,
                                    [2] IMPLICIT Indexed-Colour-Selection }

Text-Alignment         ::= SEQUENCE { Horizontal-Alignment,
                                      Vertical-Alignment }

Horizontal-Alignment ::= INTEGER { normal         (0),
                                   left           (1),
                                   centre         (2),
                                   right          (3) }

Vertical-Alignment     ::= INTEGER { normal         (0),
                                     top            (1),
                                     cap            (2),
                                     half           (3),
                                     base           (4),
                                     bottom         (5) }

G-Text-Par2            ::= SEQUENCE { INTEGER { cursor-style-underscore                         (0),
                                                cursor-style-block                              (1),
                                                cursor-style-cross-hair                         (2),
                                                cursor-style-custom                             (3) }
                                      INTEGER { cursor-&-geometric-drawing-position-together    (0),
                                                cursor-leads-geometric-drawing-position         (1),
                                                geometric-drawing-position-leads-cursor         (1),
                                                cursor-&-geometric-drawing-position-separate    (3) }
                                      INTEGER { char-interrow-spacing-1                         (0),
                                                char-interrow-spacing-5/4                       (1),
                                                char-interrow-spacing-3/2                       (2),
                                                char-interrow-spacing-2                         (3) }
                                      Char-Block-Dimension }

-- The relative coordinates define the size of the character field.

Char-Block-Dimensions ::= Rel-Coord
```

A.3.14.2 *Word wrap control*

The capability to wrap the presentation of characters on a word boundary rather than on a character boundary is available in one of the terminal data syntaxes. This capability cannot be directly converted to other data syntaxes; however, the effect can be achieved in the converter by issuing appropriate format effector characters.

```
Word-Wrap-Control   ::= INTEGER { Word-Wrap-On  (1),
                                  Word-Wrap-Off (2) }
```

-- Word-Wrap-On is a function from Recommendation T.101 DS III [C1 set position 5/5, (9/5)].

-- Word-Wrap-Off is a function from Recommendation T.101 DS III [C1 set position 5/6, (9/6)].

A.3.15 *Photographic string synthetic image*

All of the terminal data syntaxes provide a method of handling an array of pixels. Some of the data syntaxes also provide general photographic capabilities which provide more efficient methods of encoding the same type of data. This means that interworking between all of the terminal data syntaxes is possible for photographic data, even though it may be inefficient in some cases.

Two classes of photographic images are identified below. They are the Synthetic and the Natural Image forms of photographic. The synthetic form of photographic corresponds to the photographic capabilities of data Syntax I. Natural Image photographic coding is for further study.

```
Photo-Graphic-String-Synthetic-Image ::= CHOICE { [1] Line-Dot-Pattern,
                                                  [2] Line-Dot-Pattern-Comp,
                                                  [3] Field-Dot-Pattern,
                                                  [4] Colouring-Block,
                                                  [5] Colouring-Block-Comp,
                                                  [6] Field-Colouring-Block,
                                                  [7] Field-Colouring-Block-Comp,
                                                  [8] Free-Format-Colouring-Block }
```

-- Photographic Synthetic Image functions correspond to Recommendation T.101 Data Syntax I. These functions are suitable for displaying synthetic images such as Kanji characters, graphics, etc.

A.3.15.1 *Line dot pattern*

```
Line-Dot-Pattern      ::= SEQUENCE { y-origin-point-coordinate-Idp Abs-Coord,
                                     dot-pattern-data-Idp BITSTRING }
```

-- Line-Dot-Pattern functions indicates a selection of two colours which are defined by a colouring Block, Field Colouring Block, etc. This function gives dot pattern data of one or several lines at a time.

A.3.15.2 *Line dot pattern comp*

```
Line-Dot-Pattern-Comp ::= SEQUENCE { y-origin-point-coordinate-Idpc Abs-Coord,
                                     mh-run-length coded-data BITSTRING }
```

-- The Line-Dot-Pattern-Comp function is equivalent to the Line-Dot-Pattern function except that the dot patterns are encoded in a compressed manner using the M.H. Run Length Code.

A.3.15.3 *Field dot pattern*

```
Field-Dot-Pattern     ::= SEQUENCE { xy-origin-point-coordinate Abs-Coord,
                                     dx-dy-field-dimensions Rel-Coord,
                                     dot-pattern-data-fdp BITSTRING }
```

-- The Field-Dot-Pattern function is equivalent to the Line-Dot-Pattern function except that this function defines the dot pattern in a rectangular area.

A.3.15.4 *Colouring block*

```
Colouring-Block       ::= SEQUENCE { fg-bg-da-existence-indicator INTEGER,
                                     y-origin-point-coordinate-cb Abs-Coord,
                              SEQUENCE OF { SEQUENCE {
                                     fg-colour BITSTRING,
                                     bg-colour BITSTRING,
                                     display-attributes-cb BITSTRING } } }
```

-- The Colouring-Block function defines a photographic image by specifying the foregroung colour (FG), backgroung colour (BG), and display attributes of certain blocks ahead of which is indicated by the parameter y-origin-point-coordinate.

A.3.15.5 *Colouring block comp*

```
Colouring-Block-Comp ::= SEQUENCE { colouring-block-comp-function-id INTEGER,
                                    fg-bg-da-existence-indicator-cbc INTEGER,
                                    y-origin-point-coordinate-cbs Abs-Coord,
                             SEQUENCE OF { SEQUENCE {
                                    fg-comp-colour BITSTRING,
                                    fg-runlength BITSTRING,
                                    bg-comp-colour BITSTRING,
                                    bg-runlength BITSTRING,
                                    display-attributes-cbc BITSTRING,
                                    da-runlength BITSTRING } } }
```

-- The Colouring-Block-Comp function is equivalent to that of the Colouring-Block function except that colour and display attributes data are encoded by compressed manner as run-length code.

A.3.15.6 *Field colouring block*

```
Field-Colouring-Block  ::= SEQUENCE { field-colouring-block-function-id INTEGER,
                                      fg-bg-da-existence-indicator-fcb INTEGER,
                                      xy-origin-point-coordinate-fcb Abs-Coord,
                                      dx-dy-field-dimensions-fcb Rel-Coord,
                                SEQUENCE OF { SEQUENCE {
                                      fg-colour-fbc BITSTRING,
                                      bg-colour-fbc BITSTRING,
                                      display-attributes-fcb BITSTRING } } }
```

-- The Field-Colouring-Block function defines a photographic image by specifying the foregroung colour (FG), backgroung colour (BG), and display attributes of certain blocks which are contained in the the field allocated by xy-origin-point-coordinate and the dx-dy-field-dimensions.

A.3.15.7 *Field colouring block comp*

```
Field-Colouring-Block-Comp ::= SEQUENCE { field-colouring-block-comp-function-id INTEGER,
                                          fg-bg-da-existence-indicator-fcbc INTEGER,
                                          xy-origin-point-coordinate-fcbc Abs-Coord,
                                          dx-dy-field-dimensions-fcbc Rel-Coord,
                                    SEQUENCE OF { SEQUENCE {
                                          fg-colour-fcbc BITSTRING,
                                          fg-runlength-fcbc BITSTRING,
                                          bg-comp-colour-fcbc BITSTRING,
                                          bg-runlength-fcbc BITSTRING,
                                          display-attributes-fbc BITSTRING,
                                          da-runlength-fcbc BITSTRING } } }
```

-- The Field-Colouring-Block-Comp function is equivalent to that of the Field-Colouring-Block function except that colour and display attributes data are encoded by compressed manner as run-length code.

A.3.15.8 *Free format colouring block*

```
Free-Format-Colouring-Block ::= SEQUENCE { fg-bg-da-existence-indicator-ffcb INTEGER,
                                           fg-bg-da-code-length INTEGER,
                                           run-length-code-length-ffcb INTEGER,
                                           xy-origin-point-coordinate-ffcb Abs-Coord,
                                           dx-dy-field-dimensions-ffcb Rel-Coord,
                                     SEQUENCE OF { SEQUENCE {
                                           fg-colour-ffcb BITSTRING,
                                           runlength-ffcb BITSTRING,
                                           bg-comp-colour-ffcb BITSTRING,
                                           bg-runlength-ffcb BITSTRING,
                                           display-attributes-ffcb BITSTRING,
                                           da-runlength-ffcb BITSTRING } } }
```

-- The Free-Format-Colouring-Block function is equivalent to that of the Field-Colouring-Block-Comp function except that the code length of the Foreground, Background, Display Attributes and Run Lenght can be arbitrarily set.

A.3.16 *Photo graphic string natural image*

```
Photo-Graphic-String-Natural-Image ::= CHOICE { [0] IMPLICIT Header,
                                                [1] IMPLICIT Transfer, .
                                                [2] IMPLICIT Table-Header,
                                                [3] IMPLICIT Table-Transfer }

Header ::= SET {       [0] IMPLICIT Components OPTIONAL,
           CHOICE { [1] IMPLICIT Resolution OPTIONAL,
                    [2] IMPLICIT PixelPair OPTIONAL }
                    [3] IMPLICIT BitsPerDisplay OPTIONAL,
                    [4] IMPLICIT SamplingStructure OPTIONAL,
           CHOICE { [5] IMPLICIT Adpcm OPTIONAL,
                    [6] IMPLICIT Adct OPTIONAL } }

Components      ::= INTEGER { colorYU*V*   (0),
                              monochrome   (1) }

Resolution      ::= INTEGER { 4-2-2        (0),
                              2-1-1        (1) }

PixelPair       ::= SEQUENCE { PixHor, PixVer }

PixHor          ::= INTEGER
--   Number of horizontal pixels.

PixVer          ::= INTEGER
--   Number of vertical pixels.

BitsPerDisplay  ::= SEQUENCE OF INTEGER { 8 bits/pixel         (0),
                                          1 bit/pixel          (1),
                                          2 bits/pixel         (2),
                                          ... 9 bits/pixel     (9),... }
--   One value per component, gives the number or grey or colours a pixel may have.

SamplingStructure ::= SEQUENCE {
spatial              { INTEGER { line and orthogonal                  (0),
                                 line and orthogonal field quincunx   (1),
                                 line quincunx field orthogonal       (2),
                                 line orthogonal single field         (3),
                                 line quincunx single field           (4) }
temporal             { INTEGER { coincident                           (0),
                                 alternate samples                    (1),
                                 sequential line                      (2) } }

Adpcm           ::= SEQUENCE { INTEGER { Type dpcm                    (1) },
                               INTEGER { Subtype 1 dimension          (0) } }

Adct            ::= SEQUENCE { INTEGER { Type transform               (2) },
                               INTEGER { Subtype Cosine               (1) },
                               INTEGER { Subtype 2 dimension          (0) } }

Transfer        ::= SET { Origin, Area, Data }

Origin          ::=          [0] IMPLICIT PixelPair OPTIONAL

Area            ::=          [1] IMPLICIT PixelPair OPTIONAL

Data            ::= CHOICE { [2] IMPLICIT OCTETSTRING OPTIONAL,
--   Any value from 4/0 to 7/F.
                             [3] IMPLICIT OCTETSTRING OPTIONAL }
--   Transparent mode 8 bit/octet.

TableHeader     ::= SET { TableSet, TableSize }

TableSet        ::= [0] IMPLICIT SEQUENCE { type    ::= INTEGER,
                                            number  ::= INTEGER }

TableSize       ::= [1] IMPLICIT SEQUENCE { depth   ::= INTEGER,
                                            heigth  ::= INTEGER,
                                            width   ::= INTEGER OPTIONAL }

TableTransfer   ::= SET { TableSet, Position, Data }

Position        ::= TableSize
```

A.3.17 *Macro*

A Macro capability is available within the syntax of two of the three terminal data syntaxes. This capability permits strings of presentation data to be grouped together, so that it may be executed by the reference to a single command. In essence both terminal data syntax DS I and DS III provide the same Macro capability; however, a Macro in one data syntax cannot in general be converted to a Macro in another data syntax. This is because a Macro may contain any string of presentation data. Since the Terminal Models of the various data syntaxes differ, it is often necessary to sort the commands in the data stream in order to achieve the intended presentation effect. The arbitrary grouping of information into Macros prevents general sorting. Since the purpose of regular Macro functions are to achieve communications efficiency by eliminating the communication of repetitious code, it is possible to expand Macros in the conversion process. The conversion of a Macro is therefore the string of presentation data which it represents.

Two special forms of Macros in data syntax Ds I and DS III are Key Activated Macros and Transit Macros. Key Activated Macros link the execution of the Macro function to a local Key on the terminal. Since this operation depends upon the interaction of the user, the contents of the Macro cannot be expanded in the converter ahead of time. The converter must re-transmit the entire page of information to the terminal with the contents of the Key Activated Macro sorted and factored into the page. This problem must be handled by the Interworking Presentation Architecture. Similarly Transit Macro provides a problem in conversion. The contents of a Transit Macro must be sent back to the source upon a user interaction. In interworking this could mean that data syntax DS I data might be contained within a Transit Macro in a data syntax DS III terminal after a conversion so that it might be sent back to the source unchanged. It is necessary to be able to identify entire coding environments or to identify uniquely each code table in each Data Syntax in order to avoid confusion.

```
MACRO-String ::= CHOICE { [1] Define-Macro,
                          [2] Define-and-Execute-Macro,
                          [3] Define-Transmit-Macro,
                          [4] Define-End-of-Macro-Definition,
                          [5] Macro-Invocation }
```

-- Key Activated Macros are Macros with reference numbers 0 to 7 in data syntax DS III.

A.3.17.1 *Define macro*

```
Define-Macro              ::= SEQUENCE { SID, INTEGER }
```

-- General control character (DEF MACRO) from Recommendation T.101 DS III [C1 set position 4/0, (8/0)] and (P-DEF MACRO) from DS I [C1 set position 5/5, (9/5) followed by parameter 4/0].

-- Integer number from 0 to 95 correspondig to the Macro reference number of the Macro being defined.

A.3.17.2 *Define and execute macro*

```
Define-and-Execute-Macro  ::= SEQUENCE { SID, INTEGER }
```

-- General control character (DEFP MACRO) from Recommendation T.101 DS III [C1 set position 4/1, (8/1)] and (P-DEFP MACRO) from DS I [C1 set position 5/5, (9/5) followed by parameter 4/1].

-- Integer number from 0 to 95 correspondig to the Macro reference number of the Macro being defined.

A.3.17.3 *Define transit macro*

Define-Transmit-Macro ::= SEQUENCE { SID, INTEGER }

-- General control character (DEFT MACRO) from Recommendation T.101 DS III [C1 set position 4/2, (8/2)] and (P-DEFT MACRO) from DS I [C1 set position 5/5, (9/5) followed by parameter 4/2].

-- Integer number from 0 to 95 correspondig to the Macro reference number of the Macro being defined.

A.3.17.4 *Define end-of-macro definition*

Define-End-of-Macro-Definition ::= SID

-- General control character [END (Macro)] from Recommendation T.101 DS III [C1 set position 4/5, (8/5)] and (END MACRO) from DS I [C1 set position 5/5, (9/5) followed by parameter 4/15].

A.3.17.5 *Macro invocation*

Macro-Invocation ::= INTEGER

-- Integer number from 0 to 95 correspondig to the Macro reference number of the Macro being invoked.

-- *Note* – Macros may invoke other Macros at any time and to any depth.

A.3.18 *DRCS string*

The Dynamically Redefinable Character Set (DRCS) capability allows additional text or mosaic characters to be defined and used as regular alphanumeric text or mosaics. All three of the terminal data syntaxes include a form of DRCS capability; however, the operation of DRCS is quite different in the various Display Environments. In general it is not possible to convert exactly from one type of DRCS to another because of the boundary conditions imposed by each of the Terminal Data Syntaxes. Different limits exist on the number of DRCS characters which may be defined or the amount of memory which may be used to store DRCS characters. The definition of DRCS characters is a particular difficulty. One of the source terminal data syntaxes takes the approach of allowing any presentation information to be used in the definition of a DRCS character, including geometric drawing commands, bit (photographic) and text and even other DRCS characters. The other two source data syntaxes define DRCS characters using a bit oriented (photographic) approach. Even the two photographic approaches to the definition of DRCS are not equivalent since they have different pixel densities and serious quantization errors may result from mapping an array of pixels to another array of a different size. Three forms of DRCS definition are included in the Interworking Data Syntax to accommodate the requirements of the three source data syntaxes. The conversion process would therefore have sufficient information to make the best conversion possible.

```
DRCS-String  ::= CHOICE { [1] Define-DRCS-Type-I-1byte,
                          [2] Define-DRCS-Type-I-2byte,
                          [3] Define-DRCS-Type-II,
                          [4] Define-DRCS-Type-III,
                          [5] End-of-DRCS-Definition-Type-III,
                          [6] DRCS-Invocation,
                          [7] DRCS-Invocation-2byte }
```

A.3.18.1 *Define DRCS Type-I 1 byte*

```
Define-DRCS-Type-I-1byte ::= SEQUENCE { DRCS-I-Char-Size,
                                        DRCS-I-Code,
                                        DRCS-I-Data }

DRCS-I-Char-Size         ::= INTEGER { normal-size   (1),
                                       medium-size   (2),
                                       small-size    (3) }

DRCS-I-Code              ::= INTEGER
```

-- Integer number from 0 to 95 correspondig to the DRCS reference number of the 1 byte DRCS being invoked.

```
DRCS-I-Data              ::= BITSTRING
```

A.3.18.2 *Define DRCS Type-I 2 byte*

```
Define-DRCS-Type-I-2byte ::= SEQUENCE { DRCS-I-Char-Size,
                                        DRCS-I-Code,
                                        DRCS-I-Data }
```

-- This structure is the same as "Define-DRCS-Type-I-1byte" except that "DRCS-I-Code" is an integer number from 0 to 8835 correspondig to the DRCS reference number of the 2-byte DRCS being invoked.

A.3.18.3 *Define DRCS Type-II*

```
Define-DRCS-Type-II      ::= SEQUENCE { [1] IMPLICIT DRCS-Header OPTIONAL,
```

-- Description of general properties of the DRCS to be loaded. It is applied for all subsequent DRCS-pattern transfer units.

```
                                        [2] IMPLICIT DRCS-Pattern OPTIONAL }
```

-- Actual pattern data.

```
DRCS-Header              ::= SEQUENCE { Identification-of-Char-Set,
                                        Select-Dot-Composition }

Identification-of-Char-Set ::= SEQUENCE { repertory-info SET {
                                   repertory-# INTEGER { first repertory    (1),
                                                         second repertory   (2) },

                                   delete-existing-drcs BOOLEAN

                                 registration-info CHOICE {
                                   iso-registration     [1] IMPLICIT GRAPHICSTRING,
                                   private-drcs-#       [2] IMPLICIT INTEGER } }

Select-Dot-Composition   ::= SEQUENCE { Character-Cell-Structure,
                                        Blocking-Factor,
                                        Pixel-Characteristics }

Character-Cell-Structure ::= CHOICE { matrix-dimensions  [1] IMPLICIT SEQUENCE {
                                      horizontal             INTEGER,
                                      vertical               INTEGER },
```

-- According to SDC Type 1.

```
predefined-matrices [2] IMPLICIT INTEGER {
        n16*24    (0),
        n16*20    (1),
        n16*12    (2),
        n16*10    (3),
        n12*24    (4),
        n12*20    (5),
        n12*12    (6),
        n12*10    (7),
        n8*12     (8),
        n8*10     (9),
        n6*12     (10),
        n6*10     (11),
        n6*5      (12),
        n4*10     (13),
        n4*5      (14),
        n6*6      (15)}}
```

-- According to SDC Type-2.

```
Blocking-Factor ::= SEQUENCE { horizontal INTEGER,
                               vertical INTEGER }
```

-- Grouping of character cells, which are considered as a single character cell during character description.

```
Pixel-Characteristics ::= CHOICE { number of bits      [1] IMPLICIT INTEGER,
                                   predefined-numbers  [2] IMPLICIT INTEGER }
                                       basic-DRCS            (1),
```

-- 1 bit/dot.

```
                                       four-colour-DRCS      (4),
```

-- Black, red, green, yellow from 'Basic-Colour-Selection'.

```
                                       eight-colour-DRCS     (8),
```

-- First 8 colours from 'Basic-Colour-Selection'.

```
                                       sixteen-colour-DRCS   (16) }
```

-- 16 redefinable colours.

-- This data type describes the pattern for the characters of the down-loaded DRCS, according the last transmitted header unit. It contains no compression for the pattern data. Data Syntax I and III have no similar encodings and this method can be used for an adequate mapping. All encoding different from the direct method and the codes for improvement of the efficiency (S-bytes) have to be transformed to the following description.

```
DRCS-Pattern ::= SEQUENCE { first character GRAPHICSTRING,
```

-- Code of the first character or character block

```
        pattern-units SEQUENCE OF {
            pattern-block-#      SEQUENCE OF INTEGER,
            pattern-block        BIT STRING }}
```

-- Each pattern block contains one bit of each of the dots, starting from the top left hand corner, running row by row from left to right. The pattern block numbers are ordered from the least significant bit on. If the pattern block is preceded by two or more block numbers, the pattern block is applied to all of them. The block numbers are in the range of 0 to 'pixel-characteristics'-1. The length of the pattern block equals to the number of pixels in the block.

A.3.18.4 *Define DRCS Type-III*

Define-DRCS-Type-III ::= INTEGER

-- A function from Recommendation T.101 DS III [C1 set position 4/3, (8/3)].

-- Integer number from 0 to 95 correspondig to the DRCS reference number of the DRCS character being defined to be followed by data string.

A.3.18.5 *End-of-DRCS definition Type-III*

End-of-DRCS-Definition-Type-III ::= GRAPHICSTRING

-- General control character [END (DRCS)] from Recommendation T.101 DS III [C1 set position 4/5, (8/5)].

A.3.18.6 *DRCS invocation*

DRCS-Invocation ::= INTEGER

-- Integer number from 0 to 95 correspondig to the DRCS reference number of the DRCS being invoked.

A.3.18.7 *DRCS invocation 2 byte*

DRCS-Invocation-2byte ::= INTEGER

-- Integer number from 0 to 8835 correspondig to the DRCS reference number of the 2-byte DRCS being invoked.

A.3.19 *Fill pattern control string*

The capability to fill a geometrically defined area with an arbitrary Fill Pattern, interior style, hatch or texture, is provided in two of the source videotex data syntaxes. Since one of the terminal Videotex data syntaxes, data syntax DS I, does not provide this capability it must be accommodated in the conversion process by assigning distinguishing colours or other means to indicate the difference between patterned areas. The method by which this capability is supported in the other two source data syntaxes is quite different. Data syntax DS III provides four predefined texture patterns, including solid fill, and four redefinable texture masks. These masks are rectilinear and are referenced to the origin of the normalized display area. This means that abutting areas filled with the same pattern will align perfectly. In Interior style patterns defined in data syntax DS II, the pattern may be defined on a parallelogram shaped area and is referenced to the origin of the area. Data syntax DS II also provides eight predefined fill patterns (hatch patterns). In general, any texture of interior style pattern may be simulated in the conversion process; however, secondary effects such as exact alignment of patterns cannot be guaranteed. Texture patterns in data syntax DS III are defined by including any string of presentation data in the definition of the pattern whereas interior styles defined in data syntax DS II are defined in terms of a cell array. The conversion process must resolve the pattern before the conversion. Limits to global variables, such as the available amount of texture memory, is defined by the boundary value condition indicators in the state vector.

```
Fill-Pattern-Control-String ::= CHOICE { [1] Define-Texture,
                                         [2] End-of-Texture-Definition,
                                         [3] Texture-Mask-Size,
                                         [4] Set-Pattern-Representation,
                                         [5] Pattern-Selection }
```

A.3.19.1 *Define texture*

Define-Texture ::= INTEGER

-- A function from Recommendation T.101 DS III [C1 set position 4/4, (8/4)].

-- Integer number from 4 to 7 correspondig to the redefinable Texture Mask to be defined. Note texture masks 0 to 3 are predefined and cannot be redefined to be followed by data string.

A.3.19.2 *End-of-texture definition*

End-of-Texture-Definition ::= GRAPHICSTRING

-- General control character [END (TEXTURE)] from Recommendation T.101 DS III [C1 set position 4/5, (8/5)].

A.3.19.3 *Texture mask size*

Texture-Mask-Size ::= Rel-Coord

-- Establish the texture mask size up to the limit defined by the boundary conditions.

-- A function from Recommendation T.101 DS III [C1 set position 2/3, (10/3)].

A.3.19.4 *Set pattern representation*

Set-Pattern-Representation ::= SEQUENCE { pattern-index INTEGER,

-- This number corresponds to the current pattern definition. It can be referenced by subsequent fill area style indices.

delta-x INTEGER,
delta-y INTEGER,

-- A grid of delta-x* delta-y* cells is specified. The colour of each cell is individually given by the

pattern-cell-data Pixel-Array-Data }

-- The colour array is associated with the cells as follows: the element (1, delta-y) is associated with the cell having the pattern reference point at one corner. Elements with increasing first dimension are associated with succesive cells in the direction of the pattern with vector; elements with decreasing second dimension are associated with succesive cells in the direction of the pattern height vector.

-- These definitions of patterns are from DS II and are applicable in conjugation with the fill area attributes defined by the data type "Display-Element-Attributes".

A.3.19.5 *Pattern selection*

Pattern-Selection ::= INTEGER

-- Integer number from 4 to 7 correspondig to the Texture Mask being selected.

A.3.20 *Music string*

The music capability is an option unique to only one of the terminal data syntaxes. It requires special capabilities for presentation and cannot be converted in any reasonable manner. Music information is included in the Interworking Data Syntax for future compatibility so that interworking may be accomplished between information from data syntax DS I and any future versions of data syntax DS II or III which might include a music capability.

```
Music-String ::= CHOICE { [1] Music-Code-Sequence,
                          [2] Music-Control-Sequence }
```

A.3.20.1 *Music code sequence*

Music-Code-Sequence ::= GRAPHICSTRING

-- Characters from Recommendation T.101 DS I [Musical Tone Set (pitch/duration)]. Note that the Musical Tone set is a two byte set which can be described as the combination of two one byte sets, one for duration and one for pitch. Reference is made to Recommendation T.101 since this code table has not yet been registered.

A.3.20.2 *Music code sequence*

Music-Control-Sequence ::= GRAPHICSTRING

-- Control characters from Recommendation T.101 DS I (Musical Control C1 Set). The Musical Control set contains the functions: Start Music Sequence, End Music Sequence, Start Melody Part, Start Rythm Part, End Part, Music Label, Jump to Part, Music Repeat, Music Branch, Sound Level, Change of Timbre, Long Duration Rest or Tone. Reference is made to Recommendation T.101 since this code table has not yet been registered.

A.3.21 *Telesoftware string*

Telesoftware-String ::= Further Study

A.3.22 *Audio data string*

Audio-Data-String ::= Further Study

APPENDIX I

(to Recommendation T.101)

Text and mosaic character repertoires

In the Interworking Data Syntax all text and mosaic characters are assigned a code name so that they may be uniquely identified. An exhaustive Repertoire of all of the text and mosaic characters used in the data syntaxes specified in Recommendation T.101 is presented below. This simplifies many of the references to graphic character sets used in the IDS since only the code names need be used in the body of the ASN.1 description of the Interworking Data Syntax. None of the videotex data syntaxes make use of all of the text and mosaic characters identified below. There are areas where there is a large amount of overlap between the various data syntaxes. In order to aid transcoding and conversions several categories have been identified. Separate repertoires have been defined for each of these categories.

I.1 *Repertoire I – Common alphanumeric text characters*

Repertoire I contains the common repertoire of basic alphanumeric text characters. These characters are taken from the primary and supplementary characters from Recommendation T.51, with registered final characters 4/0 and 6/2 respectively. In addition this includes the SPACE character (SP01) and the DELETE character (SM34). The descriptive names given to characters differ between the various terminal data syntaxes defined in Recommendation T.101 and between the repertoire of alphanumeric characters defined in ISO standards ISO 6937. Composite names are used here, which endeavour to include the full range of meanings specified in the different terminal data syntaxes identified in Recommendation T.101, and achieve the maximum level of commonality with ISO 6937.

I.1.1 *Latin alphabetic characters*

Name Code	*Descriptive name*	*Name Code*	*Descriptive name*
LA01	small a	LC15	small c with circumflex accent
LA02	capital A	LC16	capital C with circumflex accent
LA11	small a with acute accent	LC21	small c with caron
LA12	capital A with acute accent	LC22	capital C with caron
LA13	small a with grave accent	LC29	small c with dot above
LA14	capital A with grave accent	LC30	capital C with dot above
LA15	small a with circumflex accent	LC41	small c with cedilla
LA16	capital A with circumflex accent	LC42	capital C with cedilla
LA17	small a with diaeresis or umlaut mark	LD01	small d
LA18	capital A with diaeresis or umlaut mark	LD02	capital D
LA19	small a with tilde	LD21	small d with caron
LA20	capital A with tilde	LD22	capital D with caron
LA23	small a with breve	LD61	small d with stroke
LA24	capital A with breve	LD62	capital D with stroke (Icelandic eth)
LA27	small a with ring	LD63	small eth, Icelandic
LA28	capital A with ring	LE01	small e
LA31	small a with macron	LE02	capital E
LA32	capital A with macron	LE11	small e with acute accent
LA43	small a with ogonek	LE12	capital E with acute accent
LA44	capital A with ogonek	LE13	small e with grave accent
LA51	small æ diphthong	LE14	capital E with grave accent
LA52	capital Æ diphthong	LE15	small e with circumflex accent
LB01	small b	LE16	capital E with circumflex accent
LB02	capital B	LE17	small e with diaeresis or umlaut
LC01	small c	LE18	capital E with diaeresis or umlaut
LC02	capital C	LE21	small e with caron
LC11	small c with acute accent	LE22	capital E with caron
LC12	capital C with acute accent	LE29	small e with dot above

Name Code	Descriptive name
LE30	capital E with dot above
LE31	small e with macron
LE32	capital E with macron
LE43	small e with ogonek
LE44	capital E with ogonek
LF01	small f
LF02	capital F
LG01	small g
LG02	capital G
LG11	small g with acute accent
LG15	small g with circumflex accent
LG16	capital G with circumflex accent
LG23	small g with breve
LG24	capital G with breve
LG29	small g with dot above
LG30	capital G with dot above
LG42	capital G with cedilla
LH01	small h
LH02	capital H
LH15	small h with circumflex accent
LH16	capital H with circumflex accent
LH61	small h with stroke
LH62	capital H with stroke
LI01	small i
LI02	capital I
LI11	small i with acute accent
LI12	capital I with acute accent
LI13	small i with grave accent
LI14	capital I with grave accent
LI15	small i with circumflex accent
LI16	capital I with circumflex accent
LI17	small i with diaeresis or umlaut mark
LI18	capital I with diaeresis or umlaut mark
LI19	small i with tilde
LI20	capital I with tilde
LI30	capital I with dot above
LI31	small i with macron
LI32	capital I with macron
LI43	small i with ogonek
LI44	capital I with ogonek
LI51	small ij ligature
LI52	capital IJ ligature
LI61	small i without dot
LJ01	small j
LJ02	capital J
LJ15	small j with circumflex accent
LJ16	capital J with circumflex accent
LK01	small k
LK02	capital K
LK41	small k with cedilla
LK42	capital K with cedilla
LK61	small k, Greenlandic
LL01	small l
LL02	capital L
LL11	small l with acute accent

Name Code	Descriptive name
LL12	capital L with acute accent
LL21	small l with caron
LL22	capital L with caron
LL41	small l with cedilla
LL42	capital L with cedilla
LL61	small l with stroke
LL62	capital L with stroke
LL63	small l with middle dot
LL64	capital L with middle dot
LM01	small m
LM02	capital M
LN01	small n
LN02	capital N
LN11	small n with acute accent
LN12	capital N with acute accent
LN19	small n with tilde
LN20	capital N with tilde
LN21	small n with caron
LN22	capital N with caron
LN41	small n with cedilla
LN42	capital N with cedilla
LN61	small eng, Lapp
LN62	capital eng, Lapp
LN63	small n with apostrophe
LO01	small o
LO02	capital O
LO11	small o with acute accent
LO12	capital O with acute accent
LO13	small o with grave accent
LO14	capital O with grave accent
LO15	small o with circumflex accent
LO16	capital O with circumflex accent
LO17	small o with diaeresis or umlaut mark
LO18	capital O with diaeresis or umlaut mark
LO19	small o with tilde
LO20	capital O with tilde
LO25	small o with double acute accent
LO26	capital O with double acute accent
LO31	small o with macron
LO32	capital O with macron
LO51	small œ ligature
LO52	capital Œ ligature
LO61	small o with slash
LO62	capital O with slash
LP01	small p
LP02	capital P
LQ01	small q
LQ02	capital Q
LR01	small r
LR02	capital R
LR11	small r with acute accent
LR12	capital R with acute accent
LR21	small r with caron
LR22	capital R with caron
LR41	small r with cedilla

Name Code	Descriptive name
LR42	capital R with cedilla
LS01	small s
LS02	capital S
LS11	small s with acute accent
LS12	capital S with acute accent
LS15	small s with circumflex accent
LS16	capital S with circumflex accent
LS21	small s with caron
LS22	capital S with caron
LS41	small s with cedilla
LS42	capital S with cedilla
LS61	small sharp s, German
LT01	small t
LT02	capital T
LT21	small t with caron
LT22	capital T with caron
LT41	small t with cedilla
LT42	capital T with cedilla
LT61	small t with stroke
LT62	capital T with stroke
LT63	small thorn, Icelandic
LT64	capital thorn, Icelandic
LU01	small u
LU02	capital U
LU11	small u with acute accent
LU12	capital U with acute accent
LU13	small u with grave accent
LU14	capital U with grave accent
LU15	small u with circumflex accent
LU16	capital U with circumflex accent
LU17	small u with diaeresis or umlaut mark
LU18	capital U with diaeresis or umlaut mark
LU19	small u with tilde
LU20	capital U with tilde
LU23	small u with breve
LU24	capital U with breve
LU25	small u with double acute accent
LU26	capital U with double acute accent
LU27	small u with ring
LU28	capital U with ring
LU31	small u with macron
LU32	capital U with macron
LU43	small u with ogonek
LU44	capital U with ogonek
LV01	small v
LV02	capital V
LW01	small w
LW02	capital W
LW15	small w with circumflex accent
LW16	capital W with circumflex accent
LX01	small x
LX02	capital x
LY01	small y
LY02	capital Y
LY11	small y with acute accent
LY12	capital Y with acute accent
LY15	small y with circumflex accent
LY16	capital Y with circumflex accent
LY17	small y with diaeresis or umlaut mark
LY18	capital Y with diaeresis or umlaut mark
LZ01	small z
LZ02	capital Z
LZ11	small z with acute accent
LZ12	capital Z with acute accent
LZ21	small z with caron
LZ22	capital Z with caron
LZ29	small z with dot above
LZ30	capital Z with dot above

I.1.2 *Non-alphabetic characters*

I.1.2.1 *Decimal digits*

Name Code	Descriptive name
ND01	digit 1
ND02	digit 2
ND03	digit 3
ND04	digit 4
ND05	digit 5
ND06	digit 6
ND07	digit 7
ND08	digit 8
ND09	digit 9
ND10	digit 0

I.1.2.2 *Currency signs*

Name Code	Descriptive name
SC01	general currency sign
SC02	pound sign
SC03	dollar sign
SC04	cent sign
SC05	yen sign

I.1.2.3 *Punctuation marks*

Name Code	*Descriptive name*	*Name Code*	*Descriptive name*
SP01	SPACE	SP13	colon
SP02	exclamation mark	SP14	semicolon
SP03	inverted exclamation mark	SP15	question mark
SP04	quotation mark	SP16	inverted question mark
SP05	apostrophe	SP17	angle quotation mark left
SP06	opening (left) parenthesis	SP18	angle quotation mark right
SP07	closing (right) parenthesis	SP19	single quotation mark left
SP08	comma	SP20	single quotation mark right
SP10	hyphen or minus sign	SP21	double quotation mark left
SP11	period (or decimal point)	SP22	double quotation mark right
SP12	solidus (slant)		

I.1.2.4 *Arithmetic signs*

Name Code	*Descriptive name*	*Name Code*	*Descriptive name*
SA01	plus sign	SA05	greater-than sign
SA02	plus/minus sign	SA06	divide sign
SA03	less-than sign	SA07	multiply sign
SA04	equals sign		

I.1.2.5 *Subscripts and superscripts*

Name Code	*Descriptive name*	*Name Code*	*Descriptive name*
NS01	superscript 1	NS03	superscript 3
NS02	superscript 2		

I.1.2.6 *Fractions*

Name Code	*Descriptive name*	*Name Code*	*Descriptive name*
NF01	fraction 1/2	SM40	fraction 3/8 (equivalent to NF19)
NF04	fraction 1/4	SM41	fraction 5/8 (equivalent to NF20)
NF05	fraction 3/4	SM42	fraction 7/8 (equivalent to NF21)
SM39	fraction 1/8 (equivalent to NF18)		

I.1.2.7 *Miscellaneous*

Name Code	*Descriptive name*	*Name Code*	*Descriptive name*
SM01	number sign	SM17	micro sign
SM02	percent sign	SM18	ohm sign
SM03	ampersand	SM19	degree sign
SM04	star, asterisk	SM20	ordinal indicator, masculine
SM05	commercial at	SM21	ordinal indicator, feminine
SM06	opening (left) square bracket	SM24	section sign
SM07	reverse solidus	SM25	paragraph sign, pilcrow
SM08	closing (right) square bracket	SM26	middle dot
SM11	opening brace, left curly bracket	SM30	leftward arrow
SM12	central horizonal bar jointive	SM31	rightward arrow
SM13	central vertical line jointive	SM32	upward arrow
SM14	closing brace, right curly bracket	SM33	downward arrow

Name Code	Descriptive name
SM34	DELETE
SM35	registered sign (equivalent to SM53)
SM36	copyright sign (equivalent to SM52)
SM37	trademark sign (equivalent to SM54)
SM38	musical symbol (equivalent to SM93)
SM43	arrowhead upwards, circumflex shape
SM44	upper reverse solidus, grave accent shape
SM47	upper bar (not jointive) bar or tilde shape
SM48	lower bar (not jointive) low line, spacing underline (equivalent to SP09 of ISO 6937)
SM49	non spacing underline

Note — The characters SM43, SM44, SM47, SM48 have multiple names since the descriptive names for these characters differ significantly between the various terminal data syntaxes defined in Recommendation T.101. These characters were originally intended as accent characters in the original usage of the IRV code table of ISO 646. This meaning has changed since the introduction of the composite method of coding accented characters defined in CCITT Recommendation T.51, ISO 6937 and several CCITT Recommendations. As a result these characters should not be used to generated accented characters. To retain compatibility with previous standards, multiple names are described here.

I.1.2.8 *Diacritical marks (as displayed when used in conjunction with SPACE)*

Name Code	Descriptive name
SD11	acute accent
SD13	grave accent
SD15	circumflex accent
SD17	umlaut or diaeresis
SD19	tilde
SD21	caron
SD23	breve
SD25	double acute accent
SD27	ring
SD29	dot above
SD31	macron
SD41	cedilla
SD43	ogonek

I.2 *Repertoire 2 — Special alphanumeric text characters*

These characters are unique to one or two of the terminal data syntaxes and the presentation of these characters must be converted so that their presentation effect may be achieved in other terminal data syntax.

Name Code	Descriptive name
SM45	left vertical bar jointive
SM46	right vertical bar jointive
SM50	non spacing vector overbar
SM51	non spacing slant

Note — The name codes SM50 and SM51 are introduced here since name codes for these characters are not included in either the ISO registry (Registered code table number 99) or in CCITT Recommendation T.101.

I.3 *Repertoire 3 — Kana characters*

This entire set of characters is unique to only one of the terminal data syntaxes and therefore the presentation of these characters must be converted so that their presentation effect may be achieved in another terminal data syntax.

Name Code	Descriptive name
JA01	Katakana full stop
JA02	Katakana opening bracket
JA03	Katakana closing bracket
JA04	Katakana comma
JA05	Katakana conjunctive symbol
JA06	Katakana WO
JA07	Katakana small a
JA08	Katakana small i
JA09	Katakana small u
JA10	Katakana small e
JA11	Katakana small o
JA12	Katakana ya

Name Code	Descriptive name
JA13	Katakana small yu
JA14	Katakana yo
JA15	Katakana tsu
JA16	Prolonged Sound Symbol
JA17	Katakana A
JA18	Katakana I
JA19	Katakana U
JA20	Katakana E
JA21	Katakana O
JA22	Katakana KA
JA23	Katakana KI
JA24	Katakana KU
JA25	Katakana KE
JA26	Katakana KO
JA27	Katakana SA
JA28	Katakana SHI
JA29	Katakana SU
JA30	Katakana SE
JA31	Katakana SO
JA32	Katakana TA
JA33	Katakana CHI
JA34	Katakana TSU
JA35	Katakana TE
JA36	Katakana TO
JA37	Katakana NA
JA38	Katakana NI

Name Code	Descriptive name
JA39	Katakana NU
JA40	Katakana NE
JA41	Katakana NO
JA42	Katakana HA
JA43	Katakana HI
JA44	Katakana FU
JA45	Katakana HE
JA46	Katakana HO
JA47	Katakana MA
JA48	Katakana MI
JA49	Katakana MU
JA50	Katakana ME
JA51	Katakana MO
JA52	Katakana YA
JA53	Katakana YU
JA54	Katakana YO
JA55	Katakana RA
JA56	Katakana RI
JA57	Katakana RU
JA58	Katakana RE
JA59	Katakana RO
JA60	Katakana WA
JA61	Katakana N or M
JA62	Voiced sound symbol
JA63	semi-voiced sound symbol

Note – Since name codes are not included for these characters in either the ISO registry (Registration number 13) or in CCITT Recommendation T.101, codes beginning with JA are introduced here.

I.4 *Repertoire 4 – Kanji characters*

These characters occur in only one of the terminal data syntaxes although the registered Kanji character code table is sometimes used in combination with the sets of another terminal data syntax. Except in the case when Kanji character capability is available at both ends of an interchange, the presentation of these characters must be converted so that their presentation effect may be achieved in the other terminal data syntax. The Kanji character set is registered as a two byte set. CCITT Recommendation T.101 uses a sub-Repertoire of this set which includes some 3639 characters, of which 2980 are Kanji symbol characters. These characters are unique to this Repertoire. The remaining characters, including a number of special characters as well as some alphabetics for other languages, overlap with other registered character sets. For example a Cyrillic, Greek, Katakana, and Hiragana alphabet are included, as well as all but 11 of the Latin alphabet characters of Repertoire 1. The overlap characters in this code table have the same *name codes* as the equivalent characters in Repertoire 1 and can therefore be converted directly between data syntaxes. Special handling is required for the other characters. In addition there are 32 drawing characters which overlap those in Repertoire 8. These also share the same *name codes* as those in other character sets and therefore are part of the greater drawing character Repertoire (Repertoire 8).

The Repertoire of Kanji pictorial characters and unique special characters is voluminous. Since conversions from this Repertoire of characters require special handling, it is not necessary to list the Repertoire here. For reference, see CCITT Recommendation T.101 Data Syntax I which is a sub Repertoire of the ISO registration number 87. The name codes Kanji JK01 to JK2980, HK01 to HK83, and JS01 to JS366 have been introduced to identify the Kanji pictorial characters, the Hiragana characters, and the Kanji Special characters respectively.

I.5 *Repertoire 5 – Greek characters*

Name Code	*Descriptive name*
GA01	Lower case greek letter Alpha
GA02	Upper case greek letter Alpha
GA11	Lower case greek letter Alpha with accent
GA12	Upper case greek letter Alpha with accent
GB01	Lower case greek letter Beta
GB02	Upper case greek letter Beta
GD01	Lower case greek letter Delta
GD02	Upper case greek letter Delta
GE01	Lower case greek letter Epsilon
GE02	Upper case greek letter Epsilon
GE11	Lower case greek letter Epsilon with accent
GE12	Upper case greek letter Epsilon with accent
GE61	Lower case greek letter Eta
GE62	Upper case greek letter Eta
GE63	Lower case greek letter Eta with accent
GE64	Upper case greek letter Eta with accent
GF01	Lower case greek letter Phi
GF02	Upper case greek letter Phi
GG01	Lower case greek letter Gamma
GG02	Upper case greek letter Gamma
GH01	Lower case greek letter Khi
GH02	Upper case greek letter Khi
GI01	Lower case greek letter Iota
GI02	Upper case greek letter Iota
GI11	Lower case greek letter Iota with accent
GI12	Upper case greek letter Iota with accent
GI17	Lower case greek letter Iota with diaeresis
GI18	Upper case greek letter Iota with diaeresis
GI33	Lower case greek letter Iota with accent and diaeresis
GK01	Lower case greek letter Kappa
GK02	Upper case greek letter Kappa
GL01	Lower case greek letter Lambda
GL02	Upper case greek letter Lambda
GM01	Lower case greek letter Mu
GM02	Upper case greek letter Mu
GN01	Lower case greek letter Nu
GN02	Upper case greek letter Nu
GO01	Lower case greek letter Omicron
GO02	Upper case greek letter Omicron
GO11	Lower case greek letter Omicron with accent
GO12	Upper case greek letter Omicron with accent
GO61	Lower case greek letter Omega
GO62	Upper case greek letter Omega
GO63	Lower case greek letter Omega with accent
GO64	Upper case greek letter Omega with accent
GP01	Lower case greek letter Pi
GP02	Upper case greek letter Pi
GP61	Lower case greek letter Psi
GP62	Upper case greek letter Psi
GR01	Lower case greek letter Rho
GR02	Upper case greek letter Rho
GS01	Lower case greek letter Sigma
GS02	Upper case greek letter Sigma
GS61	Lower case greek letter final Sigma
GT01	Lower case greek letter Tau
GT02	Upper case greek letter Tau
GT61	Lower case greek letter Theta
GT62	Upper case greek letter Theta
GU01	Lower case greek letter Upsilon
GU02	Upper case greek letter Upsilon
GU11	Lower case greek letter Upsilon with accent
GU12	Upper case greek letter Upsilon with accent
GU17	Lower case Upsilon with diaeresis
GU18	Upper case Upsilon with diaeresis
GU33	Lower case Upsilon with accent and diaeresis
GX01	Lower case greek letter Xi
GX02	Upper case greek letter Xi
GZ01	Lower case greek letter Zeta
GZ02	Upper case greek letter Zeta
SD33	Diaeresis and accent sign with space

I.6 *Repertoire 6 – Cyrillic characters*

Name Code	*Descriptive name*
KA01	small cyrillic a
KA02	capital cyrillic a
KA61	small cyrillic ja
KA62	capital cyrillic ja
KB01	small cyrillic be
KB02	capital cyrillic be
KC01	small cyrillic tse
KC02	capital cyrillic tse
KC21	small cyrillic tche
KC22	capital cyrillic tche
KC61	small cyrillic chtcha
KC62	capital cyrillic chtcha
KD01	small cyrillic de
KD02	capital cyrillic de
KE01	small cyrillic je
KE02	capital cyrillic je
KE17	small cyrillic jo
KE18	capital cyrillic jo
KE61	small cyrillic e
KE62	capital cyrillic e
KF01	small cyrillic eff
KF02	capital cyrillic eff
KG01	small cyrillic gue
KG02	capital cyrillic gue
KH01	small cyrillic kha
KH02	capital cyrillic kha
KI01	small cyrillic i
KI02	capital cyrillic i

Name Code	*Descriptive name*
KI23	small cyrillic breve i
KI24	capital cyrillic breve i
KJ01	small cyrillic zhe
KJ02	capital cyrillic zhe
KL01	small cyrillic el
KL02	capital cyrillic el
KM01	small cyrillic em
KM02	capital cyrillic em
KN01	small cyrillic en
KN02	capital cyrillic en
KO01	small cyrillic o
KO02	capital cyrillic o
KP01	small cyrillic Pe
KP02	capital cyrillic Pe
KR01	small cyrillic er
KR02	capital cyrillic er
KS01	small cyrillic es
KS02	capital cyrillic es
KS23	small cyrillic sha

Name Code	*Descriptive name*
KS24	capital cyrillic sha
KT01	small cyrillic te
KT02	capital cyrillic te
KU01	small cyrillic v
KU02	capital cyrillic v
KV61	small cyrillic ju
KV62	capital cyrillic ju
KV01	small cyrillic ve
KV02	capital cyrillic ve
KY01	small cyrillic ieri
KY02	capital cyrillic ieri
KY61	small cyrillic ier (miagkil zhak)
KY62	capital cyrillic ier (miagkil zhak)
KY63	small cyrillic ier' (tvjordy zhak)
KY64	capital cyrillic ier' (tvjordy zhak)
KZ01	small cyrillic ze
KZ02	capital cyrillic ze
KK01	small cyrillic ka
KK02	capital cyrillic ka

I.7 *Repertoire 7 – Block mosaic characters*

Block mosaic characters are a sub-Repertoire of the mosaics used in each of the terminal data syntaxes and therefore are directly translatable between data syntaxes. Even the coding of the block mosaic characters is identical except for the Filled Mosaic character. A Filled Mosaic is included in Repertoire 5. Either of the two equivalent codings established in Recommendation T.101 (code table position 5/15 or 7/15) for the Filled Mosaic character is acceptable in a string of characters from Repertoire 5.

Note – The following convention is used to describe the shape of a 2 sub-cell wide by 3 sub-cell high mosaic character. The cells are numbered from the upper left, to upper right, centre left, centre right, lower left, and lower right as 1,2,3,4,5,6 respectively. This code is used in identifying each character in the Repertoire.

1	2
3	4
5	6

T0803710-89

Mosaic sub-cell structure

Name Code	*Descriptive name*
MG00	Mosaic empty character
MG01	Block Mosaic 1
MG02	Block Mosaic 2
MG03	Block Mosaic 1,2
MG04	Block Mosaic 3
MG05	Block Mosaic 1,3
MG06	Block Mosaic 2,3
MG07	Block Mosaic 1,2,3
MG08	Block Mosaic 4

Name Code	*Descriptive name*
MG09	Block Mosaic 1,4
MG10	Block Mosaic 2,4
MG11	Block Mosaic 1,2,4
MG12	Block Mosaic 3,4
MG13	Block Mosaic 1,3,4
MG14	Block Mosaic 2,3,4
MG15	Block Mosaic 1,2,3,4
MG16	Block Mosaic 5
MG17	Block Mosaic 1,5
MG18	Block Mosaic 2,5

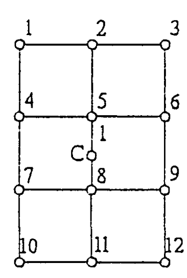
Name Code	Descriptive name
MG19	Block Mosaic 1,2,5
MG20	Block Mosaic 3,5
MG21	Block Mosaic 1,3,5
MG22	Block Mosaic 2,3,5
MG23	Block Mosaic 1,2,3,5
MG24	Block Mosaic 4,5
MG25	Block Mosaic 1,4,5
MG26	Block Mosaic 2,4,5
MG27	Block Mosaic 1,2,4,5
MG28	Block Mosaic 3,4,5
MG29	Block Mosaic 1,3,4,5
MG30	Block Mosaic 2,3,4,5
MG31	Block Mosaic 1,2,3,4,5
MG32	Block Mosaic 6
MG33	Block Mosaic 1,6
MG34	Block Mosaic 2,6
MG35	Block Mosaic 1,2,6
MG36	Block Mosaic 3,6
MG37	Block Mosaic 1,3,6
MG38	Block Mosaic 2,3,6
MG39	Block Mosaic 1,2,3,6
MG40	Block Mosaic 4,6
MG41	Block Mosaic 1,4,6

Name Code	Descriptive name
MG42	Block Mosaic 2,4,6
MG43	Block Mosaic 1,2,4,6
MG44	Block Mosaic 3,4,6
MG45	Block Mosaic 1,3,4,6
MG46	Block Mosaic 2,3,4,6
MG47	Block Mosaic 1,2,3,4,6
MG48	Block Mosaic 5,6
MG49	Block Mosaic 1,5,6
MG50	Block Mosaic 2,5,6
MG51	Block Mosaic 1,2,5,6
MG52	Block Mosaic 3,5,6
MG53	Block Mosaic 1,3,5,6
MG54	Block Mosaic 2,3,5,6
MG55	Block Mosaic 1,2,3,5,6
MG56	Block Mosaic 4,5,6
MG57	Block Mosaic 1,4,5,6
MG58	Block Mosaic 2,4,5,6
MG59	Block Mosaic 1,2,4,5,6
MG60	Block Mosaic 3,4,5,6
MG61	Block Mosaic 1,3,4,5,6
MG62	Block Mosaic 2,3,4,5,6
MG63	Filled Mosaic 1,2,3,4,5,6 (also equivalent to MG64-T.101 DSIII)

I.8 *Repertoire 8 – Sub cell aligned smooth mosaics 1*

A general set of Sub Cell Aligned Smooth Mosaic characters are part of the Repertoire of mosaic characters used in two of the terminal data syntaxes, and have the same coding. These characters are identified separately as a Repertoire of the interworking data syntax since these characters are directly translatable between two of the data syntaxes. The presentation of these characters must be converted so that their presentation effect may be achieved in a terminal using a data syntax which does not support these characters.

Note – The following convention is used to describe the shape of a 2 sub-cell wide by 3 sub-cell high mosaic character containing Sub Cell Aligned Smooth Mosaics. The vertices of the sub-cells are numbered from the upper sub-cell upper left corner as illustrated below. The smooth mosaic is filled below (B), above (A) to the right (R) or the left (L) of a line or lines through the indicated vertices.



Mosaic sub-cell vertices

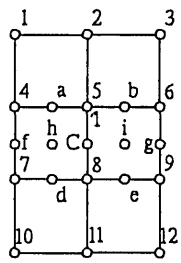
Name Code	Descriptive name
SG01	Sub Cell Aligned Smooth Mosaic B-7,11
SG02	Sub Cell Aligned Smooth Mosaic B-7,12
SG03	Sub Cell Aligned Smooth Mosaic B-4,11
SG04	Sub Cell Aligned Smooth Mosaic B-4,12
SG05	Sub Cell Aligned Smooth Mosaic B-1,11
SG06	Sub Cell Aligned Smooth Mosaic B-1,12
SG07	Sub Cell Aligned Smooth Mosaic B-4,2
SG08	Sub Cell Aligned Smooth Mosaic B-4,3
SG09	Sub Cell Aligned Smooth Mosaic B-7,2
SG10	Sub Cell Aligned Smooth Mosaic B-7,3
SG11	Sub Cell Aligned Smooth Mosaic B-10,2
SG12	Sub Cell Aligned Smooth Mosaic B-7,6
SG13	Sub Cell Aligned Smooth Mosaic R-1,c,10
SG14	Sub Cell Aligned Smooth Mosaic B-1,c,3
SG15	Sub Cell Aligned Smooth Mosaic A-10,c,12
SG16	Sub Cell Aligned Smooth Mosaic L-3,c,12
SG17	Sub Cell Aligned Smooth Mosaic B-4,9
SG18	Sub Cell Aligned Smooth Mosaic B-2,12
SG19	Sub Cell Aligned Smooth Mosaic B-1,9
SG20	Sub Cell Aligned Smooth Mosaic B-2,9
SG21	Sub Cell Aligned Smooth Mosaic B-1,6
SG22	Sub Cell Aligned Smooth Mosaic B-2,6
SG23	Sub Cell Aligned Smooth Mosaic B-10,3
SG24	Sub Cell Aligned Smooth Mosaic B-11,3
SG25	Sub Cell Aligned Smooth Mosaic B-10,6
SG26	Sub Cell Aligned Smooth Mosaic B-11,6
SG27	Sub Cell Aligned Smooth Mosaic B-10,9
SG28	Sub Cell Aligned Smooth Mosaic B-11,9
SG29	Sub Cell Aligned Smooth Mosaic A-7,11
SG30	Sub Cell Aligned Smooth Mosaic A-7,12
SG31	Sub Cell Aligned Smooth Mosaic A-4,11

Name Code	Descriptive name
SG32	Sub Cell Aligned Smooth Mosaic A-4,12
SG33	Sub Cell Aligned Smooth Mosaic A-1,11
SG34	Sub Cell Aligned Smooth Mosaic A-1,12
SG35	Sub Cell Aligned Smooth Mosaic A-4,2
SG36	Sub Cell Aligned Smooth Mosaic A-4,3
SG37	Sub Cell Aligned Smooth Mosaic A-7,2
SG38	Sub Cell Aligned Smooth Mosaic A-7,3
SG39	Sub Cell Aligned Smooth Mosaic A-10,2
SG40	Sub Cell Aligned Smooth Mosaic A-7,6
SG41	Sub Cell Aligned Smooth Mosaic L-1,c,10
SG42	Sub Cell Aligned Smooth Mosaic A-1,c,3
SG43	Sub Cell Aligned Smooth Mosaic A-11,9
SG44	Sub Cell Aligned Smooth Mosaic A-10,9
SG45	Sub Cell Aligned Smooth Mosaic A-11,6
SG46	Sub Cell Aligned Smooth Mosaic A-10,6
SG47	Sub Cell Aligned Smooth Mosaic A-11,3
SG48	Sub Cell Aligned Smooth Mosaic A-10,3
SG49	Sub Cell Aligned Smooth Mosaic A-2,6
SG50	Sub Cell Aligned Smooth Mosaic A-1,6
SG51	Sub Cell Aligned Smooth Mosaic A-2,9
SG52	Sub Cell Aligned Smooth Mosaic A-1,9
SG53	Sub Cell Aligned Smooth Mosaic A-2,12
SG54	Sub Cell Aligned Smooth Mosaic A-4,9
SG55	Sub Cell Aligned Smooth Mosaic R-3,c,12
SG56	Sub Cell Aligned Smooth Mosaic B-10,c,12

I.9 *Repertoire 9 – General smooth mosaics*

An additional form of Smooth Mosaic characters are available in only one of the terminal data syntaxes. These smooth mosaic characters align with half sub-cell boundaries. Since these characters are unique to only one of the terminal data syntaxes, the presentation of these characters must be converted so that their presentation may be achieved in another terminal data syntax.

Note – The notation used to describe these characters is the same as that used in Repertoire 6 except that intermediate points which identify to the half sub cell are introduced. In the case that four numbers are given, such as for the characters (SG58 and SG71), the area is bounded on all sides by the four sub-cell positions. In two special cases, for the characters (SG68 and SG81), two areas are filled within the mosaic cell.



T0803730-89

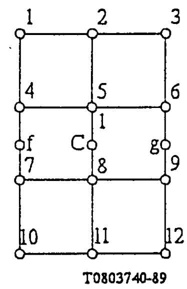
Mosaic sub-cell vertices and intermediate points

Name Code	*Descriptive name*	*Name Code*	*Descriptive name*
MS01	General Smooth Mosaic B-f,c	MS18	General Smooth Mosaic A-e,d & B-a,d
MS03	General Smooth Mosaic B-h,i	MS19	General Smooth Mosaic A-h,i
MS04	General Smooth Mosaic B-F,C,2	MS20	General Smooth Mosaic L-2,c,g
MS05	General Smooth Mosaic A-f,11	MS21	General Smooth Mosaic A-11,g
MS06	General Smooth Mosaic A-1,g	MS22	General Smooth Mosaic A-f,3
MS07	General Smooth Mosaic L-1,g,10	MS23	General Smooth Mosaic R-3,f,12
MS08	General Smooth Mosaic B-f,12	MS24	General Smooth Mosaic B-10,g
MS09	General Smooth Mosaic A-f,11,g	MS25	General Smooth Mosaic B-f,2,g
MS10	General Smooth Mosaic L-3,f,12	MS26	General Smooth Mosaic R-1,g,10
MS11	General Smooth Mosaic A-f,2,g	MS27	General Smooth Mosaic B-f,11,g
MS12	General Smooth Mosaic L-1,c,10 & R-3,c,12	MS28	General Smooth Mosaic A-1,c,3 & B-10,c,12
MS16	General Smooth Mosaic B-f,g		
MS17	General Smooth Mosaic B-c,g		

I.10 *Repertoire 10 – Drawing characters*

A number of drawing characters, consisting primarily of line drawing characters are available in the various terminal data syntaxes. These consist of the drawing characters (DG01 to DG04, DG13 to 24, and DG32) which are common between DS I and DS II of CCITT Recommendation T.101, drawing characters (DG35 to DG50) which are unique to DS 1, drawing characters (DG05 to DG12, and DG25 to 31) which are unique to DS II and drawing characters (DG33 and DG34) which are unique to DS III.

Note – The notation used to describe these characters identifies the character sub-cell vertices and intermediate points in a similar manner to the way smooth mosaics are identified in Repertoire 7; however, only lines are used here to connect the vertex and intermediate points. Some of the drawing graphics contain more than one line component. This is identified by the description of two line elements separated by "&". In addition, the line weight (or width) may vary on some drawing characters. Heavy weight (wider) line elements are indicated by (W). Special drawing graphics are described in words.



Character sub-cell vertices and intermediate points

Name Code	*Descriptive name*
DG01	Drawing Character W-f,g & c,2
DG02	Drawing Character W-f,g & c,11
DG03	Drawing Character 2,11 & W-c,g
DG04	Drawing Character 2,11 & W-f,c
DG05	Drawing Character 2,f,11
DG06	Drawing Character 2,g,11
DG07	Drawing Character f,11,g
DG08	Drawing Character f,2,g
DG09	Drawing Character f,2
DG10	Drawing Character g,2
DG11	Drawing Character f,11
DG12	Drawing Character g,11
DG13	Drawing Character W-f,g & 2,11
DG14	Drawing Character 2,11
DG15	Drawing Character f,g
DG16	Drawing Character 11,c,g
DG17	Drawing Character f,c,11
DG18	Drawing Character 2,c,g
DG19	Drawing Character f,c,2
DG20	Drawing Character 2,11 & c,g
DG21	Drawing Character 2,11 & f,c
DG22	Drawing Character f,g & c, 11
DG23	Drawing Character f,g & c,2
DG24	Drawing Character f,g & 2,11
DG25	Drawing Character f,g (with arrow head on right)
DG26	Drawing Character f,g (with arrow head on left)
DG27	Drawing Character 2,11 (with arrow head on top)
DG28	Drawing Character 2,11 (with arrow head on bottom)
DG29	Drawing Character (centre small dot)
DG30	Drawing Character (centre large dot)
DG31	Drawing Character (centre large hollow dot)
DG32	Drawing Character (speckled character)
DG33	Drawing Character 3,10
DG34	Drawing Character 1,12
DG35	Drawing Character W-2,11
DG36	Drawing Character W-f,g
DG37	Drawing Character W-11,c,g
DG38	Drawing Character W-f,c,11
DG39	Drawing Character W-2,c,g
DG40	Drawing Character W-f,c,2
DG41	Drawing Character W-2,11 & W-c,g
DG42	Drawing Character W-2,11 & W-f,c
DG43	Drawing Character W-f,g & W-c,11
DG44	Drawing Character W-f,g & W-c,2
DG45	Drawing Character W-f,g & W-2,11
DG46	Drawing Character W-2,11 & c,g
DG47	Drawing Character W-2,11 & f,c
DG48	Drawing Character f,g & W-c,11
DG49	Drawing Character f,g & W-c,2
DG50	Drawing Character W-2, 11 & f,g

Note – The name codes DG33 to DG50 are introduced here since name codes for these characters are not included in either the ISO registry or in CCITT Recommendation T.101.

I.11 *Repertoire 11 – Special mosaics*

A number of special mosaic shapes exist in the various data syntaxes. These special mosaic shapes are generally unique and occur in only one of the terminal data syntaxes. The presentation of these characters must be converted so that their presentation effect may be achieved in another terminal data syntax.

Name Code	*Descriptive name*	*Name Code*	*Descriptive name*
MS02	Centre-Small-Square	MS29	Open-Right-Half-Oval
MS13	Open-Left-Half-Oval	MS30	Filled-Right-Half-Oval
MS14	Filled-Left-Half-Oval	MS31	Reverse-Right-Half-Oval
MS15	Reverse-Left-Half-Oval		

APPENDIX II

(to Recommendation T.101)

Default states of the interworking data syntax

The Interworking Data Syntax permits a choice of the default states to be established. These defaults correspond to the default states of the terminal oriented data syntaxes corresponding to the source of the data. Referencing Tables II-1/T.101 to II-3/T.101 provides a convenient way of establishing the states required for interworking with a particular source of data. Only three default tables are described below. If a particular profile of a data syntax does not correspond exactly to the default table given below, then explicit state vector commands may be used to alter the particular states which are different.

The default tables include the default values for the function and parameters of the data syntax as well as the default boundary conditions. The Terminal Model Precedence indicator is always assumed to be initially "1", and of course at the initial point, or at a reset, all global variables are assumed to be altered.

Note – In Tables II-2/T.101 below a question mark "(?)" implies that the information for that entry in the table needs to be confirmed. A question mark alone indicates that the information needs to be supplied in a future version of the document.

TABLE II-1/T.101

Default-set-I: defaults corresponding to Data Syntax I

Default boundary condition values	
– Screen-Dimensions	– 0.969, 0.797
– Colour-Map-Limit	– 16
– Presentation-Sub-Area	– full screen
– Char-Mode-Constraints	– 15.5, 8
– Coordinate-Limit-Polygon	– 256 (fill form)
– Coordinate-Limit-Spline	– not applicable
– Presentation-Resolution	– 248, 204 (basic rank)
– Macro-Seg-Memory-Limit	– 3072 bytes (min)
– DRCS-Memory-Limit	– 416 bytes (min)
Default presentation parameters	
current-text-position	– upper left corner
current-foreground-colour	– white
current-auxilary-colour	– blue reduced intensity
lining-state	– off
flash-blink-state	– off
basic-char-size-state	– char = normal 16/256, 24/256
colouring-block-state	– 4/256, 4/256
conceal-state	– off
char-invert-box-state	– off
char-marking-state	– not applicable
screen-protection-state	– protect
display-control-state	– scroll = off
device-control-state	– not applicable
cursor-control-state	– not applicable
current-geometric-position	– lower left corner
geometric-control-1-state	– line texture = solid, texture pattern = solid highlight = off logical pel = 0,0
geometric-control-2-state	– not applicable
wait-state	– none
general-text-state	– not applicable
p-text-state	– char rotation = 0° char path = right
char spacing = 1	
	cursor = underscore interrow = single space
geometric-text-state	– not applicable
DRCS-definition-state	– none
macro-definition-state	– none
texture-pattern-state	– not applicable
music-part-memory-state	– musical tone = fixed musical control = fixed part memory = none
animation-configuration-state	– no animation frames frame order = normal
workstation-configuration-state	– not applicable

TABLE II-1/T.101 *(cont.)*

Default colour map – Data Syntax I					
	Map address		Colour value		
			R	G	B
black	0	0000	0001	0001	0001
red	1	0001	1111	0000	0000
green	2	0010	0000	1111	0000
yellow	3	0011	1111	1111	0000
blue	4	0100	0000	0000	1111
magenta	5	0101	1111	0000	1111
cyan	6	0110	0000	1111	1111
white	7	0111	1111	1111	1111
transparent	8	1000	0000	0000	0000
RI red	9	1001	0111	0000	0000
RI green	10	1010	0000	0111	0000
RI yellow	11	1011	0111	0111	0000
RI blue	12	1100	0000	0000	0111
RI magenta	13	1101	0111	0000	0111
RI cyan	14	1110	0000	0111	0111
grey	15	1111	0111	0111	0111

TABLE II-2/T.101

Default-set-II: defaults corresponding to Data Syntax II

Default boundary condition values

– Screen-Dimensions	– 1, 1
– Colour-Map-Limit	– 32
– Presentation-Sub-Area	– not applicable
– Char-Mode-Constraints	– 40, 24
– Coordinate-Limit-Polygon	– 128
– Coordinate-Limit-Polycurve	– not applicable
– Presentation-Resolution	– (?)
– Macro-Seg-Memory-Limit	– (?)
– DRCS-Memory-Limit	– 2048 bytes
– Direct Colours	– 8 colours

Default presentation parameters

current-text-position	– upper left corner
current-foreground-colour	– white
current-auxiliary-colour	– transparent
lining-state	– off
flash-blink-state	– off
basic-char-size-state	– 1/40, 5/128, 40 char/24 rows
conceal-state	– off
char-invert-box-state	– off
char-marking-state	– off
screen-protection-state	– not protected
display-control-state	– Implicit scrolling activated. No defined scrolling area.
device-control-state	– display device = on auxiliary device = off recording device = stop hard copy device = stop
cursor-control-state	– off
geometric-control-1-state	– not applicable
geometric-control-2-state	
– Clipping-Rectangle	0.0, 0.0 and 1.0, 1.0
– deferral-mode	asap
– Fill-Area-Colour-Index	1
– Fill-Area-Interior-Style	hollow
– Fill-Area-Style-Index	1
– Set-Highlighting	off
– implicit-regeneration	allowed
– Line-Type	solid
– Line-Width-Scale-Factor	1.0
– Marker-Size-Scale-Factor	1.0
– Marker-Type	asterisk
– Pattern-Vectors	0.0, 1.0 and 1.0, 0.0
– Polyline-Colour-Index	1
– Polymarker-Colour-Index	1
– regeneration-flag	postpone
– Ws-Viewport	the largest square which fits into the display area
– Ws-Window	0.0, 0.0 and 1.0, 1.0
– workstation-identifier	0
geometric-text-state	
– Char-Expansion-Factor	1.0
– Char-Spacing(continous)	0.0
– Char-Rotation(continous)	0.0, 7.0/320.0 and 7.0/320.0, 0.0
– Text-Alignment	normal, normal
– Text-Colour-Index	1

– Text-Precision	string
– Char-Path	char-path-right
Fill-Pattern-Control	
– Pattern-Representation	
delta-x	1
delta-y	1
pattern-cell-data	1
pattern-index	1
DRCS-definition-state	– none defined
macro-definition-state	– not applicable
texture-pattern-state	– ?
music-part-memory-state	– not applicable
animation-configuration-state	– not applicable
Segment-Control-State	
– Segment-Priority	0.0
– Segment-Transform-Matrix	1.0; 0.0; 0.0
– segment-visibility	visible

Default colour map – Data Syntax II

	Map address		Colour value		
			R	G	B
black	0	00000	000000	000000	000000
red	1	00001	111111	000000	000000
green	2	00010	000000	111111	000000
yellow	3	00011	111111	111111	000000
blue	4	00100	000000	000000	111111
magenta	5	00101	111111	000000	111111
cyan	6	00110	000000	111111	111111
white	7	00111	111111	111111	111111
transparent	8	01000	---	---	---
RI red	9	01001	011111	000000	000000
RI green	10	01010	000000	011111	000000
RI yellow	11	01011	011111	011111	000000
RI blue	12	01100	000000	000000	011111
RI magenta	13	01101	011111	000000	011111
RI cyan	14	01110	000000	011111	011111
grey	15	01111	011111	011111	011111
black	16	10000	000000	000000	000000
red	17	10001	111111	000000	000000
green	18	10010	000000	111111	000000
yellow	19	10011	111111	111111	000000
blue	20	10100	000000	000000	111111
magenta	21	10101	111111	000000	111111
cyan	22	10110	000000	111111	111111
white	23	10111	111111	111111	111111
black	24	11000	000000	000000	000000
red	25	11001	111111	000000	000000
green	26	11010	000000	111111	000000
yellow	27	11011	111111	111111	000000
blue	28	11100	000000	000000	111111
magenta	29	11101	111111	000000	111111
cyan	30	11110	000000	111111	111111
white	31	11111	111111	111111	111111

TABLE II-3/T.101

Default-set-III: defaults corresponding to Data Syntax III

Default boundary condition values	
– Screen-Dimensions	– 0.9999, 0.78125
– Colour-Map-Limit	– 16
– Presentation-Sub-Area	– full screen
– Char-Mode-Constraints	– 40, 20
– Coordinate-Limit-Polygon	– 256
– Coordinate-Limit-Polycurve	– 256
– Presentation-Resolution	– 256, 200 (nominal)
– Macro-Seg-Memory-Limit	– 3072 bytes
– DRCS-Memory-Limit	– 3072 bytes (shared with Macro)
Default presentation parameters	
current-text-position	– lower left corner
current-foreground-colour	– colour = white mode = direct
current-auxilary-colour	– none
lining-state	– off
flash-blink-state	– off
basic-char-size-state	– dx = 1/40, dy = 1/128
conceal-state	– not applicable
char-invert-box-state	– char = normal size, invert = off, box (transparent) = off
char-marking-state	– not applicable
screen-protection-state	– protected
display-control-state	– scroll = off
device-control-state	– not applicable
cursor-control-state	– off (invisible)
geometric-control-1-state	– line texture = solid texture pattern = solid texture mask = 1/40, 5/128 highlight = off logical pel = 0,0
geometric-control-2-state	– not applicable
wait-state	– none
general-text-state	– char rotation = 0° char path = right
char spacing = 1	
	cursor = underscore interrow = single space
p-text-state	– not applicable
geometric-text-state	– not applicable
DRCS-definition-state	– none defined
macro-definition-state	– none defined
texture-pattern-state	– none defined
music-part-memory-state	– not applicable
animation-configuration-state	– not applicable
workstation-configuration-state	– not applicable

TABLE II-3/T.101 *(cont.)*

Default colour map – Data Syntax III					
	Map address		Colour value		
			R	G	B
nominal black	0	0000	000	000	000
	1	0001	001	001	001
	2	0010	010	010	010
grey	3	0011	011	011	011
	4	0100	100	100	100
	5	0101	101	101	101
	6	0110	110	110	110
nominal white	7	0111	111	111	111
	8	1000	000	000	111
	9	1001	000	101	111
	10	1010	000	111	100
hues	11	1011	010	111	000
	12	1100	111	111	000
	13	1101	111	010	000
	14	1110	111	000	100
	15	1111	101	000	111

Recommendation T.150

TELEWRITING TERMINAL EQUIPMENT

TABLE OF CONTENTS

This Recommendation consists of four parts, combined in one document

1 Scope

This Recommendation specifies technique-oriented characteristics of telewriting and the application of telewriting in combination with voice communication. Service-oriented requirements are defined in Recommendation F.730. In the development of this Recommendation, compatibility with other telematic services is taken into account. This Recommendation is structured in four parts:

Part 1 – Fundamental characteristics

Part 2 – Telewriting together with telephony

Part 3 – Zone coding

Part 4 – Differential chain coding.

Part 1 – Fundamental characteristics

1 Introduction

1.1 Telewriting is a communication technique that enables the exchange of handwritten information through telecommunication means. The handwritten information may consist of text in handwriting, drawings, diagrams, etc.

1.2 By means of telewriting terminal equipment, the TRACE of the writing instrument as produced at the sending side, is reproduced at the receiving side including the effect of movement.

1.3 In the sending part of the terminal the handwritten input information is converted into a digital signal: the coded representation of the handwritten information. Next, this digital signal is converted into a signal suitable for transmission.

1.4 In the receiving part of the terminal the received signal is converted into a digital signal, corresponding with the coded representation as described in above. From this digital signal, the handwritten information is reproduced.

1.5 The reproduction of the handwritten information can take place on a screen, on paper or both. In this Recommendation, the characteristics of communication through telewriting are defined with respect to the image on a screen (soft copy). Reproduction on paper (hard copy) is considered to be an optional function under local control.

1.6 Storage may take place between the writing (the input process) and the reproduction (the output process). When retrieved from a store, the message will appear on the receiver's screen in the same way as in the case of a direct connection.

1.7 A page of handwritten information (or part of it) could be reproduced as a still picture. This application, however, is not covered in the present text.

1.8 Telewriting can be used in various ways:

- as independent communication technique,
- in combination with voice communication through a telephone network,
- in the context of teleconferencing,
- in the context of information retrieval.

2 Definitions

2.1 telewriting image

A collection of telewriting presentation elements, to be displayed together.

Note – The telewriting image can exist in visible form at the output device, or in the form of a coded representation.

2.2 presentation element

Basic graphic element used to construct an image.

Examples of telewriting presentation elements are: trace, closed area, background.

2.3 **coding rectangle**

Rectangular area representing the coding space in horizontal and vertical direction, available for coding of a telewriting image.

2.4 **image area**

(previously: text area)

Rectangular part of the display area, to be considered as the image of the coding rectangle.

2.5 **background**

Presentation element being a rectangular area with the same size as the image area, acting as a reference area on which telewriting foreground information can be presented.

2.6 **trace**

Presentation element being a curve of an arbitrary shape, starting from a defined position, being completed incrementally and ending at a defined position.

2.7 **closed area**

Presentation element being an area enclosed within one trace which constitutes a closed line.

2.8 **marker**

Marked representation of a single position in a telewriting image.

Note – A marker is not a permanent part of a telewriting image, but exists only as long as it is activated.

2.9 **attribute**

A particular property which applies to a presentation element or to a group of presentation elements.

Examples: line thickness, colour.

3 References

In the text of this Recommendation the following Recommendations/standards are referred to:

- Rec. F.730: Service oriented requirements for telewriting applications.
- Rec. T.101: International interworking for videotex services; Annex C, data syntax II.
- Rec. V.21: 300 bits per second duplex modem standardized for use in the general switched telephone network.
- ISO 9281: Information processing – Identification of picture coding methods.

4 Presentation functionalities

4.1 This section describes a set of presentation functionalities. This set of functionalities is intended as a repertoire of presentation functionalities for telewriting in general. For a specific application a subset may be defined.

4.2 In the description of presentation functionalities, the concept of TRACE is being used. A trace is a curve of an arbitrary shape, starting from a defined position, being completed incrementally and ending at a defined position. Handwritten information is considered to consist of traces.

4.3 Representation of the handwritten information is accomplished by the sequential reconstruction of the individual traces. This implies that the effect of movement is retained during each reproduction.

4.4 Telewriting information is to be displayed on the display area of some output device. The display area is considered to be a two-dimensional surface.

4.5 The display area is subdivided into an image area and a border area; see Figure 1-1/T.150.

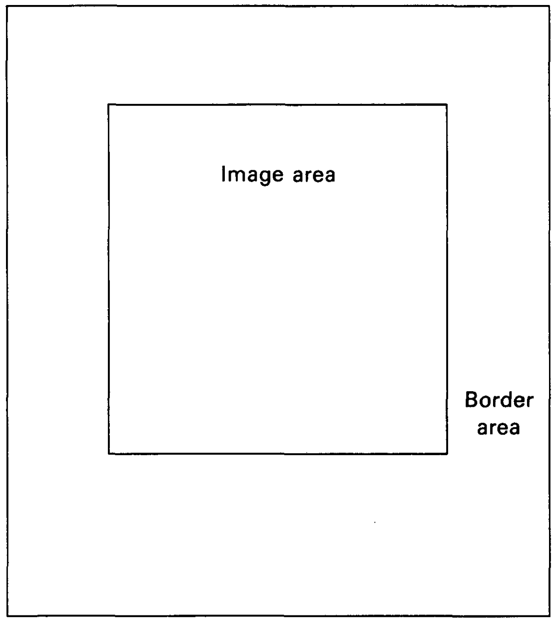


FIGURE 1-1/T.150

Subdivision of display area

4.6 The border area surrounds the image area. External form and dimensions of the border area are not specified. The presence of a border area is not mandatory. It is however inevitable in certain implementations.

4.7 The image area is rectangular. The two shorter edges of the image area have a vertical orientation, the two longer edges have a horizontal orientation. The length ratio of shorter and longer edges is 3:4.

4.8 The position of telewriting information on the display area is defined with respect to the edges of the image area.

4.9 Information on the display area is composed of presentation elements of three categories:

- foreground,
- background,
- border area.

4.10 Foreground and background presentation elements are defined in the image area only.

Border area presentation elements are defined in the border area only. The use of the border area is not defined for telewriting.

4.11 Foreground presentation elements include trace, marker and closed area.

4.12 The presentation elements have the following characteristics:

- *Trace*: This is the curve as defined in § 2.6 of this part; the essence of the handwritten information is represented by one trace or by any combination of traces; the image area can contain an undefined number of traces at a time.
- *Marker*: This is a marked representation of a single position; it behaves as if it is overlaid on the foreground; a moving marker does not create a trace; a marker can be switched on and off; one user can generate only one marker at a time. The image area can contain one locally generated marker and one remotely generated marker.
- *Closed area*: This is the area that is enclosed within a closed trace; this closed trace is the perimeter. A trace is a closed trace if it intersects itself; a trace that is nearly closed can be converted into a closed trace, by the addition of the lacking part of the trace.
- *Background*: The background is a defined reference area on which foreground information is to be imaged; if the full image area is filled with foreground information, the background is not visible.

– *Border area* : The border area is independent of the information in the image area.

In case of a CRT display the border area is the remaining part between image area and edges of the display area.

In case of a cell-structured display device, the image area may coincide exactly with the display area. In that case no border area remains.

4.13 The various presentation elements can have attributes assigned to them as defined in Table 1-1/T.150.

TABLE 1-1/T.150

Attributes of telewriting presentation elements

Presentation element	Attributes
Trace	Line thickness, line texture, colour
Marker	Shape, size, colour
Closed area	Area texture, colour (interaction or area attributes with background attributes to be defined)
Background	Area texture, colour
Border area	Not defined

Note – The concept of colour includes "intensity".

4.14 Once an image is displayed, subsequent modification of attributes is restricted as follows:

– trace: attributes unchangeable;

– marker: attributes can be changed at any instant;

– closed area: attributes unchangeable;

– background: attributes can be changed at any instant.

4.15 In case of intersection of two traces, the image of the older trace is interrupted as far as it coincides with the newer trace.

4.16 In case of intersection of a trace and a marker, the image of the trace is interrupted as far as it coincides with the marker. After removal of the marker, the image of the original trace is restored.

4.17 With respect to erasure of foreground information, a distinction is made regarding the area in which erasure takes place:

– full image area;

– defined part of the image area;

– individual traces.

4.18 *Erasure of the full image area*

All foreground information in the image area is removed; the background assumes a pre-defined appearance.

4.19 *Erasure of a defined part of the image area*

An area is identified either by means of a closed trace or as a defined square, within which all foreground information is to be removed including the perimeter itself.

4.20 *Erasure of individual traces*

An existing trace is covered by a thicker trace with the same attributes as the background: this type of erasure is processed in the same way as a trace.

4.21 Any modification of background information can take place for the full image area only.

5 Principles of telewriting coding

5.1 Telewriting coding relates to coding of telewriting information in foreground and background and to erasure functions.

5.2 This section contains principles of telewriting coding. In Parts 3 and 4, details of telewriting coding are defined for two methods, namely zone coding and differential chain coding, respectively.

5.3 The coding is defined at the "telewriting coding interface", TCI. This interface is introduced for ease of reference, but need not exist physically.

5.4 In the sending part of the telewriting terminal, the signal at the TCI contains all data originating from handwritten input, selection of attributes and use of erasure functions.

5.5 The signals at the TCI, both in sending and receiving parts, do not contain data pertaining to transmission or communication functions.

5.6 In the receiving part of the telewriting terminal, the signal at the TCI contains all data required to image the information in accordance with the intentions of the originator.

5.7 The concept of the TCI is illustrated in Figure 1-2/T.150.

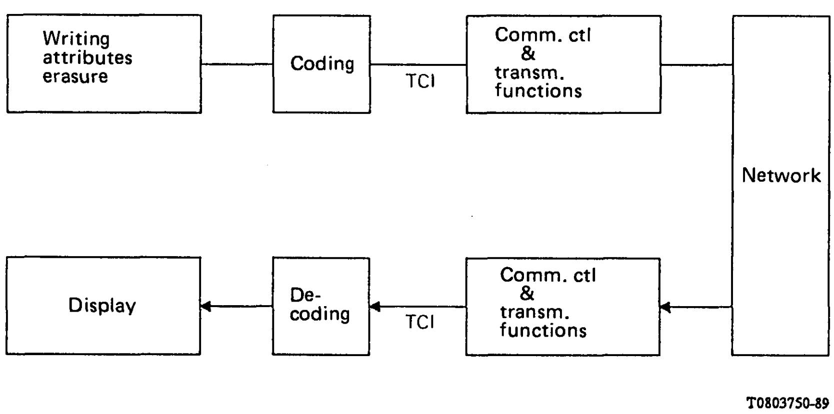


FIGURE 1-2/T.150

Telewriting coding interface, TCI

5.8 The signal at the TCI includes x and y coordinate information regarding telewriting presentation elements.

5.9 The x and y coordinates are related to a unit area of 1 × 1. This implies that the respective values of x and y always lie between 0 and 1 (0 included, 1 not included).

5.10 The origin of the coordinate system is in the lower left corner. The x-axis is horizontal, the y-axis is vertical.

5.11 The horizontal size of the telewriting image area corresponds with x = 1, the vertical size of this image area corresponds with y = 0.75. See Figure 1-3/T.150.

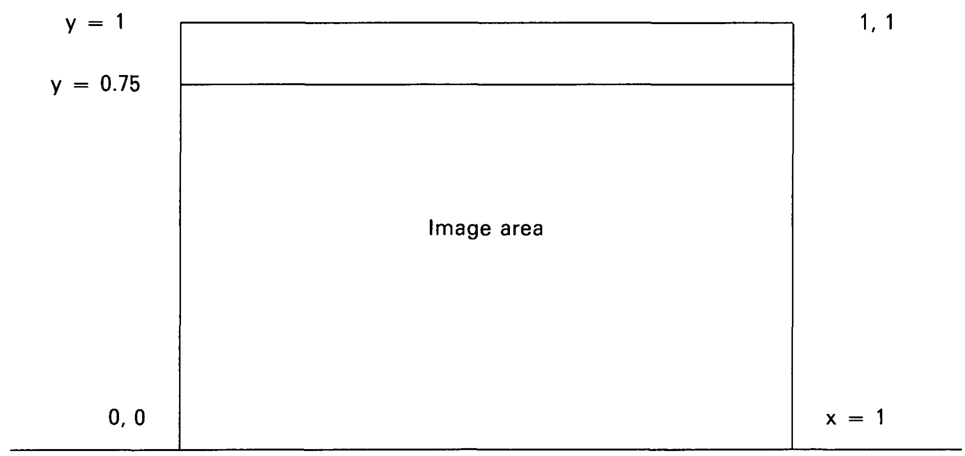


FIGURE 1-3/T.150

Position of image area within unit area

5.12 All coordinates of the telewriting information are quantized relative to a measurement grid in the unit area. The resolution of this grid determines the accuracy.

5.13 The default resolution is 512 × 512 grid units. The telewriting coding can optionally also accommodate grid resolutions of 1024 × 1024 and 2048 × 2048 grid units.

Part 2 – Telewriting together with telephony

1 General

1.1 This part of the Recommendation defines the use of telewriting in combination with voice communication through a telephone network (PSTN).

1.2 For this application, both sides of the connection must have a combined telephone and telewriting terminal.

1.3 The combined telephone and telewriting terminal should, as long as the telewriting transmission function is switched off, behave like a normal telephone set, both for incoming and outgoing calls. In this situation, the full bandwidth is available for transmission of speech signals.

1.4 During a telephone conversation, the telewriting transmission function at either side of the connection, may be switched on and off, manually or automatically.

1.5 Remark that in this part of the Recommendation "switching on and off" of the telewriting function refers to the telewriting transmission functions. Regardless of this, the telewriting equipment may be used locally, whether or not a telephone connection exists.

1.6 By means of the telewriting terminal, the user can generate information. This includes: creation of traces, marker switching on and off, movement of the marker, use of erasure functions.

1.7 In this part, distinction is made between "basic terminal" and "enhanced terminal".

1.8 The enhanced terminal is not defined yet, but compared to the basic terminal it is anticipated to have additional capabilities regarding unattended operation, transmission facilities and presentation functionalities.

2 Main characteristics of the basic terminal

2.1 In this section, a basic terminal is defined.

In the basic terminal a set of functions is implemented that is to be considered as a minimum requirement; thus a basic level of compatibility is defined.

2.2 A basic terminal includes a telephone apparatus, a writing device and a display device. Circuitry to implement control functions may be accommodated in a separate unit or may be included in one of the devices mentioned.

2.3 Information, generated at either side of the connection will be reproduced on the display devices at both sides of the connection.

2.4 Both sides of the connection can contribute, one after another, to the same image.

2.5 In the basic terminal, transmission of telewriting signals is accomplished through a sub-channel, segregated from the speech channel. Transmission of speech signals and telewriting signals can take place simultaneously.

2.6 Half-duplex transmission is used for conveying the telewriting signals through the sub-channel, i.e. the transmitter is prevented from sending as long as the associated receiver receives telewriting signals from the other side.

2.7 The total power level of speech plus telewriting signals should conform to the limits normally applicable to speech transmission and data transmission.

2.8 The basic terminal can assume three modes of operation. The characteristics pertinent to each mode, are described in Table 2-1/T.150.

TABLE 2-1/T.150

Modes of operation of the basic terminal

Speech only	The telewriting function remains in the OFF condition.
Speech plus telewriting	The telewriting function can be switched ON after the establishment of a connection. Speech signals and telewriting signals can be sent simultaneously.
Telewriting only	This mode can be switched ON after the establishment of a connection. The sending of speech signals is blocked, the power level of the telewriting signals is increased correspondingly. Reception of speech signals is still possible.

2.9 In this Recommendation, the expression "telewriting ON" is used as a common indication for either "speech plus telewriting" or "telewriting only".

2.10 A basic terminal may be able to continue transmission and reception of telewriting signals after termination of the human conversation. In this case, the telewriting transmission function will be switched OFF automatically after completion of the telewriting transmission. (Defined in more detail later on.)

2.11 For the coding of telewriting information, two methods are recognized for use at the sending side: tone coding (defined in Part 3) and differential chain coding (defined in Part 4).

At the receiving side, the basic terminal should be able to properly accept telewriting signals coded according to either method.

3 Presentation functionalities of the basic terminal

3.1 The general description of presentation functionalities, as given in Part 1, § 4, applies.

With respect to this general description certain restrictions apply, as defined in the following points.

3.2 The presentation functionalities as described for the basic terminal are to be regarded as default capabilities.

If required, characteristics of terminals with a higher level of sophistication will be described in a section on enhanced terminal.

3.3 The basic terminal employs a monochrome display device. The writing device generates coded representations of monochrome images only.

3.4 The attributes applying to the basic terminal are given in Table 2-2/T.150.

TABLE 2-2/T.150

Attributes applying to the basic terminal

Presentation elements	Attributes
Image size	Horizontal: 512 GU Vertical: 0.75 × 512 GU Options, the receiver must be able to accept: Horizontal: 1024 and 2048 GU Vertical: 0.75 × 1024 and 0.75 × 2048 GU.
Trace	
– thickness	Unit thickness, as used in the output device. Options: 2 × and 3 × unit thickness.
– texture	Solid, no options.
– colour	Monochrome, as used in the output device. The receiver must be able to accept the codes of traces with colours: red, green, blue, yellow, magenta, cyan, white, black. A black trace has the same colour as the background (used for erasure).
Closed area	
– texture	Solid.
– colour	Same as background colour (used only for partial erasure). The receiver must be able to accept the codes of closed areas with colours: red, green, blue, yellow, magenta, cyan, white, black.
Background	
– texture/colour	No information about the background is transmitted. Background can only be imagined as dark screen. This corresponds with colour black.
Border area	Border area is not specified, no information about the border area is transmitted.
Marker	
– shape	PLUS sign; other shapes may be possible depending on terminal implementation.
– size	Not specified.
– colour	Marker colour is not transmitted; on a monochrome device the marker appears in foreground colour; on a colour device the marker may assume a colour under local control.
Full erasure	Black background is restored.
Partial erasure	1) closed area; 2) overwriting with thicker black trace.

GU Grid units

4 Transmission for the basic terminal

4.1 Transmission of the modulated telewriting signal takes place in a small frequency band, segregated from the speech channel. This band is referred to as the sub-channel.

4.2 The centre of the sub-channel is located at 1750 Hz. Details of the implementation are not given here, but the requirements of §§ 4.6 and 4.7 should be met.

4.3 The binary telewriting signal is converted into a signal suitable for transmission, by means of frequency shift modulation. Details are the same as those specified in Recommendation V.21 for channel 2 (the high channel).

4.4 The modulation rate is 300 Bd, the bit rate is 300 bit/s.

4.5 The V.21 requirements for channel 2 are summarized as follows: The nominal mean frequency of the transmission signal is 1750 Hz. The frequency deviation is + or − 100 Hz. Consequently, the nominal characteristic frequencies are 1850 Hz and 1650 Hz respectively. The higher frequency corresponds to a binary 0.

4.6 The amount of speech signal power that can reach the local and remote telewriting receivers, should be sufficiently low to avoid errors in the demodulated telewriting signal.

4.7 The amount of telewriting signal power that can reach the local and remote telephone receivers (i.e. the loudspeaker part) should be sufficiently low to avoid disturbance of the conversation.

4.8 In the mode of operation "telewriting only", the output power of the telewriting transmitter shall be in accordance with the requirements described in Recommendation V.21.

4.9 In the mode of operation "speech plus telewriting", the modulated Telewriting signal should be attenuated by 4 dB with regard to the level determined by § 4.8. If experience shows that also the power of the speech signal should be adapted, relevant requirements will be included in the next issue of this Recommendation.

4.10 In the case of long-distance communication an echo suppressor may be present in the link. This will hamper the "speech plus telewriting" mode. Since, generally, disabling of the echo suppressor cannot be guaranteed to solve the problem, it is recommended to use the "telewriting only" mode, alternating with the "speech only" mode.

4.11 The telewriting data as well as communication control commands are structured in 8-bit bytes.

For transmission, each byte is packed in an 11-bit transmission word as defined below.

4.12 The structure of each transmission word is as follows:

1startbit, binary value ZERO

8 bits representing telewriting or control data

1 parity bit

stopbit, binary value ONE.

This structure is illustrated in Figure 2-0/T.150.

Start	Data	Parity	Stop
1 bit	8 bits	1 bit	1 bit

FIGURE 2-0/T.150

Structure of a transmission word

4.13 For the value of the parity bit, EVEN parity applies. This Recommendation does not specify any action for the basic terminal in case of reception of an erroneous parity bit.

4.14 The transmission words are conveyed in start-stop mode, i.e. the pause following a transmission word until the occurrence of the next transmission word, may in principle have any duration. However, the bits constituting the transmission word should be transmitted as a contiguous sequence at the appropriate bit rate.

4.15 In addition to its task of transporting bits, the data send signal may assume one of three possible states:

– MARK signal: a binary ONE condition, with a duration significantly longer than a bit period.

– SPACE signal: a binary ZERO condition; this condition is not used in the framework of this Recommendation.

– Carrier OFF: no send signal present.

5 Transmission blocks

5.1 To define the transmission structure, the concept of transmission block is introduced. In the general case, a transmission block contains transmission words and MARK signals. However, also transmission blocks containing MARK signals only may occur.

5.2 The beginning of a transmission block is identified by the occurrence of one out of two defined combinations of MARK signal and carrier OFF condition, referred to as start combination No. 1 and start combination No. 2.

5.3 The start combinations are defined as follows:

– start combination No. 1 carrier OFF during at least 130 ms, followed by MARK signal of 100 ± 20 ms followed by carrier OFF during 100 ± 20 ms followed by MARK signal of 200 ± 20 ms.

– start combination No. 2 carrier OFF during at least 130 ms, followed by MARK signal of 400 ± 20 ms.

See illustration in Figure 2-1/T.150.

The use of these start combinations is defined later.

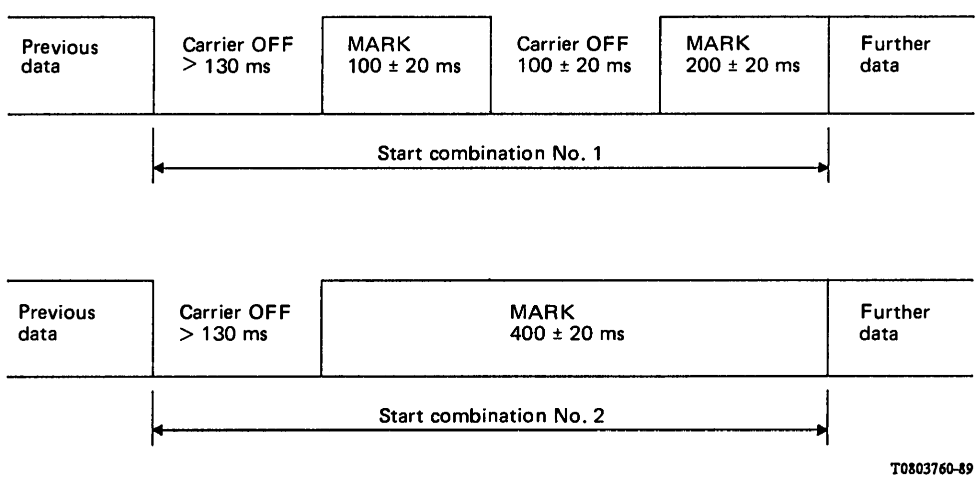


FIGURE 2-1/T.150

Start combinations

5.4 Immediately following the start combination of a transmission block, one of the following signals should be sent:

– a MARK signal

– a single transmission word

– a sequence of transmission words.

Between any two subsequent transmission words, a MARK signal may occur, representing a pause in the writing process.

5.5 Every transmission block is terminated by a MARK signal of 500 ± 20 ms. The MARK signal is to be followed by a carrier OFF condition of at least 130 ms.

5.6 The MARK signals representing pauses may have various durations, determined as follows:

- during PEN DOWN and absence of other telewriting activity, the MARK signal may continue without limitation;
- after PEN UP the terminal will apply a limit of 500 ± 20 ms. Within this limit the telewriting activity may continue without procedural steps. If the limit expires, the carrier will be switched OFF. Thus the transmission block is automatically terminated by the terminal. Sending of further data requires the start of a new transmission block.

5.7 The periods between transmission blocks are indicated by carrier OFF conditions.

5.8 The formats of transmission blocks are summarized in Figure 2-2/T.150.

Previous data	Carrier OFF	Start combination	Transmission words and MARK signals	MARK	Carrier OFF

T0803770-89

FIGURE 2-2/T.150

Transmission block format, summarized

6 Transmission procedure

6.1 Prior to actually sending telewriting data, the terminal is to decide whether it functions in MASTER mode or in SLAVE mode.

In case of a transmission collision, the master terminal has transmission privilege over slave terminals.

6.2 The terminal decides about the master/slave status by sending the start combination No. 1 and observing the received signal.

6.3 If the terminal, engaged in sending start combination No. 1, detects a received carrier signal at its receiver input (during a carrier OFF interval) it decides to be a slave and it postpones further attempts to send data. See Figure 2-3/T.150.

6.4 If the terminal does not detect a received carrier signal during the sending of the start combination, it decides to be a master and continues sending. See Figure 2-3/T.150.

6.5 In the case that only one terminal generates telewriting data, this terminal assumes the master status. The receiving terminal remains in the slave status.

6.6 As a header for the subsequent transmission blocks, a master terminal uses start combination No. 2, a slave terminal uses start combination No. 1. See Figure 2-4/T.150.

6.7 The master/slave status decision in a given terminal remains valid until it is cancelled as follows:

- A master terminal becomes a slave if it is not engaged in sending at the moment that another terminal sends start combination No. 1.
- A slave terminal becomes a master terminal at the moment that it sends a start combination No. 1 and no receive carrier signals are being detected.
- A master status is cancelled by "telewriting OFF".

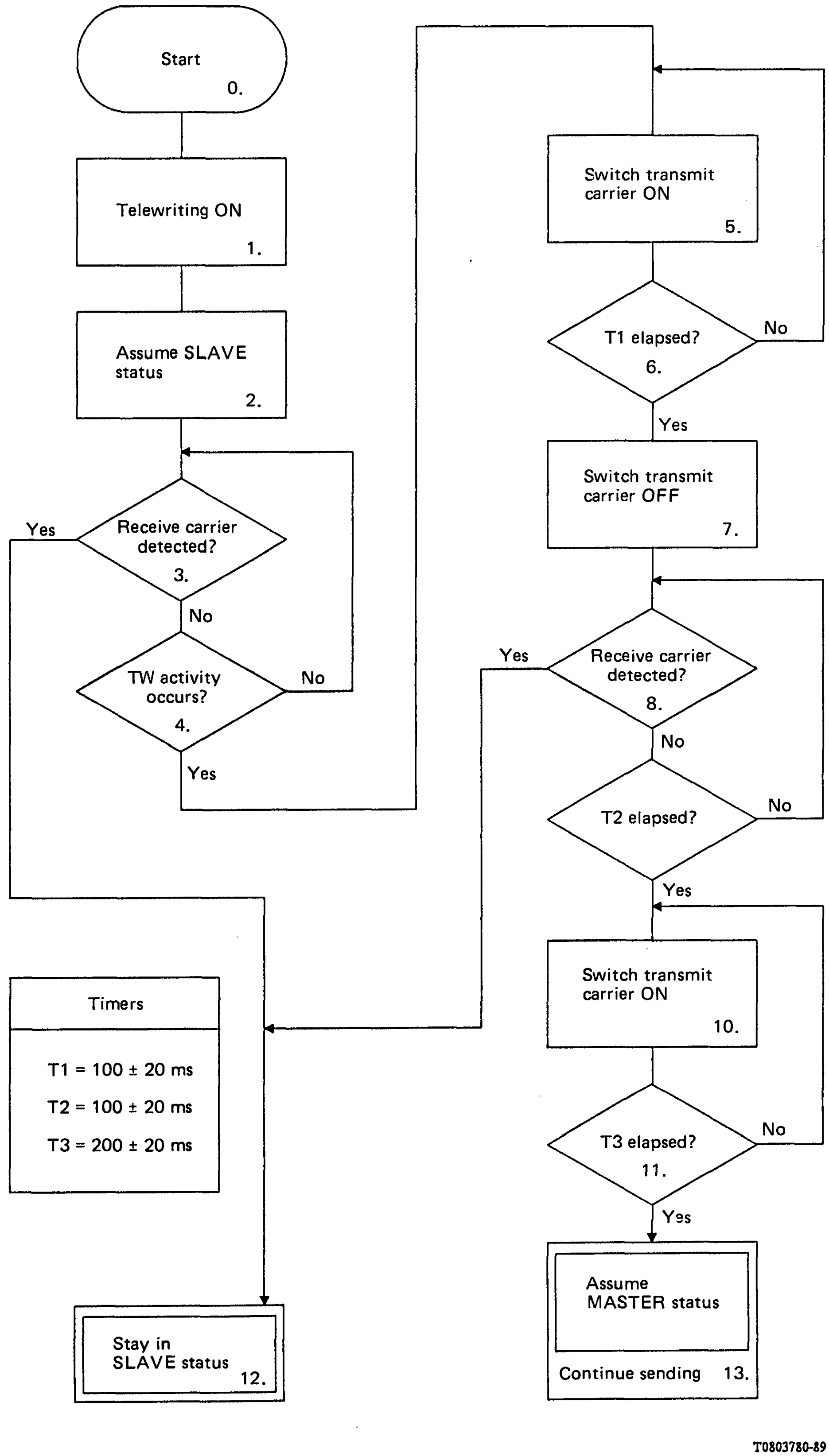


FIGURE 2-3/T.150

MASTER/SLAVE status decision procedure

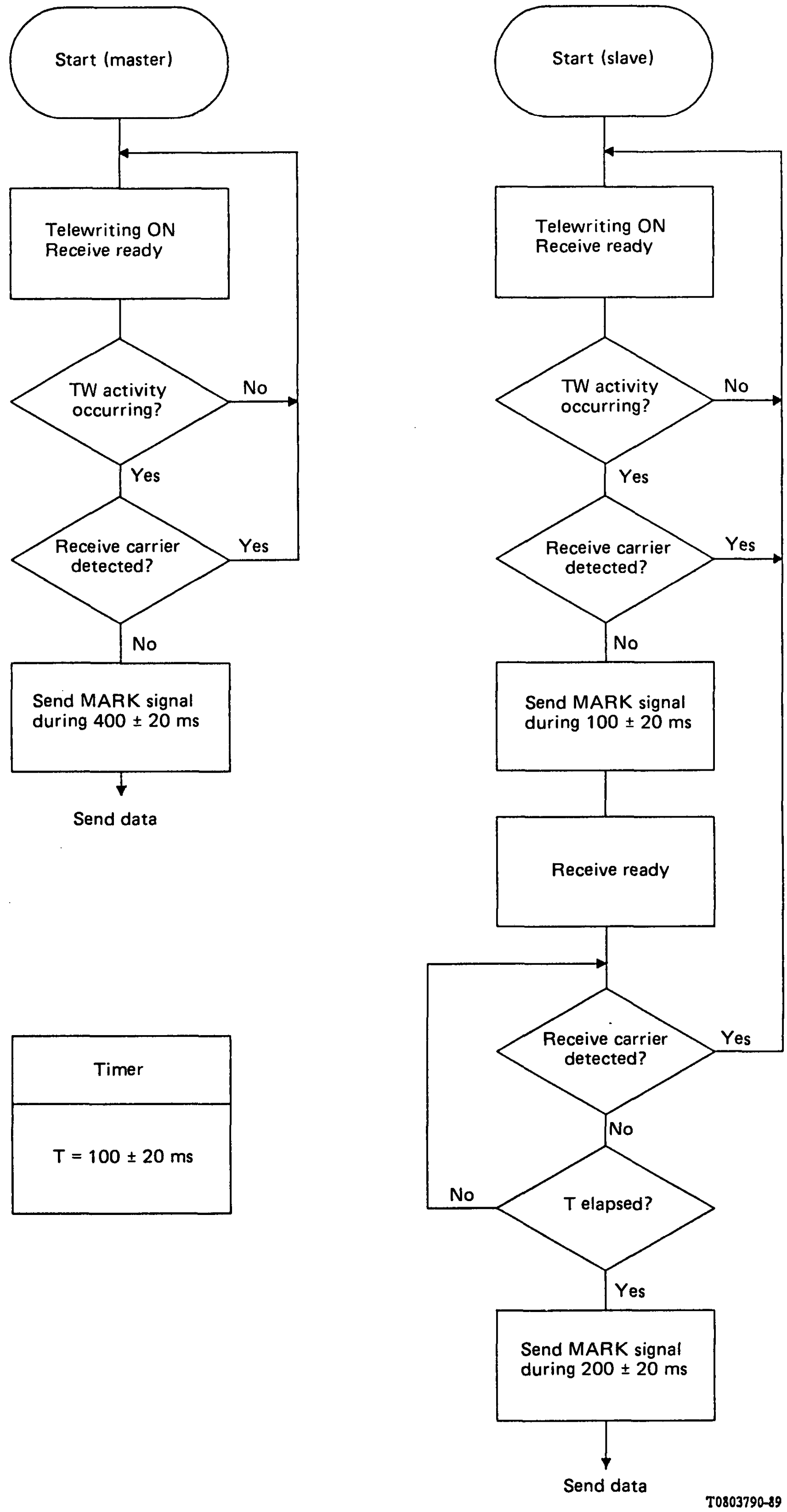


FIGURE 2-4/T.150

Transmission start for master and slave respectively

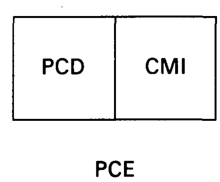
7 Coding identifier

7.1 In the communication control procedures, the existing of two coding methods is recognized, i.e. zone coding and differential chain coding respectively.

The method actually used is identified by the coding identifier PCE. (PCE = picture control entity).

A terminal receiving signals according to either method will be able to activate the appropriate decoding function, by recognizing the coding identifier.

7.2 The coding identifier is structured according to ISO 9281. In this standard, the coding identifier PCE is defined to comprise a picture coding delimiter (PCD) and a coding method identifier (CMI). See Figure 2-5/T.150.



PCE Picture control entity

PCD Picture coding delimiter

CMI Coding method identifier

FIGURE 2-5/T.150

Structure of coding identifier

7.3 (Copy of ISO 9281, § 6.2.4 modified)

The PCD shall announce or delimit the data for a particular picture coding method. The PCD shall comprise the two-byte sequence 01/11, 07/00.

7.4 (Copy of ISO 9281, § 6.2.5)

The CMI shall specify the particular coding method for the picture data that follow it. The CMI may consist of one or more octets corresponding to the bit combinations in the range 02/00 to 07/14 of an 8-bit code table.

7.5 (Copy of ISO 9281, § 6.2.6)

Each CMI identifying a particular picture coding method shall be registered with the ISO Registration Authority for Picture Coding Methods (to be set up).

7.6 The telewriting coding identifier, when included in a transmission block, occupies the first three (or more if appropriate) transmission words following the start combination. See Figure 2-6/T.150.

Carrier OFF	Start combination	Coding identifier	Telewriting data and MARK signals	MARK	Carrier OFF

FIGURE 2-6/T.150

Transmission format, including coding identifier

7.7 In a point-to-point configuration, the inclusion of the coding identifier in the first transmision block only, would in principle be sufficient for the whole session.

However, for multipoint communication, the insertion of the coding identifier in each transmission block is required.

In view of this requirement, it is recommended that the coding identifier be included in each transmission block containing telewriting data, irrespective of the configuration.

7.8 The terminal should be designed such that transmission of the coding identifier takes place automatically at the right moment.

7.9 For telewriting equipment according to this Recommendation T.150 the following bit combinations should be used in the coding identifier. See Table 2-3/T.150.

TABLE 2-3/T.150

Coding identifier bit combinations

Acronym	Bit combination
PCD (2 byte sequence)	01/11, 07/00
CMI Zone coding	02/00, 04/00
CMI Diff-chain coding	02/00, 04/01

Note – The above allocations are of a preliminary nature, pending further development of ISO 9281.

8 Communication control, general requirements

8.1 This section defines requirements for the control of data exchange for the basic telewriting terminal.

8.2 These requirements also apply to data exchange between any enhanced terminal and a basic terminal.

8.3 The requirements permit the use of a two-hop satellite circuit in the connection between two terminals.

8.4 The requirements also permit multi-point communication via a voice bridge.

8.5 Establishment and clearing of the telephone connection take place in accordance with the requirements set by the telephone network.

8.6 For the basic terminal, automatic calling and answering are not defined.

8.7 A basic terminal may, as an option, be equipped such that it can maintain the exchange of telewriting data after termination of the speech conversation. This option is identified as "automatic call termination".

8.8 The automatic call termination implies that the telewriting function (sending as well as receiving) is able to operate autonomously while the telephone apparatus is in the ON HOOK condition.

8.9 To enable automatic call termination, the terminal must be able:

- to note that sending respectively reception of a telewriting transmission block is going on, during the ON HOOK condition of the telephone apparatus,
- to recognize the end of the final telewriting transmission block,
- to switch back to the speech only mode and to clear the telephone connection.

8.10 Switching between the three modes "speech only", "speech plus telewriting" and "telewriting only" can be done manually. In addition, switching the telewriting function OFF can take place automatically by means of the communication control command SSO in the transmission signal. The transitions between modes of operation are illustrated in Figure 2-7/T.150.

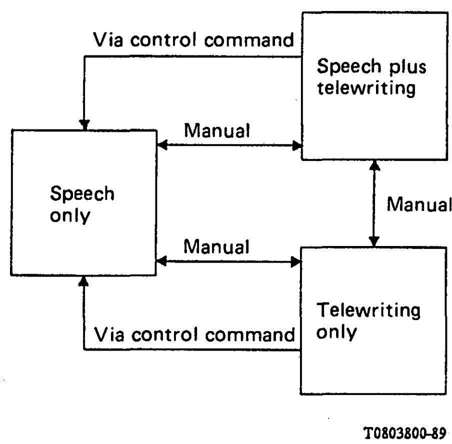


FIGURE 2-7/T.150

Transitions between modes of operation

9 Communication control commands

9.1 For control of the communication process, the commands SSO and HLO are available.

The coding of these commands is as follows:

SSO 1/7

HLO 0/5

The meaning of these commands is described in Table 2-4/T.150.

TABLE 2-4/T.150

Communication control commands

Acronym	Meaning
SSO	Set speech only This command indicates that the terminals are instructed to switch from telewriting ON to the speech only mode
HLO	Hello This command is to be sent by a terminal that expects telewriting data, but does no receive such data

9.2 A terminal will automatically send SSO upon the instruction by its local user, to switch over from the telewriting ON mode to the speech only mode.

Transmission of SSO can take place in two ways:

- At the end of the current transmission block. SSO is attached to the block, according the format defined below.
- By means of a separate transmission block. Such a block is sent specifically for conveying SSO. Format: defined below.

9.3 A terminal receiving SSO will revert to the speech only mode and does not recognize further telewriting signals.

9.4 The format for sending SSO is defined in Figures 2-8/T.150 and 2-9/T.150.

For zone coding

Start combination	PCE (ZC)	Telewriting data (ZC)	1/15	SSO	1/15	MARK	Carrier OFF

For differential chain coding

Start combination	PCE (DCC)	Telewriting data (DCC)	SSO	MARK	Carrier OFF

T0803810-89

FIGURE 2-8/T.150

Transmission block containing telewriting data and SSO

For zone coding

Start combination	PCE (ZC)	1/15	SSO	1/15	MARK	Carrier OFF

For differential chain coding

Start combination	PCE (DCC)	SSO	MARK	Carrier OFF

T0803820-89

FIGURE 2-9/T.150

Transmission block containing SSO only

9.5 The HLO command will only be sent in a block without telewriting data. The format should be as defined in Figure 2-10/T.150.

For zone coding

Start combination	PCE (ZC)	1/15	HLO	1/15	MARK	Carrier OFF

For differential chain coding

Start combination	PCE (DCC)	HLO	MARK	Carrier OFF

T0803830-89

FIGURE 2-10/T.150

Transmission block containing HLO only

9.6 The HLO command is intended for use with automatic reception. This command will be sent by a terminal if it has not received valid telewriting data during a period of 35 seconds since:

- establishment of the telephone call;
- reception of the last valid transmission block.

9.7 The terminal receiving a HLO command responds with a MARK signal of 700 ± 20 ms.

9.8 A terminal in the telewriting ON condition, receiving signals other than valid telewriting data (e.g. a tone from the telephone network) cannot enter the send mode. In this case, the terminal returns to the speech only mode without sending any command or other information (after a guard-time of 35 seconds).

10 Description of the communication process

10.1 In order to describe the full communication process, the concepts of "telewriting activity" and "telewriting session" are introduced. These are defined as follows:

- **Telewriting activity** – Any action by the user that causes the telewriting terminal (in the telewriting ON condition), to send data. Examples of such actions are: pen down, marker ON, erasure.
- **Telewriting session** – A period of time delimited by session start and session end, during which two communicating terminals have a relationship that enables them to exchange telewriting data.

10.2 The event determining session start is:

- the terminals are in the condition telewriting ON,
- at one of the terminals the first telewriting activity has occurred.

10.3 The event determining session end is:

- the terminals switch over to the telewriting OFF condition.

10.4 The session is established as soon as the coding identifier is received and recognized by the receiving terminal.

10.5 At the beginning of the session, both terminals have the slave status. During the session, only one terminal at a time can acquire the master status.

10.6 In the preceding text of this Part 2, all elements to be used in the communication process are defined now.

The process can be summarized as described in Table 2-5/T.150.

10.7 The preceding description is given for a point-to-point configuration. However, taking into account that only one terminal can have the master status, this description is applicable to a multipoint configuration as well. In this case it is indispensible that every transmission block contain a coding identifier.

TABLE 2-5/T.150

Communication process summarized

Step 1	Both parties agree by speech to switch to the telewriting ON condition.
Step 2	Following telewriting ON, each terminal is in the receive ready condition, i.e. the receiver is ON but it does not receive telewriting signals.
Step 3	The first telewriting activity occurring at one of the terminals causes that terminal to initiate the transmission of the first transmission block.
Step 4	The terminal initiating the transmission of the first transmission block assumes the master status.
Step 5	The session is established as soon as the receiving terminal has received and recognized the coding identifier contained in the first transmission block.
Step 6	Within the session, each terminal may alternatingly assume send, receive and receive ready conditions, as required by human actions and/or received signals. When appropriate, the master status will be taken over by an other terminal, as defined in the section on transmission procedures.
Step 7	In case of a transmission collision, the terminal with master status is permitted to continue sending; a terminal with slave status has to await a new opportunity.
Step 8	The session is terminated when the terminals switch to the telewriting OFF condition.

Part 3 – Zone coding

1 General

1.1 This part of the Recommendation defines details of the zone coding method.

1.2 For an application of zone coding together with telephony the combined requirements from Parts 1, 2 and 3 apply.

1.3 This part also specifies how the coded signal is to be structured in 8 bit bytes, in order to fit in the transmission words defined in Part 2.

1.4 In the writing pad, the beginning of a stroke of handwriting is recognized by the detection of the pen-down condition.

1.5 Each stroke generates a set of time serial coordinate pairs during pen-down.

1.6 The coordinates of handwriting during pen-down are sampled at a fixed rate of 40 samples/second.

1.7 The first sampling is initiated by pen-down, and continues, ending when the pen is lifted.

1.8 The sequence of coordinate pairs is converted into a coded representation according to the zone coding rules. After this conversion the stroke is represented by the presentation element TRACE.

1.9 The presentation elements are coded in the form of opcodes and operands.

1.10 The opcodes have a fixed 8-bit length; the operands have a variable length.

1.11 The telewriting coordinate information is contained in the operands.

2 Presentation elements

2.1 In tone coding, the following presentation elements are distinguished:

- trace
- marker
- partial erasure
- untrace
- set colour
- line thickness
- complete erasure.

These elements and the format of the associated command streams are defined in Table 3-1/T.150.

2.2 The opcodes are defined in Table 3-2/T.150 (notation x/y means column x, row y, in a 16 × 16 code table).

3 Zone coding description

3.1 A trace is coded as a sequence of vectors (vector = D).

3.2 The beginning of a trace is the starting point of the first vector.

3.3 The end point of a vector constitutes the starting point for the next vector in the trace.

3.4 The starting point position of the first vector of each trace is coded in the form of a pair of absolute coordinates.

3.5 The position of each endpoint is determined by means of a measurement system, the origin of which must coincide with the starting point of the vector.

3.6 Within this measurement system, the endpoint position is found through a three step approximation:

- step 1: the quadrant θ, one value out of four; see Figure 3-1/T.150;
- step 2: the zone k within the quadrant; for division and numbering, see Figure 3-2/T.150;
- step 3: the relative address A within the zone.

3.7 In the coded representation, the quadrant and zone are indicated in a differential way: dθ and dk.

3.8 A set of 30 combinations of dθ and dk are selected to be coded in a compressed form, see Table 3-3/T.150.

3.9 The relative address within the zone has a length depending upon the size of the zone.

3.10 A vector end point position of which the combination dθ and dk is not defined in Table 3-3/T.150 is coded by EFZ (escape from zone code) followed by the absolute address.

3.11 The end of a trace is indicated by PLI (pen lift indicator) following the last (relative or absolute) address.

3.12 The zone coding is defined more precisely in §§ 4 and 5. An example of this coding is given in § 6.

4 Definitions of terms used in coding

4.1 The vector D_i defined by:

$$D_i = P_i - P_{i-1}$$

$$= (dx_i, dy_i) = (x_i - x_{i-1}, y_i - y_{i-1})$$

where P_i is the i-th coordinate pair during pen-down.

TABLE 3-1/T.150

Presentation element commands

Trace TRn	– The TRn command draws line segments that are defined by a co-ordinate information operand. – The TRn command stream is: ISP, TRn, ... co-ordinate information ... ISP.
Marker MKn	– The MKn command draws a marker pattern, the center of which is specified by a co-ordinate information operand. – The MKn command stream is: ISP, MKn, ... co-ordinate information ... ISP.
Partial Erasure PEn	– The PEn command erases the closed area defined by a co-ordinate information operand. – The PEn command stream is: ISP, PEn, ... co-ordinate information ... ISP.
Untrace UTn	– The UTn command erases the square area (with its sides parallel to the sides of the unit area) the centre of which is specified by a co-ordinate operand. – The size of the square is defined as follows: $(32 \times 2^{n-9} - 1) \times (32 \times 2^{n-9} - 1)$ grid units. – The UTn command stream is: ISP, UTn ... co-ordinate information ... ISP.
Set Colour SC*	– The SC* command sets a colour attribute to a particular trace. The colour attribute* can be set at the values: * = R: red * = B: blue * = G: green * = M: magenta * = Y: yellow * = C: cyan * = W: white – The effect of an SC* command remains valid until the next SC* or CE command. – The SC* command stream is: ISP, SC*, ISP, TRn, ... co-ordinate information ... ISP.
Line Thickness LT*	– The LT* command sets a line thickness that is defined by *, as follows: * = 1: one grid unit width, * = 2: two grid units width, * = 3: three grid units width. – The effect of a LT* command remains valid until the next LT* or CE command. – The LT* command stream is: ISP, LT*, ISP, TRn, ... co-ordinate information ... ISP.
Complete Erasure CE	– The displayed image is erased completely. – The CE command is: ISP, CE, ISP.

n determines the grid resolution,

n = 9 means: grid resolution = 512 × 512 (default value),

n = 10 means: grid resolution = 1024 × 1024,

n = 11 means: grid resolution = 2048 × 2048,

ISP Information Separator.

TABLE 3-2/T.150

Zone coding presentation opcodes

Element	Command	Code
Trace	TR 9 TR 10 TR 11	12/9 12/10 12/11
Marker	MK 9 MK 10 MK 11	13/9 13/10 13/11
Partial erasure	PE 9 PE 10 PE 11	14/9 14/10 14/11
Untrace	UT 9 UT 10 UT 11	15/9 15/10 15/11
Set colour	SC R SC G SC Y SC B SC M SC C SC W	11/0 11/1 11/2 11/3 11/4 11/5 11/6
Line thickness	LT 1 LT 2 LT 3	10/0 10/1 10/2
Complete erasure	CE	0/12

4.2 The quadrant number of the i-th vector, θ_i, is defined as (see Figure 3-1/T.150):

$\theta_i = 1$ for $dx \geqslant 0, dy \geqslant 0$

$= 2$ for $dx < 0, dy \geqslant 0$

$= 3$ for $dx < 0, dy < 0$

$= 4$ for $dx \geqslant 0, dy < 0$

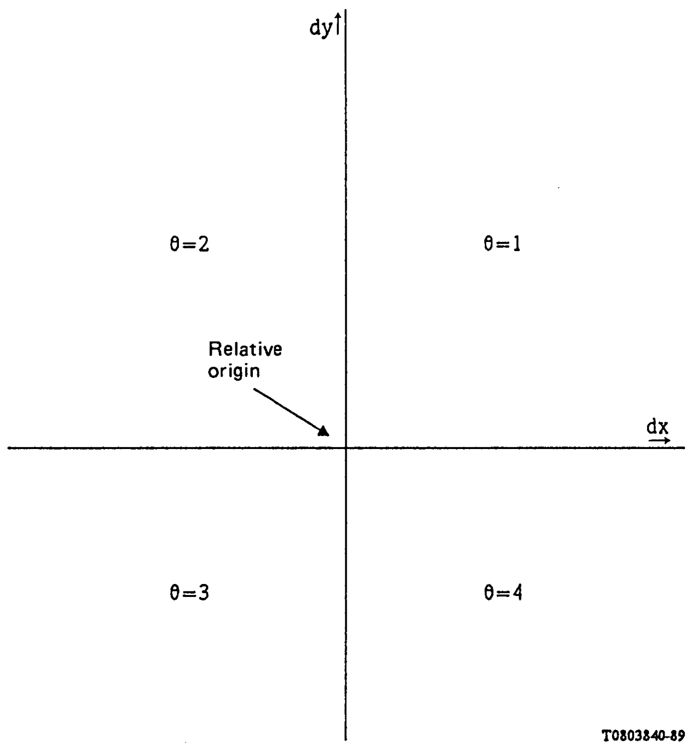


FIGURE 3-1/T.150

Definition of quadrant number

4.3 *Zone division and zone designation number*

The space of vectors without signs is divided into square zones. The zones are numbered counter-clockwise, as shown in Figure 3-2/T.150.

The zone width is taken as the power of two. Thus the width of the k-th zone is defined as:

$W(k) = 2$ for $k = 1$

$= 2 \times 2^{(k-2)/3}$ for $k > 1$

4.4 The k-th zone Z_k is defined as:

1) for $k = 1$

$Z_k = (|dx|, |dy|); 0 \leqslant |dx| \leqslant W(k)-1, 0 \leqslant |dy| \leqslant W(k)-1$

2) for $k > 1$

a) for $k = 0 \pmod 3$

$Z_k = (|dx|, |dy|); W(k) \leqslant |dx| \leqslant 2W(k)-1, W(k) \leqslant |dy| \leqslant 2W(k)-1$

b) for $k = 1 \pmod 3$

$Z_k = (|dx|, |dy|); 0 \leqslant |dx| \leqslant W(k)-1, W(k) \leqslant |dy| \leqslant 2W(k)-1$

c) for $k = 2 \pmod 3$

$Z_k = (|dx|, |dy|); W(k) \leqslant |dx| \leqslant 2W(k)-1, 0 \leqslant |dy| \leqslant W(k)-1$

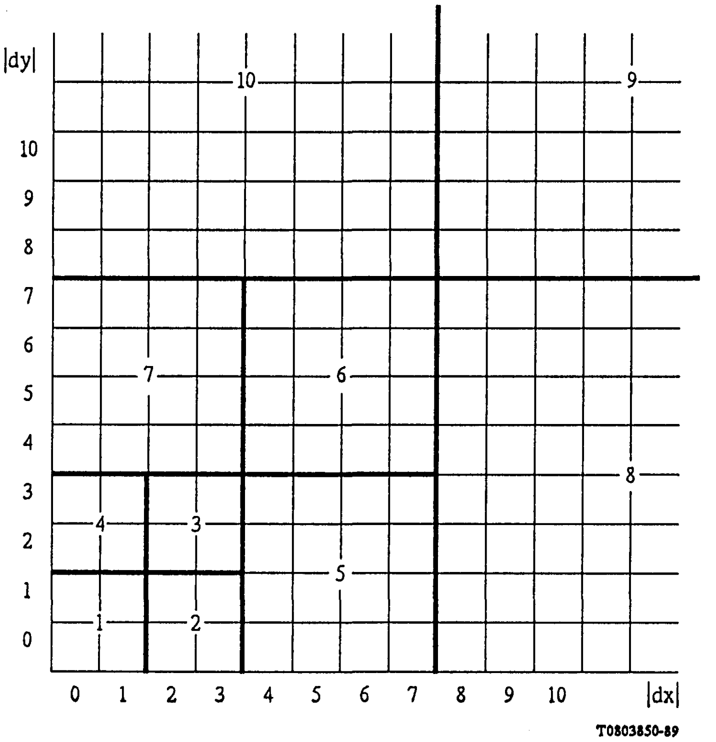


FIGURE 3-2/T.150

Zone division and zone designation number

4.5 The origin of the relative addresses in each zone is the lower left corner. The relative address in the k-th zone, (A_x, A_y), is defined as:

1) for $k = 1$

   $A_x = dx, A_y = dy$

2) for $k > 1$

   a) for $k = 0 \pmod 3$

      $A_x = |dx| - W(k), A_y = |dy| - W(k)$

   b) for $k = 1 \pmod 3$

      $A_x = |dx|, A_y = |dy| - W(k)$

   c) for $k = 2 \pmod 3$

      $A_x = |dx| - W(k), A_y = |dy|$

4.6 Quadrant number difference $d\theta_i$ is defined as:

$$d\theta_i = \theta_i - \theta_{i-1}$$

where $\theta_0 = 1$ for simplicity.

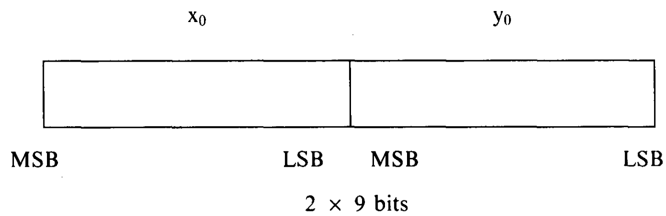
4.7 Zone number diffdrence dk_i is defined as:

$$dk_i = k_i - k_{i-1}$$

where k_i is the zone number obtained by the i-th vector, and $k_0 = 1$ for simplicity.

5 Specification of the coding

5.1 The first pen-down point is represented by the binary expression of the absolute coordinate pair (x_0, y_0), as follows:



MSB Most significant bit

LSP Least significant bit

5.2 All successive pen-down points are represented by zone codes (ZC) and relative addresses (A_x, A_y).

5.3 The zero vector (0, 0) is not coded and transmitted. It is also possible the zone vector ($|X_i - X_{i-1}| \leqslant 1$, $|Y_i - Y_{i-1}| \leqslant 1$) will be rejected before being coded.

5.4 The zone code is defined in Table 3-3/T.150. The table specifies a zone code number 1 to 30 and a bit combination for 30 combinations of $d\theta$ and dk.

5.5 The relative addresses (A_x, A_y) are represented by:

5.6 The bit length L is decided by:

$L = 2 \log_2 W(k)$.

5.7 For the combination of $d\theta$ and dk, not defined in Table 3-3/T.150, the absolute addresses (x_i, y_i) follow EFZ, instead of ZC.

5.8 A stroke is terminated by the pen lift indicator (PLI) as soon as the pen is lifted.

5.9 The full data format of a stroke is illustrated in Figure 3-3/T.150.

6 A coding example

The trace of handwritten information is shown in Figure 3-4/T.150, where P_i is the sampled point. An example of how to encode the coordinate data is shown in Table 3-4/T.150. The zone coded bit stream is shown in Figure 3-5/T.150.

7 Data structure

7.1 The zone coding opcodes and operands and the opcodes representing control commands are transmitted in the form of data packets.

7.2 Each packet consists of a header octet ISP (information separator), followed by an integral number of octets, and terminated by an ISP octet.

7.3 A packet may contain an undetermined number of opcodes. Boundaries of opcodes coincide with the boundaries of octets.

7.4 Data of variable length (the operand) is preceded by an opcode. After each operand the packet is terminated by an ISP octet at the earliest regular octet boundary.

7.5 If the end of the operand does not coincide with an octet boundary, the remaining bit positions until the octet boundary shall be filled with bits of the value ZERO.

At the receiving end, these zeros are ignored.

TABLE 3-3/T.150

Zone code table

Zone code No.	dθ	dk	Length of the code (bit)	Code (the left bit is LSB)
1	0	0	2	01
2	3	0	4	00 01
3	1	0	4	11 11
4	0	3	4	00 10
5	0	1	4	10 11
6	0	−3	4	11 10
7	3	3	5	10 01 1
8	0	−1	5	00 11 1
9	3	−1	6	10 01 01
10	3	−3	6	10 00 01
11	2	0	6	00 11 01
12	1	3	6	10 10 01
13	1	1	6	10 00 11
14	1	−3	6	10 10 11
15	0	4	6	10 00 10
16	0	2	6	00 00 11
17	0	−2	6	00 00 01
18	3	2	7	10 00 00 1
19	3	1	7	10 01 00 1
20	2	3	7	10 10 10 0
21	1	2	7	10 10 00 1
22	1	−1	7	00 11 00 1
23	1	−2	7	10 01 00 0
24	0	6	7	00 00 00 1
25	0	−4	7	00 11 00 0
26	0	−6	7	10 10 00 0
27	3	6	8	10 10 10 10
28	2	1	8	10 00 00 01
29	2	−1	8	10 10 10 11
30	2	−3	8	00 00 00 01
PLI			3	11 0
EFZ			6	00 00 10
NULL			8	00 00 00 00

PLI Pen lift indicator

EFZ Escape from zone code

A_x	A_y
MSB … LSB	MSB … LSB

EFZ	x_i	y_i
	MSB	MSB

x_0	y_0	ZC(1)	A(1)	ZC(2)	A(2)	...	PLI

x_0, y_0 Starting address

ZC(i) Zone code of the i-th vector

A(i) Relative address of the ith vector

PLI Pen lift indicator

FIGURE 3-3/T.150

Stroke data format

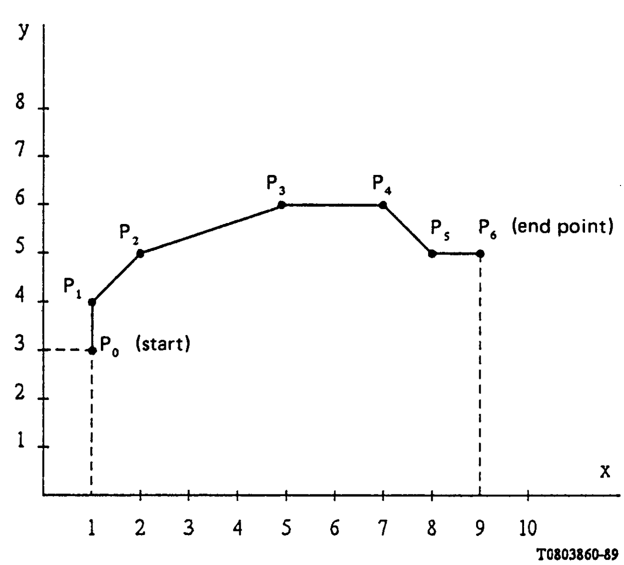


FIGURE 3-4/T.150

Handwritten information trace

TABLE 3-4/T.150

Coding example

i	x, y	dx, dy	θ	k	dθ	dk	ZC	Ax, Ay	W(k)	L/2	ZC-code
0	1, 3		(1)	(1)							
1	1, 4	0, 1	1	1	0	0	1	0, 1	2	1	01
2	2, 5	1, 1	1	1	0	0	1	1, 1	2	1	01
3	5, 6	3, 1	1	2	0	1	5	1, 1	2	1	1011
4	7, 6	2, 0	1	2	0	0	1	0, 0	2	1	01
5	8, 5	1, −1	4	1	3	−1	9	1, 1	2	1	100101
6	9, 5	1, 0	1	1	1	0	3	1, 0	2	1	1111

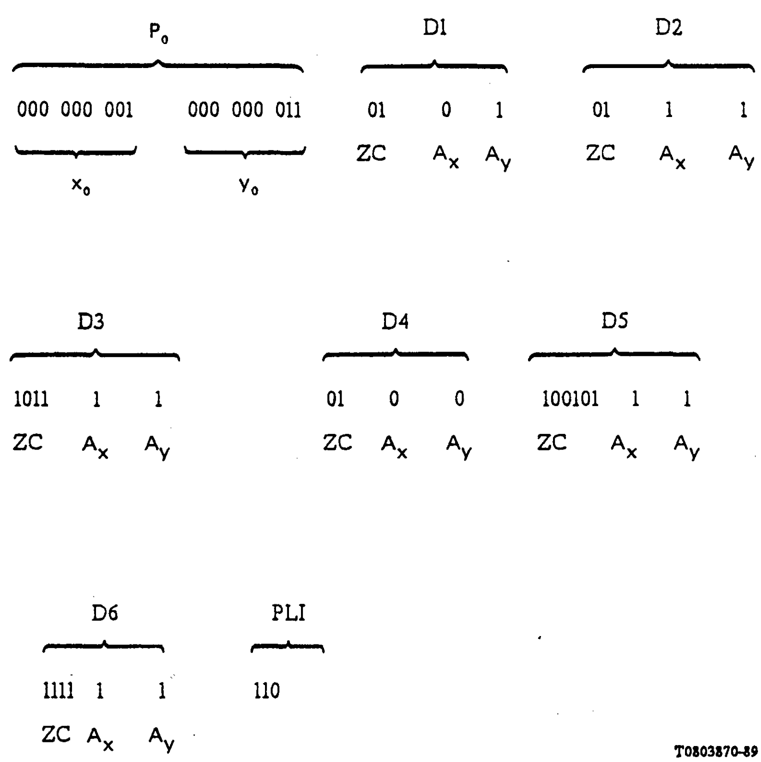


FIGURE 3-5/T.150

Zone coded bit stream

7.6 Successive packets may be sent contiguously, separated by a single ISP octet. See Figure 3-6/T.150.

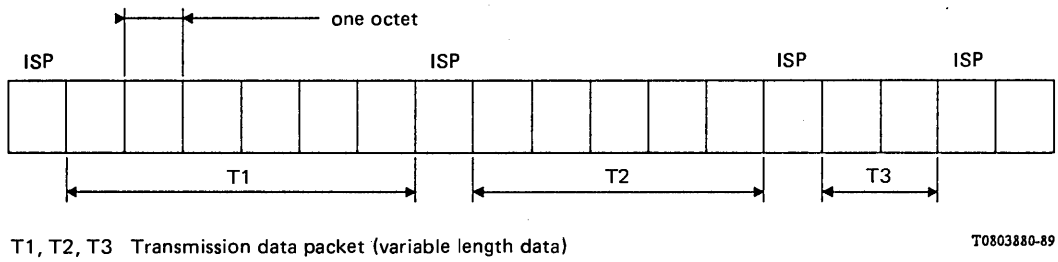


FIGURE 3-6/T.150

Packet organization

7.7 If one of the octets containing variable length data accidentally imitates an ISP octet, the transmitter inserts an extra ISP octet, so that the imitation is duplicated. See Figure 3-7/T.150.

If the imitation results from a combination of bits in two adjacent octets, no action is taken.

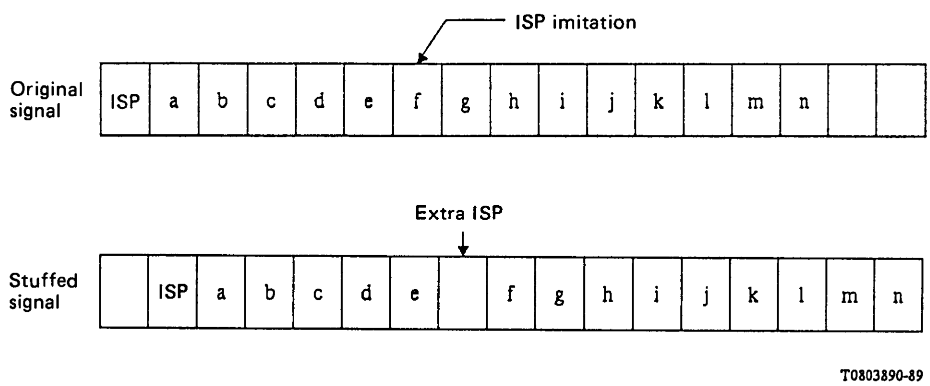


FIGURE 3-7/T.150

ISP octet stuffing

7.8 The receiver ignores the second ISP octet from each pair of ISP octets.

8 Temporary pen-stop

8.1 During the writing process, the pen may stop at an arbitrary instant, remaining on the writing surface. As a consequence, the completion of the current operand is suspended.

8.2 Generally, the instant of pen-stop does not coincide with a byte boundary. In order to provide the receiving party with up-to-date information including the correct pen-stop position, the content of the incomplete byte should be transmitted prior to the MARK signal representing the writing pause.

8.3 The above can be achieved by means of the insertion of 8 extra bits, the NULL bits, in the bitstream. Each NULL bit has the binary value Zero.

8.4 The NULL bits are subdivided into two groups, one group preceding the MARK signal, the other group following the MARK signal.

8.5 The number of NULL bits in the first group equals the number of open bit positions in the current byte. This number is referred to as N.

8.6 By the inclusion of N NULL bits the current byte is complete and can be transmitted. It is followed by the MARK signal.

8.7 As soon as the next writing activity occurs, the MARK signal is terminated.

8.8 The remaining 8-N NULL bits are to occupy the leading bit-positions of the first byte after the MARK signal.

8.9 The NULL bit mechanism is illustrated in Figure 3-8/T.150.

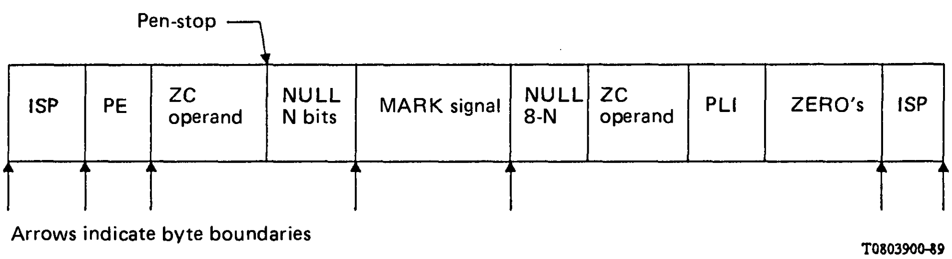


FIGURE 3-8/T.150

NULL bit insertion mechanism

9 Control commands

9.1 This section defines control commands, affecting the functioning of the terminal at the presentation level.

These commands are:

- complete erasure,
- escape,
- information separator.

9.2 *Complete erasure CE*

This command is defined already in Table 3-1/T.150. It is repeated here because of the buffer control aspect.

The displayed image is erased completely, both at the sending side and the receiving side. Also the telewriting data in the transmission buffer at the sending side, and in the reception buffer at the receiving side is erased.

9.3 *Excape ESC*

This is a code extension command. ESC is to be followed by an 8-bit operand, defining an alternative code table. ESC + operand is to be sent by an enhanced Telewriting terminal prior to each enhanced operation function. Details are to be defined in a section on enhanced terminal.

9.4 *Information separator ISP*

ISP acts as a delimiter of command packets as defined in § 7. The terminal should check received data streams for pairs of ISP octets and, where required, should reject every second ISP octet.

9.5 The coding of the above commands is defined in Table 3-5/T.150 (the notation x/y means column x, row y, in a 16 × 16 code table).

TABLE 3-5/T.150

Coding of control commands

Function	Acronym	Coding
Complete erasure	CE	0/12
Escape	ESC	1/11
Information separator	ISP	1/15

10 Summary code table

A summary of the coding for the opcodes is given in Figure 3-9/T.150. All elements included have been defined in the previous sections.

bbbb 4321		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	b8	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	b7	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	b6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	b5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0000	0											LT1	SCR				
0001	1											LT2	SCG				
0010	2											LT3	SCY				
0011	3												SCB				
0100	4												SCM				
0101	5	HLO											SCC				
0110	6												SCW				
0111	7		SSO														
1000	8																
1001	9													TR9	MK9	PE9	UT9
1010	10													TR10	MK10	PE10	UT10
1011	11		ESC											TR11	MK11	PE11	UT11
1100	12	CE															
1101	13																
1110	14																
1111	15		ISP														

T0803910-89

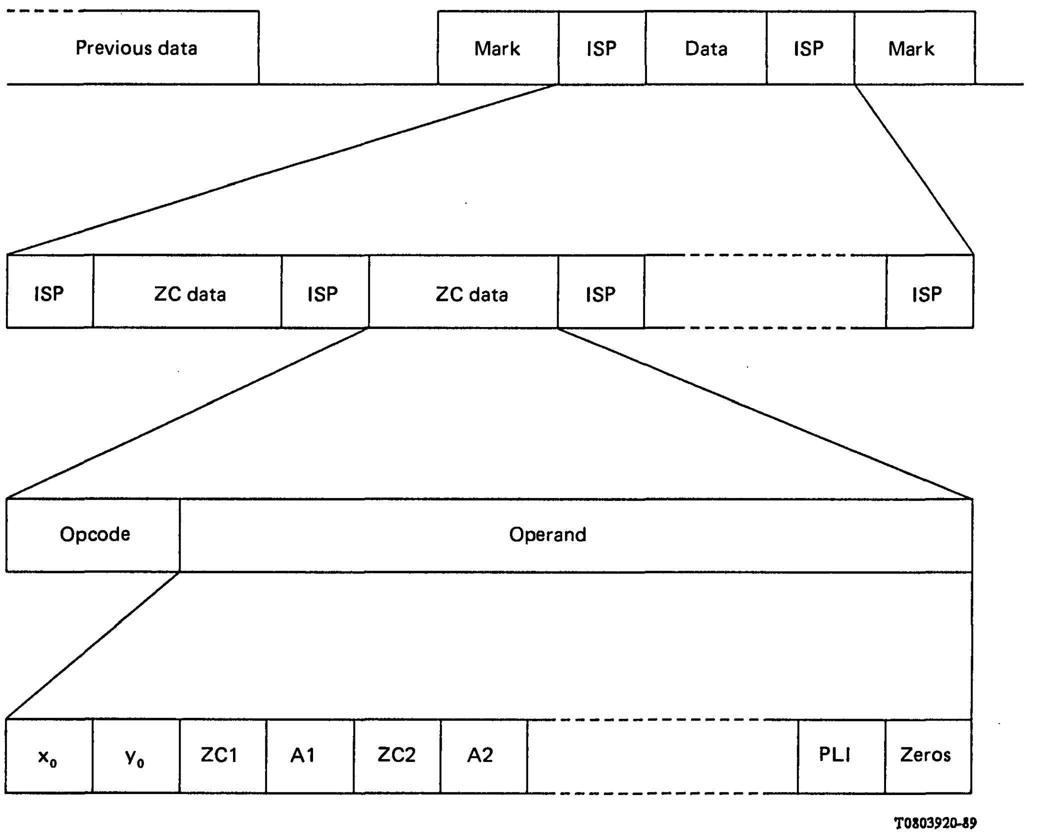
S8 0643A/272-12'87/ZWA
b1 LSB
b8 MSB

FIGURE 3-9/T.150

Summary code table

11 Summary transmission data format

The transmission data format is illustrated in Figure 3-10/T.150.



Mark Mark polarity
ISP Information separator octet
ZC Zone coding
x_0, y_0 Starting coordinate
A Relative address
PLI Pen lift indicator
Zeros Added zeros, to give the operand a length of an integral number of octets

FIGURE 3-10/T.150

Summary transmission data format

12 Zone coding basic terminal

12.1 The basic terminal must be able to receive and correctly process the following presentation element commands:

TR 9, MK 9, PE 9, CE, ISP.

12.2 The following presentation elements are optional:

TR 10, TR11
MK 10, MK 11
PE 10, PE 11
UT 9, UT 10, UT 11.

I.e. the transmitter may or may not be equipped with these commands.

The receiver must be able to receive and correctly process these commands.

12.3 The following control commands are optional:

ESC, LT*, SC*

I.e. the receiver will accept these commands but does not undertake any further action.

Part 4 – Differential chain coding

1 General

1.1 This part of the Recommendation defines details of the differential chain coding method.

1.2 For an application of differential chain coding together with telephony, the combined requirements from Parts 1, 2 and 4 apply.

1.3 Differential chain coding is derived from the Videotex geometric coding as defined in Recommendation T.101, Annex C (CEPT Videotex).

1.4 The telewriting functionalities are nearly a subset of the Videotex geometric functionalities, as defined in Recommendation T.101, Annex C.

1.5 Differential chain coding was developed for compression purposes. In this coding method, the statistical properties of handwriting are employed.

1.6 This coding method uses spatial sampling of curves, as distinct from sampling with a fixed frequency. The size of the sampling steps is determined by the size of the so-called coding ring.

1.7 The precision of this coding method is expressed in grid units, GU. In the default situation, one GU corresponds to the binary fraction 2 ** –9 of the unit length.

1.8 Each stroke of handwriting is processed by the writing pad circuitry and converted into a coded form.

The coded representation of a stroke is called TRACE.

1.9 The coding of the presentation element trace, as well as the coding of the remaining presentation elements is defined in terms of 7 bit coding.

1.10 Conversion into 8 bit structured coding as required for transmission, is also specified in this Recommendation.

1.11 The word "byte" where used in this Recommendation, refers to a combination of 7 or 8 bits, whatever is appropriate in the given context.

2 Presentation elements

In differential chain coding, the following presentation elements are distinguished:

- trace
- marker
- closed area
- partial erasure
- background
- complete erasure.

The attributes are:

- colour
- trace thickness
- trace texture.

These presentation elements together with the attributes are described in Table 4-1/T.150.

TABLE 4-1/T.150

Differential chain coding presentation elements

Element	Description
Trace	The trace is coded as a trace opcode plus a set of co-ordinate information defining a sequence of line segments. Trace corresponds with polyline in videotex.
Marker	The marker is coded as a marker opcode plus a single co-ordinate pair defining the position ot the marker's center point.
Closed area	The closed area is coded by an opcode plus a set of co-ordinate information defining a closed perimeter. The closed area corresponds with fill area in videotex.
Partial erasure	Partial erasure is obtained by means of the closed area concept. By giving the closed area the same attributes as the background, erasure is achieved for the area enclosed in the perimeter.
Background	At initialization and after complete erasure, the background shows default appearance. Modification of the background is obtained by means of the closed area concept. The closed area is chosen to have the size of the image area. The area attributes are set to the desired background appearance.
Complete erasure	Complete erasure is obtained by means of the clear screen concept. The whole image area will be set to the default background appearance.
Colour	Colour is an attribute, applicable to trace and closed are (including background). The effect of a colour command remains valid until the next colour command.
Trace thickness	Trace thickness is an attribute. It is determinated by means of a scale factor. The effect of a trace thickness command remains valid for all subsequent traces, until the next trace thickness command.
Trace texture	Trace texture is an attribute. It is determinated by means of a parameter allowing a choice amoung defined textures. The effect of a trace texture command remains valid for all subsequent traces, until the next trace texture command.
Marker type	Marker type is an attribute. It is determinated by means of a parameter allowing a choice among defined textures. The default value of marker type is 1. If the specified value is outside the range 0 . . . 4, the marker is not displayed.

3 Description of the coding

3.1 The coded representation of a presentation element is called PRIMITIVE.

3.2 A primitive is composed of one opcode and a number of operands as required.

3.3 Certain opcodes are coded as a single byte, other opcodes are coded as combinations of two bytes.

3.4 The operand part of a primitive may utilize either basic format encoding or pointlist encoding.

3.5 In basic format encoding the operand part of the primitive contains one or more operands, each consisting of one or more bytes.

3.6 In the pointlist encoding the operand part of the primitive contains coordinate information regarding an individual point or regarding a sequence of related points.

3.7 The position of an individual point, as well as the position of each first point of a sequence, is coded in absolute coordinates, i.e. the x- and y-coordinate with respect to the origin of the coding space.

3.8 For the coding of the remaining points of a sequence, a choice is to be made among two possibilities, namely displacement mode and incremental mode.

3.9 In the displacement mode, each point (after the first) is coded by means of two size value parameters. The first size value gives the x-component of the point's displacement from the preceding point in the sequence, the second size value gives the y-component of the displacement.

3.10 In the incremental mode, a mechanism is used in which a single value, derived from a table, determines the position of a point with respect to the preceding point. This mechanism is suitable for coding a sequence of points containing a high amount of position information, such as a trace.

3.11 The mechanism, introduced in § 3.10, is based on the use of a coding ring. At the beginning of trace, the starting point determines the centre point of the first ring. The intersection of trace and ring is identified and determines the centre point of the second ring.

3.12 Each new intersection determines the centre point for the next ring. Thus, the trace is represented by the starting point plus the series of intersection points. The end of a trace is indicated by means of the end of block-code.

3.13 The method for identifying the various points on a ring utilizes small numbers for points with a high probability of being intersected and larger numbers for points with lower probability.

3.14 The numbering system for the reference points on the ring is defined in §§ 4.6 and 4.7.

4 Incremental mode mechanism

4.1 The coding data in the incremental mode does not reflect coordinate size values, but represents a sequence of points identified by means of successive coding rings. Each ring identifies one point.

4.2 A ring is a set of reference points, positioned on the perimeter of a square. The position of the square is identified through the position of its centre point. The sides of the square are parellel to the x- and y-axes.

4.3 The characteristics of the ring are determined by its radius R, its angular resolution factor p and its direction D.

The size of R is expressed in GU.

4.4 The number of reference points on a ring is N. The value of N is determined by:

$$N = \frac{8R}{2^p}, \text{ with } p = 0, 1, 2, 3.$$

It follows that the maximum number of reference points is $N = 8R$.

4.5 N must be even. If N is odd, the encoded operand (the point list) must be discarded. If N is even for the first part of the operand, but N is odd for the remaining part, the remaining part (with N being odd) is discarded.

4.6 To the reference points on the ring, point numbers are assigned as follows. The numbering starts with 0. The point with number 0 is called the direction point.

4.7 The default position for the direction point is shown in Figure 4-1/T.150. Adjacent points are numbered 1 ... N/2-1 in anticlockwise direction, and $-1 \ldots -N/2$ in clockwise direction. Figure 4-1/T.150 shows two rings with the numbered reference points.

4.8 In the figure the left ring is characterized by $R = 3$ and $p = 0$; the right ring by $R = 3$ and $p = 1$.

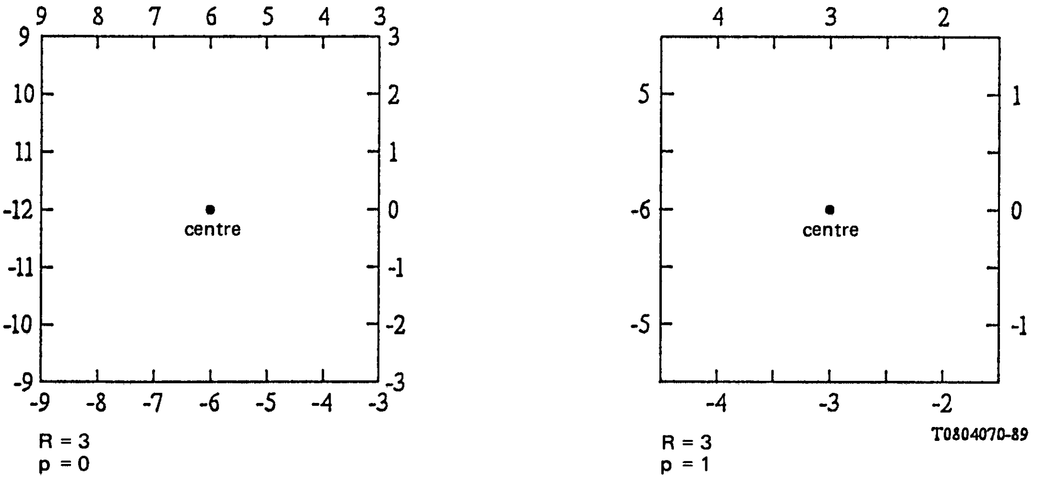


FIGURE 4-1/T.150

Two ring examples

4.9 The position of the reference points on each ring is fixed. However, the allocation of the point numbers is adapted to the trace direction as follows.

4.10 For the first ring of a sequence (at the starting point), the direction point is at default position, as shown in Figure 4-1/T.150.

4.11 As soon as the growing trace intersects the first ring, the nearest reference point is determined. This point constitutes the centre point for the next ring.

4.12 The direction point on the second ring is located at that position where the next intersection would take place if the trace continued as a straight line.

4.13 As the trace grows, the nearest reference point at each intersection is determined. The respective point numbers of these points are converted into variable length code words according to the Huffman code table, defined in Table 4-2/T.150.

4.14 The radius can have a value of R0, 2R0, 4R0 or 8R0, where R0 is the basic radius.

The angular resolution factor p can have a value of 0, 1, 2 or 3.

To modify these parameters the code table contains the codes C1 ... C6. For their use, see further on.

The basic radius R0 can be specified by the primitive "set domain ring". The default basic radius follows from:

default basic radius = 2 ** max (0, −8-granularity code).

4.15 The length of the code table is fixed. The point numbering ranges from −20 to +19. For the encoding in cases of rings with a higher number of reference points, two escape codes are defined: IM-ESC 1 and IM-ESC 2. For their use, see § 5.

4.16 At the end of the trace no further intersections occur. The variable length coded string is terminated by end of block.

TABLE 4-2/T.150

Huffman code table for differential chain coding

Code No.	Length	Code-word	Point number
1	2	00	0
2	2	10	1
3	2	01	−1
4	4	1100	2
5	4	1101	−2
6	6	111000	3
7	6	111001	−3
8	6	111010	4
9	6	111011	−4
10	8	11110000	5
11	8	11110001	−5
12	8	11110010	6
13	8	11110011	−6
14	8	11110100	7
15	8	11110101	−7
16	8	11110110	8
17	8	11110111	−8
18	10	1111100000	9
19	10	1111100001	−9
20	10	1111100010	10
21	10	1111100011	−10
22	10	1111100100	11
23	10	1111100101	−11
24	10	1111100110	12
25	10	1111100111	−12
26	10	1111101000	13
27	10	1111101001	−13
28	10	1111101010	14
29	10	1111101011	−14
30	10	1111101100	15
31	10	1111101101	−15
32	10	1111101110	16
33	10	1111101111	−16
34	10	1111110000	17
35	10	1111110001	−17
36	10	1111110010	18
37	10	1111110011	−18
38	10	1111110100	19
39	10	1111110101	−19
40	10	1111110110	C1
41	10	1111110111	−20
42	10	1111111000	C2
43	10	1111111001	C3
44	10	1111111010	C4
45	10	1111111011	C5
46	10	1111111100	C6
47	10	1111111101	IM-ESC 1
48	10	1111111110	IM-ESC 2
49	10	1111111111	End of block

5 Change of coding parameters

5.1 The escape codes IM-ESC 1 and IM-ESC 2 enable the extension of the point numbering range on the ring. I.e. also points outside the range −20 to +19 can be addressed. By the code IM-ESC 1, the absolute value of the point number is increased by 20, the sign remains unchanged.

By the code IM-ESC 2, the absolute value of the point number is increased by 40, the sign remains unchanged.

5.2 The two escape codes can be used in combination with each other in any desired order. Some examples in Table 4-3/T.150 illustrate their use. The number between [ ] represents the point number.

TABLE 4-3/T.150

Use of escape codes, examples

Description		Intended point numbers
<IM-ESC 1>	[1]	21
<IM-ESC 1>	[−1]	−21
<IM-ESC 2>	[14]	54
<IM-ESC 2>	[−12]	−52
<IM-ESC 1> <IM-ESC 2>	[6]	66
<IM-ESC 2> <IM-ESC 1>	[−18]	−78

5.3 The codes C1 up to C6 are used to change the parameters R and p that define the ring to be used. The use of these codes is defined in §§ 5.4 and 5.10.

By the use of these codes the direction point is set at default position.

5.4 The range in which the parameters should remain is as follows:

R: R0, 2R0, 4R0, 8R0 (with R0 being the basic radius);

p: 0, 1, 2, 3.

5.5 Code C1 means: change R and p to the next higher value. E.g. if radius is R, the next higher is 2R; if p = 0 the next higher is 1.

R cannot become greater tan 8R0 and p cannot become greater than 3. E.g. if current radius is 8R0 or current p = 3, the code C1 has no effect.

5.6 Code C2 means: change R and p to the next lower value. The effect of C2 is the inverse of code C1.

R cannot become smaller than R0 and p cannot become smaller than 0. E.g. if the current radius is R0 or the current p = 0, the code C2 has no effect.

5.7 Code C3 means: change R to the next higher value. The code C3 has no effect if the current radius = 8R0.

5.8 Code C4 means: change p to the next higher value. The code C4 has no effect if the current p = 3.

5.9 Code C5 means: change R to the next lower value. The code C5 has no effect if the current radius = R0.

5.10 Code C6 means: change p to the next lower value. The code C6 has no effect if the current p = 0.

6 Coding formats

6.1 The coding is specified in terms of 7-bit coding. For use in the 8 bit environment as specified for transmission, bit No. b8 of each octet shall be set to ZERO.

6.2 For reference, an empty 7-bit code table is shown in Figure 4-2/T.150.

	b7	0	0	0	0	1	1	1	1
	b6	0	0	1	1	0	0	1	1
	b5	0	1	0	1	0	1	0	1
bbbb 4321		0	1	2	3	4	5	6	7
0000	0								
0001	1								
0010	2								
0011	3								
0100	4								
0101	5								
0110	6								
0111	7								
1000	8								
1001	9								
1010	10								
1011	11								
1100	12								
1101	13								
1110	14								
1111	15								
		Reserved for control functions		Opcodes		Operands			

T0803930-89

FIGURE 4-2/T.150

General structure of the code table for differential chain coding

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

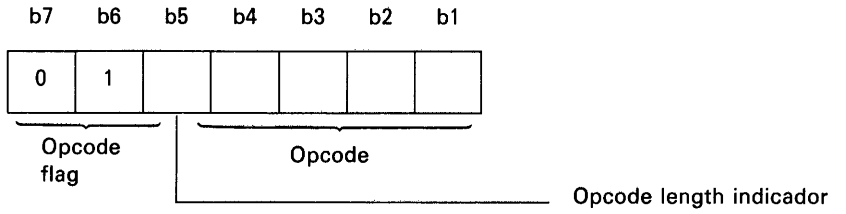


FIGURE 4-3/T.150

Opcode encoding structure

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

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

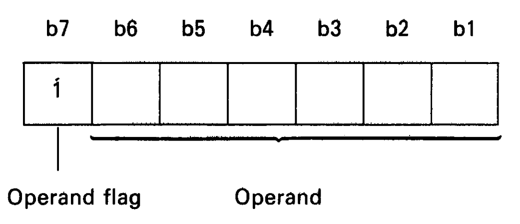


FIGURE 4-4/T.150

Operand encoding structure

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

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

- point P
- colour index CI
- integer number I
- real number R

These data types are coded according to the basic format.

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

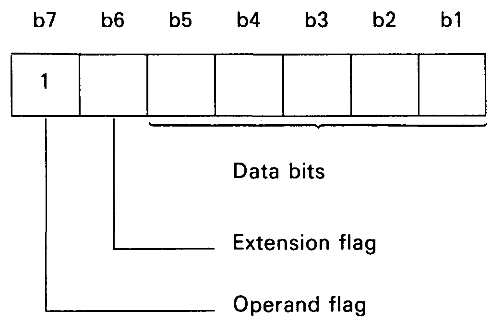


FIGURE 4-5/T.150

The basic format structure

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

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

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

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

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

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

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

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

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

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

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

7 Incremental mode coding format

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

- first point's position;
- DCC introducer;
- incremental sequence.

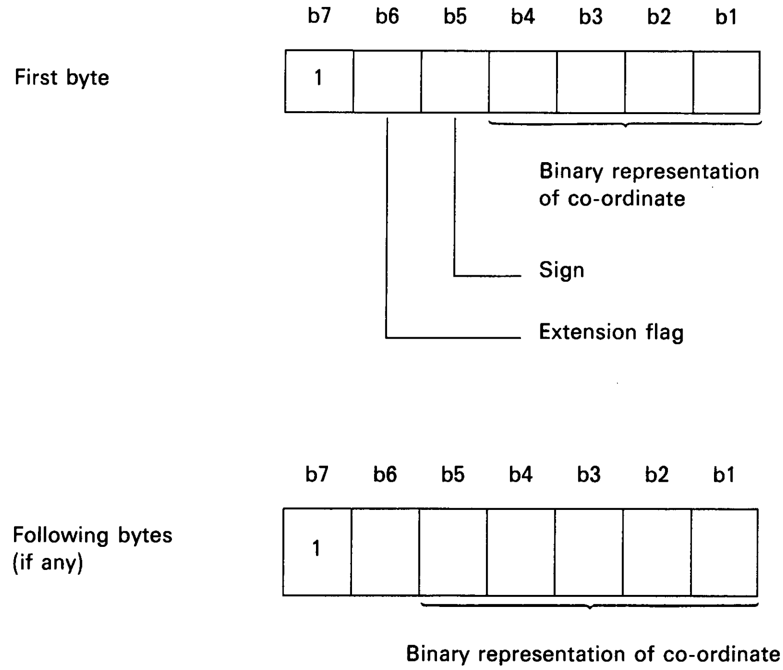


FIGURE 4-6/T.150

Absolute co-ordinate encoding structure

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

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

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

	b7	b6	b5	b4	b3	b2	b1
First byte	1	0	1	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1
Second byte	1	0	0	0	0	0	1

FIGURE 4-7/T.150

DCC introducer encoding

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

b7	b6	b5	b4	b3	b2	b1
1						
	Data					

FIGURE 4-8/T.150

Incremental sequence encoding

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

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

8 Displacement mode coding format

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

- first points position;
- following points.

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

9 Encoding of the primitives

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

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

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

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

Trace

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

OR

<Trace opcode: 2/0> <point: first point>
<DCC introducer: 5/0, 4/1> <Incremental sequence>

Closed area

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

OR

<Closed area opcode: 2/1>
<Point: first point> <DCC introducer: 5/0, 4/1>
<Incremental sequence>

Marker

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

OR

<Marker opcode: 3/2, 2/11, 5/2> <point: first point> (1)
<DCC introducer: 5/0 4/1> <Incremental sequence>

Clear

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

TABLE 4-4/T.150

Incremental Trace Coding opcodes

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

TABLE 4-5/T.150

Notational conventions

Item	Meaning
<symbols>	1 occurrence
<symbols> (n)	n or more occurrences, with n ⩾ 1
[comments]	Explanation of a production
<x : y>	Construction x with meaning y.

9.4 The attribute setting primitives are encoded as follows:

Trace thickness

<Set trace thickness opcode: 3/1, 2/1>
<real = trace thickness scale factor>

Trace texture

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

<integer: 0>	[SOLID]
<integer: 1>	[DASHED]
<integer: 2>	[DOTTED]
<integer: 3>	[DASHED DOTTED]
<all other values>	[RESERVED]

Trace colour

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

<index: 0>	[black]
<index: 1>	[red]
<index: 2>	[green]
<index: 3>	[yellow]
<index: 4>	[blue]
<index: 5>	[magenta]
<index: 6>	[cyan]
<index: 7>	[white]

Closed area interior style

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

<integer: 0>	[HOLLOW]
<integer: 1>	[SOLID]
<integer: 2>	[PATTERN]
<integer: 3>	[HATCH]
<all other values>	[RESERVED]

Closed area style index

<Set closed area style index opcode: 3/1, 2/6>
<integer: closed area style index> = interior style HATCH

<integer: 0>	[vertical lines]
<integer: 1>	[horizontal lines]
<integer: 2>	[45 degrees lines]
<integer: 3>	[−45 degrees lines]
<integer: 4>	[closed lines, vertical and horizontal]
<integer: 5>	[crossed lines, 45 and −45 degrees]
<all other values>	[reserved]

Closed area colour index

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

<index: 0>	[black]
<index: 1>	[red]
<index: 2>	[green]
<index: 3>	[yellow]
<index: 4>	[blue]
<index: 5>	[magenta]
<index: 6>	[cyan]
<index: 7>	[white]

Marker type

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

<integer: 0>	[DOT]
<integer: 1>	[PLUS SIGN]
<integer: 2>	[ASTERISK]
<integer: 3>	[CIRCLE]
<integer: 4>	[DIAGONAL CROSS]
<all other values>	[RESERVED]

Marker size

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

Marker colour

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

<index: 0>	[black]
<index: 1>	[red]
<index: 2>	[green]
<index: 3>	[yellow]
<index: 4>	[blue]
<index: 5>	[magenta]
<index: 6>	[cyan]
<index: 7>	[white]

9.5 The protocol descriptor primitives are encoded as follows:

Set domain ring

<Set domain ring opcode: 3/2, 2/4>
<integer: angular resolution factor>
<integer: basic radius of the ring>

Set coordinate precision

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

<integer: magnitude code>	[4]
<integer: granularity code>	[1 −9, −10, −11]
<integer: default exponent>	[1 −9, −10, −11]
<integer: explicit exponent allowed>	[1]

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

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

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

10 Example of differential chain coding

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

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

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

TABLE 4-6/T.150

T.150 coordinate values and reference point number

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

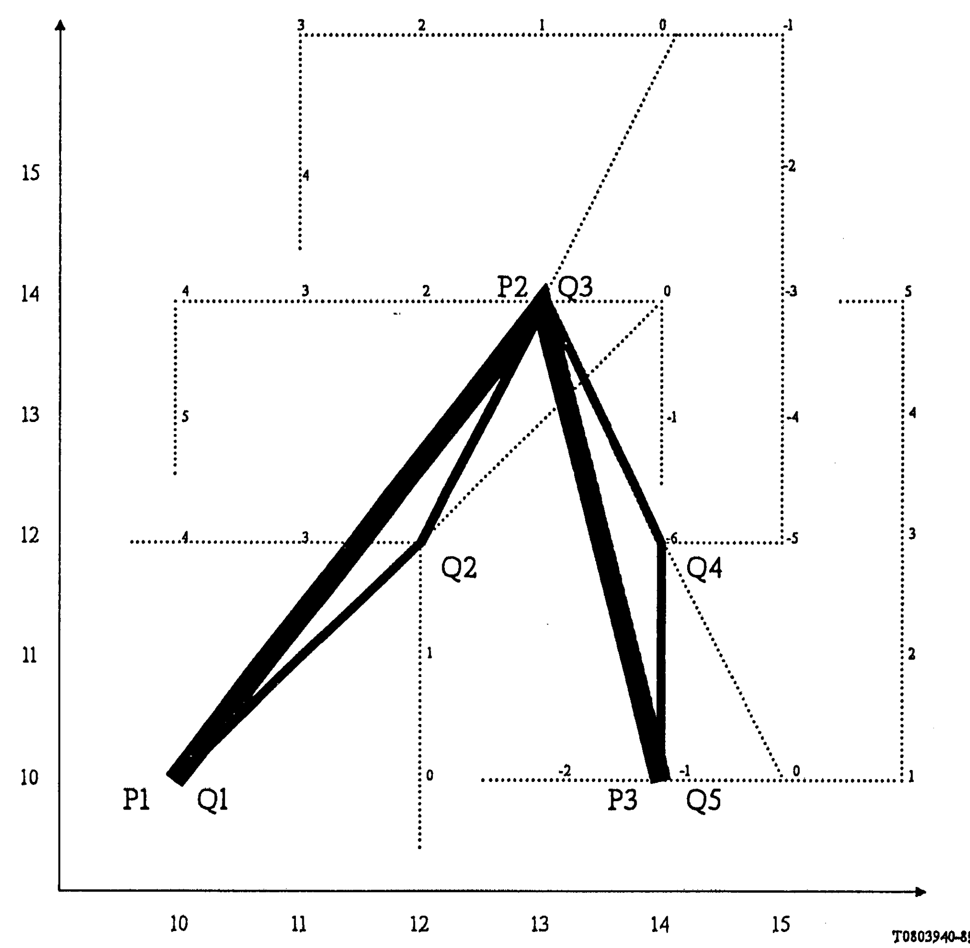


FIGURE 4-9/T.150

Example of incremental mode encoding (R = 2, p = 0)

Trace opcode	Q1 (absolute coordinates)		DCC introducer	
00100000	01001010	01001010	01010000	01000001
	value +10	value +10		

Q2 Q3	Q4	Q5	End of Bloc	
01110010	01111100	01110111	01111111	01110000
+2 +1	−6	−1	EOB	

Byte sequence: 2/0 4/10 4/10 5/0 4/1 7/2 7/12 7/7 7/15 7/0

FIGURE 4-10/T.150

DCC coded bitstream

01100000		
trace opcode		
01001010	01001010	P_1 = (10, 10)
value +10	value +10	
01000011	01000100	$P_2 - P_1$ = (3, 4)
value +3	value +4	
01000001	01010100	$P_3 - P_2$ = (1, −4)
value +1	value −4	

Byte sequence: 2/0 4/10 4/10 4/3 4/4 4/1 5/4

FIGURE 4-11/T.150

Displacement mode coded bitstream

GENERAL PRINCIPLES OF TELEMATIC INTERWORKING

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

The CCITT,

considering

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

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

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

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

unanimously declares

that this Recommendation describes the general principles for telematic interworking.

CONTENTS

0 Introduction

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

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

- allowing efficient interchange of information between telematic terminals which cannot interact directly;
- providing additional features to standardized telematic services, such as document storage and multiaddressing;
- providing access to or participation in CCITT defined services such as telex, interpersonnal messaging, directory services, etc.

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

- participation of telematic users in the IPM service;
- teletex to telex interworking.

Other TIAs are for further study.

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

1 Scope and field of application

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

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

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

2 References

This Recommendation cites the documents listed below:

- Rec. T.330: Telematic access to interpersonal messaging system
- Rec. X.400: Message handling systems: System and service overview
- Rec. X.402: Message handling systems: Overall architecture
- Rec. X.420: Message handling systems: Interpersonal messaging system.

3 Definitions

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

4 Symbols and abbreviations

AU	Access unit
C	Conditional/consumer
CF	Conversion facility
IPM	Interpersonal messaging
IPMS	Interpersonal messaging system
IPM-UA	Interpersonal messaging user agent
M	Multiple
MS	Message store
MTA	Message transfer agent
MTS	Message transfer system
PDS	Physical delivery system
PTTXAU	Public teletex access unit
TIA	Telematic interworking application
TIAS	Telematic interworking abstract service
TIU	Telematic interworking unit
TLM	Telematic
TLMA	Telematic agent
TLMAU	Telematic access unit
TLM-TER	Telematic terminal
TTX	Teletex
UA	User agent

5 Conventions

This Recommendation makes use of no specific convention.

6 Telematic interworking model

6.1 *Overview*

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

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

6.2 *Telematic interworking involving the MTS*

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

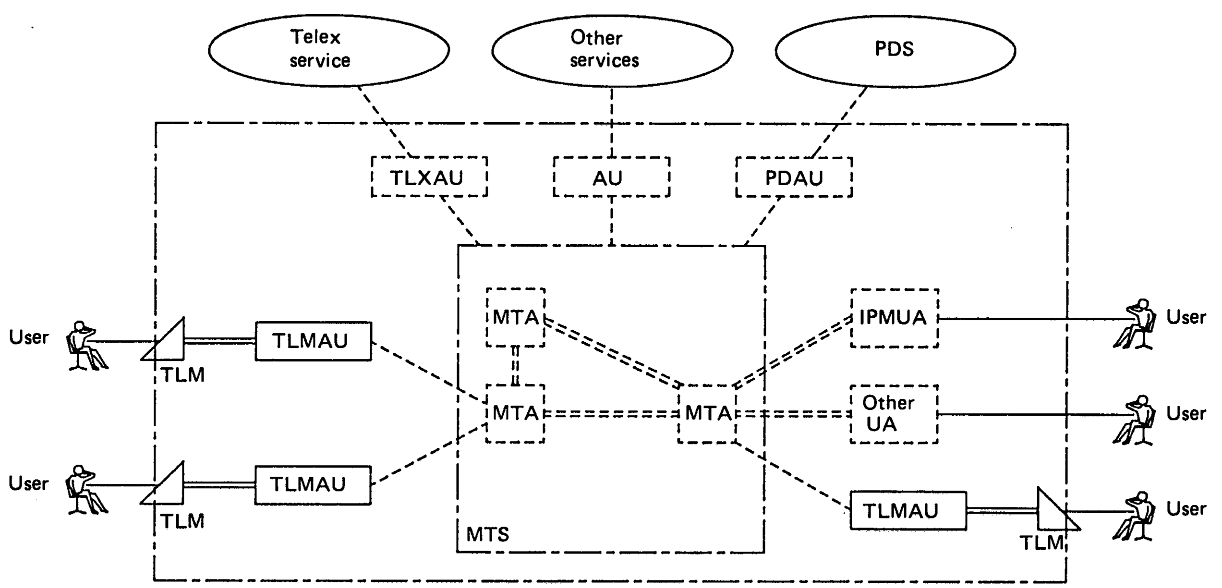
It allows:

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

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

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

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



TLM Telematic terminal
TLMAU Telematic access unit
TLXAU Telex access unit
PDAU Physical delivery access unit
IPM Interpersonal messaging
UA User agent
MTS Message transfer system
MTA Message transfer agent
PDS Physical delivery system
AU Access unit

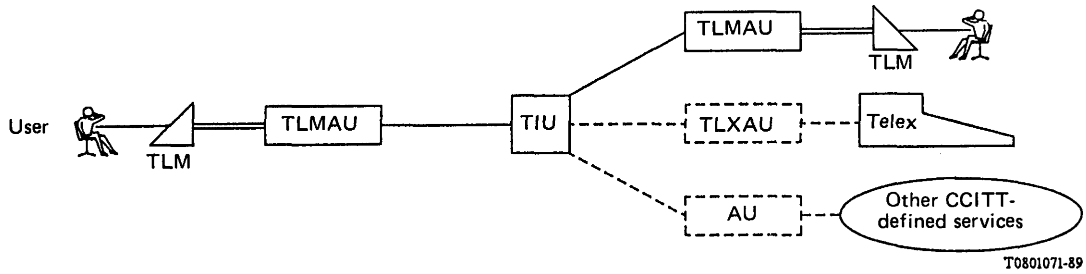
——— Relation without any associated protocol
═══ Relation which may be associated to a protocol
------ Not defined in T.300 Series of Recommendations

FIGURE 1/T.300

Telematic interworking model involving the MTS

6.3 *Telematic interworking not involving the MTS*

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



TLM Telematic terminal
TLMAU Telematic access unit
TLXAU Telex access unit
TIU Telematic interworking unit
AU Access unit

FIGURE 2/T.300

Telematic interworking model not involving the MTS

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

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

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

6.4.1 *TIF involving the MTS*

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

TABLE 1/T.300

Physical confirgurations of a TIF involving the MTS

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

M Multiple

[ ] Optional

MS Message store

Two rules apply to define a TIF in this case:

– At least one TLMAU is present.

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

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

6.4.2 *TIF not involving the MTS*

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

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

TABLE 2/T.300

Physical configurations of a TIF not involving the MTS

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

M Multiple

[ ] Optional

7 The telematic interworking system

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

7.1 *Definition of the TISs*

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

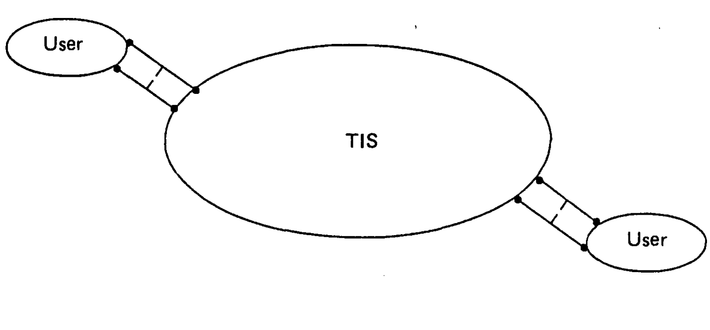


FIGURE 3/T.300

Definition of the TIS

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

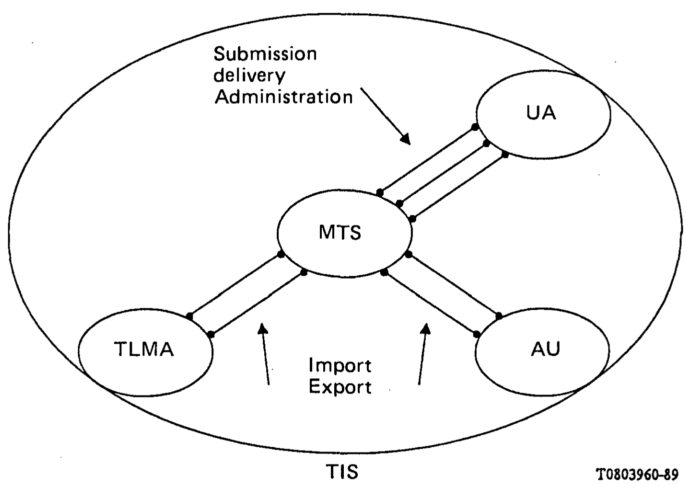
All ports and operations are application-dependent, and thus are described in appropriate (application-specific) Recommendations.

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

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

7.2 *TIS involving the MTS*

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



UA User agent
AU Access unit
TLMA Telematic agent
MTS Message transfer system

FIGURE 4/T.300

TIS involving the MTS

The refinement or the TIS exposes the following component objects:

- The MTS, as defined in Recommendations X.402 and X.411.
- The TLMA, providing the user of a telematic terminal with the telematic interworking abstract service (TIAS).
- The UA, which provide TIAS to users not using a telematic terminal. One application is the IPM-UA, in Recommendation X.420.
- The AUs, which allow intercommunication with other CCITT-defined services. An example of such an AU is the PDAU, defined in the X.400 Series of Recommendations.

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

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

7.3 *TIS not involving the MTS*

This case is for further study.

8 Refinement of the TLMA principles of telematic access protocols

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

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

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

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

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

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

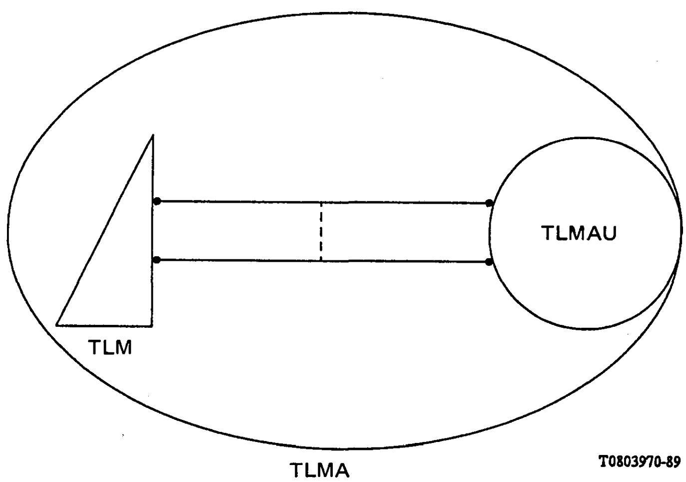


FIGURE 5/T.300

Refinement of the TLMA

Recommendation T.330

TELEMATIC ACCESS TO INTERPERSONAL MESSAGE SYSTEM

(Melbourne, 1988)

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

The CCITT,

considering

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

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

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

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

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

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

unanimously declares

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

CONTENTS

0 Introduction

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

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

1 Scope and field of application

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

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

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

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

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

2 References

This Recommendation cites the documents listed below.

2.1 *Telematic interworking*

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

2.2 *Message handling systems*

- Rec. X.400: Message handling systems: System and service overview
- Rec. X.402: Message handling systems: Overall architecture
- Rec. X.407: Message handling systems: Abstract service definition conventions
- Rec. X.411: Message handling systems: Message transfer system: Abstract service definition and procedures
- Rec. X.413: Message handling systems: Message store: Abstract service definition
- Rec. X.419: Message handling systems: Protocol specifications
- Rec. X.420: Message handling systems: Interpersonal messaging system

2.3 *Control procedures*

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

2.4 *ASN.1 coding*

- Rec. X.208: Specification of abstract syntax notation one (ASN.1)
- Rec. X.219: Remote operation

2.5 *Address*

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

2.6 *Character repertoires*

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

2.7 *Intercommunication*

- Rec. F.422: Intercommunication between Teletex service and IPM service.
- Rec. F.203: Network based storage for the Teletex service.

3 Definitions

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

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

4 Abbreviations

ASN.1	Abstract syntax notation one
AU	Access unit
C	Conditional/consumer
CDC	Command document continue
CF	Conversion facility
CSCC	Command session change control
CSS	Command session start
DN	Delivery status notification
DS	Document storage
G3	Group 3 facsimile
G4	Group 4 facsimile
ID	Identity
IP	Interpersonal
IPM	Interpersonal messaging
IPMAS	Interpersonal messaging abstract service
IPME	Interpersonal messaging environment
IPMS	Interpersonal messaging system

IPM-UA	Interpersonal messaging user agent
IPN	Interpersonal notification
M	Mandatory
MS	Message store
MT	Message transfer
MTA	Message transfer agent
MTAS	Message transfer abstract service
MTS	Message transfer system
NDN	Non-delivery status notification
NL	New line
NRN	Non-receipt notification
O/R	Originator/receipt
PDAU	Physical delivery access unit
PTTXAU	Public Teletex access unit
P5	Telematic access protocol
RN	Receipt status notification
S	Supplier
TAPDU	Telematic access protocol data unit
TIA	Telematic interworking application
TID	Terminal identification
TLM	Telematic
TLMA	Telematic agent
TLMAU	Telematic access unit
TLM-TER	Telematic terminal
TLXAU	Telex access unit
TTX	Teletex
UA	User agent

5 Conventions

This Recommendation uses the descriptive conventions identified below.

5.1 *ASN.1*

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

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

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

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

5.2 *Grade*

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

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

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

6 Overview of telematic access to IPMS

6.1 *Abstract model*

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

6.2 *Functional model*

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

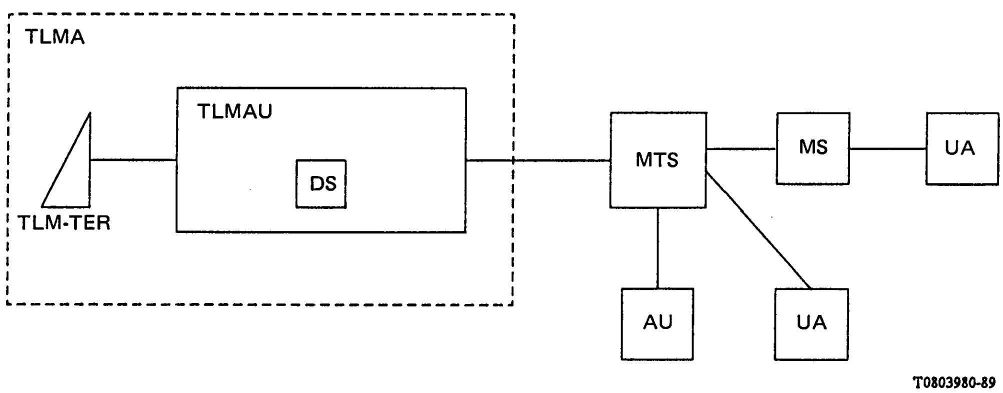


FIGURE 1/T.330

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.

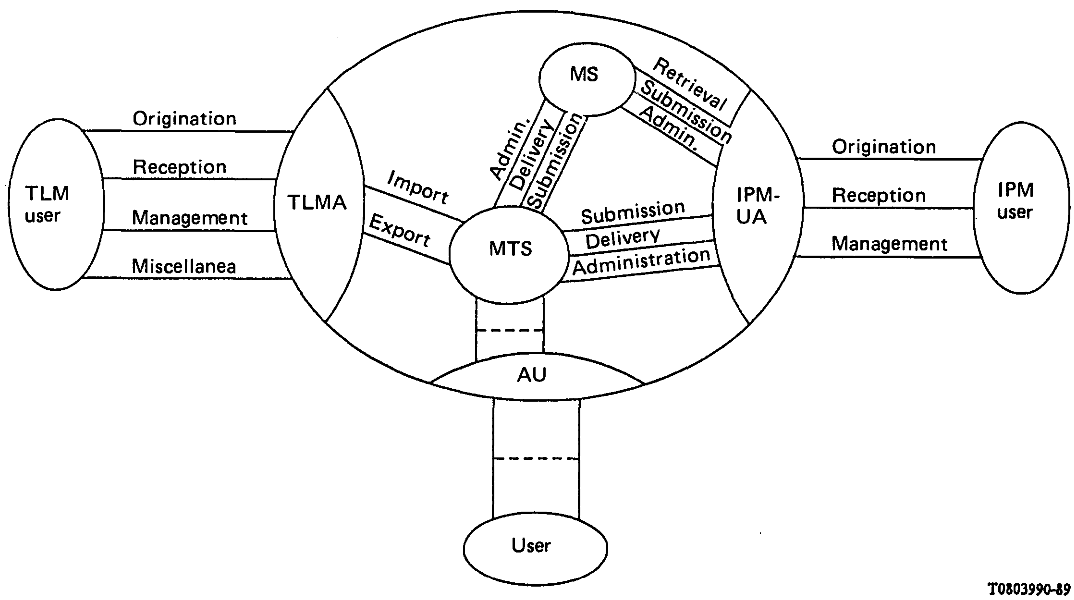


FIGURE 2/T.330

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,
                            DSList,
                            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.

```
tlma-refinement REFINE tlma AS
  tlmau        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.

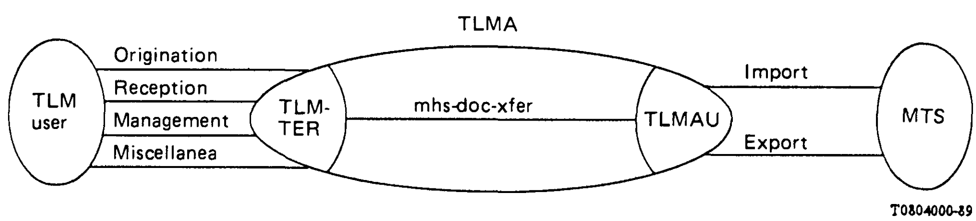


FIGURE 3/T.330

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 *DSList*

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

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

8.2.10 *DSDelete*

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

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

8.2.11 *DSFetch*

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

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

8.2.12 *MessageStatus*

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

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

9 **Abstract errors**

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

a) Subscription error

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

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

b) Name error

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

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

c) Cancellation error

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

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

d) Subscription profile error

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

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 TLM ⇆ TLMAU
Operation	TAPDU name	TLMAU status	
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	→ ← ←
DSList	(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 (remarque 1) C (remarque 1) C (remarque 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 erminal. The OutputMessage-TAPDU is only applicable in the case that DS is subscribed.

10.1.1.1 *Send-TAPDU*

The Send-TAPDU comprises following elements:

Send-TAPDU

```
Send-TAPDU ::= SEQUENCE {
    [0] SEQUENCE {
        send [0] SendTAPDUId,
            [1] SEQUENCE {
                quantityOfDocs          QuantityOfDocsElementId,
                  number-of-docs           NumberOfAssociatedDocuments } OPTIONAL },
        -- See Note 1
    [1] SET {
            [0] SEQUENCE {
                priority                PriorityElementId,
                  priority-ind             PriorityValue DAFAULT normal } OPTIONAL,
            [1] SEQUENCE {
                perMessageIndicators    PerMessageIndicatorsElementId,
                            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              ConversionElementId,
                  conversion-info          ConversionInfoValue } OPTIONAL,
            [3] SEQUENCE {
                contentinfo             ContentInfoElementId,
                  content-return-request   ContentReturnRequestValue } OPTIONAL,
            [4] SEQUENCE {
                returnAddress           ReturnAddressElementId,
                  postal-address           PostalAddressValue OPTIONAL,
            [5] SEQUENCE {
                latestDelivery          LatestDeliveryElementId,
                  latest-delivery-time     DateandTime } OPTIONAL },
            [6] SEQUENCE {
                to                      ToElementId,
                            SET OF SEQUENCE {
                  primary-recipient   [0] ORDescriptor,
                                      [1] RecOptions } } OPTIONAL,
        -- See Note 2
            [7] SEQUENCE {
                cc                      CCElementId,
                            SET OF SEQUENCE {
                  copy-recipient      [0] ORDescriptor,
                                      [1] RecOptions } } OPTIONAL,
        -- See Note 2
            [8] SEQUENCE {
                bcc                     BCCElementId,
                            SET OF SEQUENCE {
                  blind-copy-recipient [0] ORDescriptor,
                                       [1] RecOptions } } OPTIONAL },
        -- See Note 2
```

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             LanguagueInd } 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 ControlInfo 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 following elements:

SendAck-TAPDU

```
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 } } }
```

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 following elements:

Exception-TAPDU

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

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 following elements:

Probe-TAPDU

```
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.
```

10.1.2.2 *ProbeAck-TAPDU*

The ProbeAck-TAPDU comprises following elements:

ProbeAck-TAPDU

```
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 } } }
```

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 following elements:

ExplicitRN-TAPDU

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

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 following elements:

Deliver-TAPDU

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

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 ReceiptStatuNotice-TAPDU comprises 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 } }
            conversion-prohibited [1] ConversionProhibitedValue OPTIONAL } } 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            SuplementaryInformation } 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 DeliveryStatuNotice-TAPDU comprises 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                   SuplementaryInformation } 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,
                      diagostic-code [1] DiagnosticCodeValue OPTIONAL } } } OPTIONAL },
contentReturned [3] ContenReturnedElementId 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 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,
      -- For further study
        [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        SuplementaryInformation } 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 following elements:

RegisterAck-TAPDU		
RegisterAck-TAPDU ::=		
	registerAck	RegisterAckTAPDUId

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 following elements:

DSQuery-TAPDU		
DSQuery-TAPDU ::=		
	dsQuery	DSQueryTAPDUId

10.1.9.2 *DSReport-TAPDU*

The DSReport-TAPDU comprises following elements:

DSReport-TAPDU

```
DSReport-TAPDU ::= SEQUENCE {
    [0]
                    dsReport              DSReportTAPDUId,
    [1] SET OF {
                [0] SEQUENCE {
                 retrievalId              RetrievalIdElementId,
                   retrieval-id              RetrievalIdentifier },
                [1] SEQUENCE {
                 messageType              MessageTypeElementId,
                   message-type              MessageTypeValue },
                [2] SEQUENCE {
                 originator               OriginatorElementId,
                   originator-name           ORName } OPTIONAL,
                [3] SEQUENCE {
                 priority                 PriorityElementId,
                   priority-ind              PriorityValue DEFAULT normal } OPTIONAL,
                [4] SEQUENCE {
                 messageLength            MessageLengthElementId,
                   message-length            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 following elements:

MessageDelete-TAPDU

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

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 following elements:

OutputRequest-TAPDU

```
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 } }
```

10.1.11.2 *OutputMessage-TAPDU*

The OutputMessage-TAPDU comprises following elements:

OutputMessage-TAPDU

```
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,
          DeliveryStatusNotice and ReceiptStatusNotice-TAPDU. The actual components to be used
          depend upon the MessageType parameter value specified in the MessageType component. } }
```

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 *StarusQuery-TAPDU*

The StatusQuery-TAPDU comprises following elements:

StatusQuery-TAPDU

```
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
```

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 following elements:

StatusReport-TAPDU

```
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 {                    -- In case of DN, 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 } OPTIONAL },
                             [3] SEQUENCE {               -- In case of DN, this set
                                                          -- of element shall be present.
               nonDeliveryReason         NonDeliveryReasonElementId,
                                SET {
                 reason-code      [0] ReasonCodeValue,
                 diagnostic-code [1] DiagnosticCodeValue } OPTIONAL } } OPTIONAL } },
```

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 (remarque 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-TAPDUs.

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	–	–	set to "disclosure-of-recipient-prohibited"
conversion-prohibited	–	–	set to "conversion-prohibited"
alternate-recipient-allowed	–	–	set to "alternate-recipient-prohibited"
content-return-request	–	–	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 *ReceiptStatus Notice*

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 receive RN/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-ipm-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 DeliveryStatusNotice-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	
convertedInfoTypes	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 *MessageStarus*

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-, StatusRepor-t-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 mechine-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 this must be present.
Elements ::= SEQUENCE {
                        ElementId,
                        ElementValues }
ElementId ::= SEQUENCE {
                        ElementNumber OPTIONAL,
                        ElementName OPTIONAL }
    -- One of this 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 embeded space 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	21:□ 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	60:□ AUTO-OUTPUT:	
bcc	Constructor	24:□ BCC:	
bodyType	Constructor	31:□ BODY-TYPE:	
cancel	–	3.13:□ CANCEL:	
cc	Constructor	23:□ CC:	
comment	Primitive	50:□ COMMENT:	
contentIndicator	Constructor	18:□ CONTENT-INDICATOR:	
contentInfo	Primitive	17:□ CONTENT-INFO:	
contentReturned	–	72:□ CONTENT-RETURNED-INDICATION:	
conversion	Primitive	16:□ CONVERSION:	
conversionIndication	Constructor	42:□ CONVERSION-INDICATION:	
convertedInfoTypes	Primitive	44:□ CONVERTED-INFORMATION-TYPES:	
correlationInfo	Primitive	1:□ CORRELATION-INFORMATION:	
deleteAfterOutput	Primitive	80:□ DELETE-AFTER-OUTPUT:	
deliver	–	3.3:□ DELIVER:	
deliveryId	Primitive	35:□ DELIVERY-ID:	
deliveryStatusNotice	–	3.4:□ DELIVERY-STATUS-NOTICE:	
dsMode	Primitive	58:□ DS-MODE:	
dsQuery	–	3.7:□ DS-QUERY:	
dsReport	–	3.8:□ DS-REPORT:	
errors	Primitive	9:□ ERRORS:	
exception	–	3.12:□ EXCEPTION:	
expiredDiscard	Primitive	73:□ EXPIRED-DISCARD:	
explicitRN	–	3.6:□ EXPLICIT-RN:	
explicitRNAck	–	3.16:□ EXPLICIT-RN-ACK:	
forwardedInfo	Constructor	32:□ FORWARDED-INFO:	
from	Primitive	20:□ FROM:	

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

TABLE 4/T.330 *(end)*

TAPDUId and ElementId name	Type	T.61 Character coding format	Remarks
relatedIPMs	Constructor	28:□ RELATED-IPMS:	
repliedToIPM	Primitive	30:□ REPLIED-TO-IPM:	
reply	Constructor	25:□ REPLY:	
reportedMessageId	Primitive	75:□ REPORTED-MESSAGE-ID:	
reportedRecipient	Primitive	3:□ REPORTED-RECIPIENT:	
retrievalId	Primitive	38:□ RETRIEVAL-ID:	
returnAddress	Primitive	36:□ RETURN-ADDRESS:	
send	–	3.1:□ SEND:	
sendAck	–	3.14:□ SEND-ACK:	
statusQuery	–	3.19:□ STATUS-QUERY:	
statusReport	–	3.20:□ STATUS-REPORT:	
subject	Primitive	26:□ SUBJECT:	
subjectIPM	Primitive	71:□ SUBJECT-IPM:	
submissionId	Primitive	65:□ SUBMISSION-ID:	
submissionTime	Primitive	33:□ SUBMISSION-TIME:	
supplInfo	Primitive	68:□ SUPPLEMENTARY-INFORMATION:	
supplReceiptInfo	Primitive	68:□ SUPPLEMENTARY-INFORMATION:	
thisIPM	Primitive	27:□ THIS-IPM:	
thisRecipient	Primitive	39:□ THIS-RECIPIENT:	
timeOfDelivery	Primitive	4:□ TIME-OF-DELIVERY:	
timeOfReceipt	Primitive	47:□ TIME-OF-RECEIPT:	
timeOfReport	Primitive	84:□ TIME-OF-REPORT:	
tLMAUOperation	Constructor	59:□ TLMAU-OPERATION:	
to	Constructor	22:□ TO:	
typeOfReceipt	Primitive	48:□ TYPE-OF-RECEIPT:	
typeOfUA	Primitive	45:□ 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/T.330

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	Allowed	
authorizing	Constructor	–	21:□AUTHORIZING:	
authorizing-user	Constructor	General	*OR Descriptor*	
auto-acknowledgment	Primitive	Predefined	Auto-Receipt, Manual-Receipt (défaut)	
auto-forwarded	Primitive	Predefined	Auto-forwarded, Not-Auto-forwarded (default)	
auto-fwd-Comment	Primitive	General	*any Text*	
auto-fwd-ipms	Primitive	Predefined	Auto-forwarded, Not-Auto-forwarded (default)	
auto-fwd-recipient-name	Constructor	General	*OR Name*	
blind-copy-recipient	Constructor	General	*OR Descriptor*	
body-part	Primitive	Predefined	IA5Text, TLX, Voice, G3Fax, G4Fax-Class1, TTX, Videotex, Message, Mixed-Mode, Encrypted	
call-id	Primitive	General	*Call Identification*	
comments	Primitive	General	*Comments*	
content-return-request	Primitive	Predefined	Content-Return-Request	
conversion-info			NO, Yes; WLOSS	
copy-recipient	Constructor	General	*OR Descriptor*	
deferred-delivery-time	Primitive	General	*Date and Time*	
delete-after-output	Primitive	Predefined	Keep, Delete	
delivery-msg-id	Primitive	General	*Message Identifier*	
	Primitive	General	*Date and Time*	
diagnostic-code [a)]	Primitive	Predefined	Unrecognized-OR-Name, Ambiguous-OR-Name, MTS-Congestion, Loop-Detected, Recipient-Unavailable, Maximum-Time-Expired, Content-Too-Long, Conversion-Impractical, Encoded-Information-Type-Unsupported, Conversion-Prohibited, Invalid-Arguments, Implicit-Conversion-Not-Subscribed, Content-Syntax-Error, Pragmatic-Constraint-Violation, Protocol-Violation, Content-Not-Supported, Too-Many-Recipient, No-Bilateral-Agreement	

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

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

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	*OR Name*	
report-time	Primitive	General	*Date and Time*	
report-type	Primitive	Predefined	Receipt, Non-Receipt, Delivery, Non-delivery	
requested-delivery-method	Primitive	Predefined	"RDL="ANY (défaut), MAS, PD, TLX, TTX, G3, G4, IA5, VTX	
retrieval-id	Primitive	General	*Retrieval Identifier*	
return-request	Primitive	Predefined	Return-Request	
rn-request	Primitive	Predefined	RN-Request	
sensitivity	Primitive	Predefined	Personal, Private, Company-Confidential	
status	Primitive	Predefined	In-Process, Delivered, Non-Delivered	
subject-content	Primitive	General	*Subject*	
subject-ipm-id	Primitive	General	*Message Identifier*	
submission-msg-id	Primitive	General	*Message Identifier*	
submissionTime	Primitive	–	33:□SUBMISSION-TIME:	
suppl-info	Primitive	General	*Supplementary Information*	
suppl-receipt-info	Primitive	General	*Supplementary Information*	
terminal-type	Primitive	Predefined	"TTyp=" TLX, TTX, G3, G4, IA5, vtx	
this-recipient-name	Constructor	General	*OR Name*	
this-ipm-id	Constructor	General	*IPM Identifier*	
type-of-receipt	Primitive	Predefined	Manual (défaut), Automatic	
type-of-ua	Primitive	Predefined	Private, Public (défaut)	
user-report-request	Primitive	Predefined	No-Report, Non-Delivery-Report, Report	

[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>|"FN="<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>	Défaut: ADMD de la TLMAU
	Network adress	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 listne		<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			Remarks
Name	Parameter name	Code [a)]	Format/Keyword	
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			Remarks
Name	Parameter name	Code a)	Format/Keyword	
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 = 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-vaue. 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 receipient 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.62. 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 data	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 documents 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) implicity, 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)   ipms(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, ReportDeliveryEnvelope,
        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)   adsdc(2)   modules(0)   notation(1) }

DateandTime ::= 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,
                DSList,
                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  --
```

```
-- 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] NumberOfAssociatedDocuments 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
```

ANNEX B

(to Recommendation T.330)

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],[nrn-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:□ OBSOLETED-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-TPDU*

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]

]

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

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	Constructeur	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	

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:□OBSOLETED-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 indentification		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 No applicable

X Applicable

IMAGING PROCESS OF CHARACTER INFORMATION ON FACSIMILE APPARATUS

(Melbourne, 1988)

0 Introduction

This Recommendation defines the page formatting characteristics of the information conversion process, for imaging character-information on a facsimile apparatus.

1 Scope and field of application

This Recommendation intends to specify the method for positioning in a correct manner character structured messages, in facsimile pages, conforming to the format defined in Recommendation T.4 Group 3 facsimile apparatus, and Recommendation T.563 for facsimile Group 4 apparatus. The intention is to insure the printing of the information inside the guaranteed printable area of the facsimile pages.

This Recommendation specify in terms of facsimile picture elements (pels), the initial position and spacing of the characters along the lines and the spacing between the lines, for creating a facsimile page from the character structured information. It does not define neither the shape, nor the repertoire of the characters to be imaged, which is of the responsibility of the application.

Different types of character encoded information, to be printed on both Group 3 and Group 4 facsimile apparatus, are considered, namely teletex, telex, videotex and IA 5 text.

Consideration of mixed mode information requires further study.

This Recommendation applies when interworking implying image conversion between different type of information, including facsimile, is required within a regulated service, such as MHS service.

2 References

- Recommendation F.1: Operational provisions for the international public telegram service.
- Recommendation S.5: Standardization of page-printing start-stop equipment and cooperation between page-printing and tape-printing start-stop equipment (ITA No. 2).
- Recommendation T.4: Standardization of Group 3 facsimile apparatus for document transmission.
- Recommendation T.50: International Alphabet No. 5.
- Recommendation T.60: Terminal equipment for use in the Teletex service.
- Recommendation T.61: Character repertoire and coded character sets for the international Teletex service.
- Recommendation T.563: General aspects of Group 4 facsimile apparatus.
- Recommendation X.408: Message handling systems: Encoded information type conversion rules.

3 Definitions

See the Recommendations quoted in reference.

4 Basic principles

The figures given in this Recommendation, in terms of number of picture elements and facsimile lines, refer to the standard values defined for the resolution in Recommendation T.4 for Group 3 facsimile apparatus (1728 pels/215 mm and 3.85 lines/mm), and in Recommendation T.563 for Group 4 facsimile apparatus (1728 pels/219.46 mm and 200 lines/25.4 mm).

These figures are defined relatively to the transmitted facsimile lines.

The reference position of the first character line is defined using the base line and home position concepts, as defined in Recommendation T.60, Annexes A and B.

5 Imaging of Teletex messages

The figures of this Recommendation are derived from the figures given in Recommendation T.60 for the basic printable area of Teletex, with the same capability of printing a defined number of characters in the left margin, relatively to the home position.

The Teletex pages, to be imaged on facsimile equipments, conform to the character repertoire and character sets defined in Recommendation T.61 and to the pages formats defined for the basic printable area in Recommendation T.60.

5.1 *Reference position*

Table 1/T.351 defines the position of the 5th base line and of the home position, relatively to which is referred the first line of written text, according to Recommendation T.60.

Both horizontal and vertical orientations of the paper are considered. For the horizontal orientation, it is assumed that the printing on the facsimile equipment is performed, starting from the left side of the written text.

TABLE 1/T.351

Reference position for printing Teletex pages

	Home position		5th base line	
	Horizontal orientation (fax line)	Vertical orientation (pel)	Horizontal orientation (pel)	Vertical orientation (fax line)
Fax G3	79th	205th	1474th	82th
Fax G4	148th	205th	1474th	164th

5.2 *Character spacing*

From the figures of Table 1/T.351, the first character position on each line is defined; the following characters along the line are positioned according to a character spacing value. Table 2/T.351 defines the values to be used for imaging the characters along the lines, depending on the possible value of the spacing defined for the Teletex page format.

TABLE 2/T.351

Values of character spacings

	Character spacing value		
	Defined for Teletex (mm)	Horizontal orientation (fax line)	Vertical orientation (pels)
Fax G3	2.54 2.12 1.69	10 8 7	20 16 14
Fax G4	2.54 2.12 1.69	20 16 14	20 16 14

5.3 *Line spacing*

The position of the first character line in the facsimile page is defined from the figures of Table 1/T.351; the following character line are positioned according to a line spacing value. Table 3/T.351 defines the values to be used for imaging the characters lines on the facsimile page, depending on the line spacing value defined for the Teletex page format.

TABLE 3/T.351

Values of line spacing

Line spacing value			
	Defined for Teletex (mm)	Horizontal orientation (pels)	Vertical orientation (fax lines)
Fax G3	4.23 6.35 8.47 3.175 5	32 48 64 24 38	16 24 32 12 19
Fax G4	4.23 6.35 8.47 3.175 5	32 48 64 24 38	32 48 64 24 38

5.4 *Superscript/subscript*

In the Teletex message, the superscript and subscript functionalities, result in an offset in the printing position of the concerned characters. The allowed offset value for imaging the Teletex pages on the facsimile page is defined in Table 4/T.351.

TABLE 4/T.351

Offset values for supercript/subscript

Offset value			
	Defined for Teletex (mm)	Horizontal orientation (pels)	Vertical orientation (fax lines)
Fax G3	2.12	16	8
Fax G4	2.12	16	16

6 Imaging of Telex messages

The telex messages conform to the character set defined in Recommendation F.1, and to the format defined in Recommendation S.5.

When imaged on a facsimile equipment, the telex messages are divided in pages of 55 text lines, as a maximum.

Only the vertical orientation of the paper is to be considered.

The first character of each text line in a page is positioned according to the home position defined in Table 1/T.351.

The first line is referred to the 5th base line, as defined in Table 1/T.351.

The character spacing values to be used are those defined in Table 2/T.351 for the 2.54 mm spacing.

The line spacing values to be used are those defined in Table 3/T.351 for the 4.23 mm spacing.

7 Imaging of videotex messages

For further study.

8 Imaging of IA 5 text messages

IA 5 messages conform to the character set defined in Recommendation T.50.

When imaged on a facsimile equipments, IA 5 messages are formatted in pages of 55 lines of 80 characters.

Only the vertical orientation of the paper is to be considered.

The first character of each text line in a page is positioned according to the home position defined in Table 1/T.351.

The first line is referred to the 5th base line, as defined in Table 1/T.351.

The character spacing values to be used are those defined in Table 2/T.351 for the 2.12 mm spacing.

The line spacing values to be used are those defined in Table 3/T.351 for the 4.23 mm spacing.

Recommendation T.390

TELETEX REQUIREMENTS FOR INTERWORKING WITH THE TELEX SERVICE

(Malaga-Torremolinos, 1984)

1 General

1.1 *Scope*

1.1.1 Recommendation F.200 lays down the provisions for the operation of the automatic international Teletex service. In particular, Recommendation F.200 defines the basic service requirements and principles for interworking between Teletex and telex services via a conversion facility (CF).

1.1.2 Recommendation T.60 defines the requirements for terminal equipment used in the international Teletex service, including the interworking with telex terminals, and the conversion table to the telex repertoire in case of interworking.

1.1.3 Recommendation T.61 defines the character repertoire and the coded character sets for the Teletex service.

1.1.4 Recommendation T.62 defines the end-to-end control procedures to be used within the Teletex service as well as between a Teletex terminal and a Teletex/telex conversion facility (CF).

1.1.5 Recommendation T.70 defines the network-independent basic transport service applicable to Teletex terminals, as well as between a Teletex terminal and a CF.

1.1.6 This Recommendation defines the requirements additional to Recommendation T.62 for use between a Teletex terminal and a CF to provide interworking between Teletex and telex services, using store-and-forward principles, in the following cases:

a) when the CF and Teletex terminal are in the same country, with message transfer in either direction;

b) when the CF and Teletex terminal are in different countries, with message transfer in the CF to Teletex terminal direction. This is subject to bilateral agreement.

Note 1 – Due to practical operational difficulties such as charging, message transfer in the direction from a Teletex terminal to a CF in another country is not considered in this Recommendation.

Note 2 – The requirements of interconnection of CFs for the international interworking between Teletex and telex services are for further study.

Note 3 – The real-time conversion operation is not considered in this Recommendation.

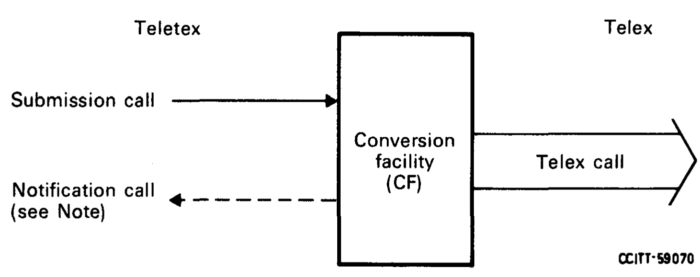
1.1.7 The provisions in this Recommendation are independent of the means and procedures for communication between the CF and the telex network. These are covered in Recommendation F.200.

1.2 *Basic Teletex/telex interworking model*

1.2.1 As illustrated in Figure 1/T.390 and in the single-message case, the communications between a CF and a Teletex terminal for the transfer of a message from Teletex to telex may consist of two calls, namely:

a) a submission call initiated by a Teletex terminal to transfer to the CF a message prepared in the telex mode as per Recommendations F.200 and T.60, and related control information;

b) a notification call initiated by the CF in which delivery or non-delivery information may be provided. This call is mandatory if the telex call is unsuccessful but optional if the telex call succeeds.



Note — A notification call is mandatory if the telex call is unsuccessful but optional if the telex call succeeds (see § 4.3.1).

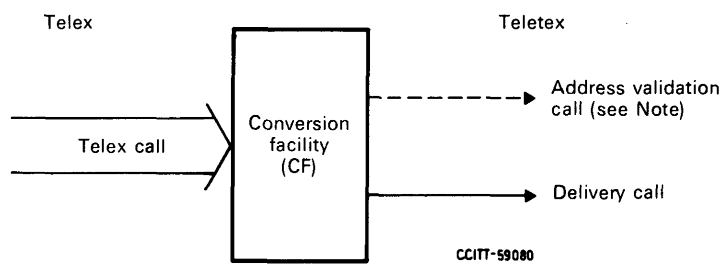
FIGURE 1/T.390

Interworking model with message transfer from Teletex to telex

1.2.2 As illustrated in Figure 2/T.390, the communications between a CF and a Teletex terminal for the transfer of a message from telex to Teletex may consist of two calls, namely:

a) an address validation call initiated by the CF to validate the called Teletex number. The requirement for this call is a national matter;

b) a delivery call initiated by the CF to transfer to the Teletex terminal a message received from the telex terminal.



Note — The requirement for this call is a national matter (see Figure 2/F.201).

FIGURE 2/T.390

Interworking model with message transfer from telex to Teletex

1.2.3 For transferring additional control information between the Teletex terminal and the CF, *control documents* are used. A control document may be associated with one or more normal documents. This set will be called a *message.* All these normal documents should be in telex mode and shall be delivered by the CF as a single *telex message.* The control document need not be prepared in telex mode. An acknowledgement control document will refer to a message and not to individual documents.

2 Session elements of procedure

2.1 The session elements of procedure shall be in accordance with Recommendation T.62. However, the following qualifications shall apply:

2.1.1 The values of mandatory parameters used during session establishment shall be as given in Table 1/T.390.

2.1.2 The value of the "terminal identifier of the calling terminal" parameter in the CSS of the delivery call depends on whether the CF and Teletex terminal are in the same or different countries.

2.1.2.1 When the CF and Teletex terminal are in the same country, it may be either the terminal identification of the CF or, alternatively, the terminal identifier derived from the calling telex answerback (A/B) in one of the following forms in which optional fields are enclosed in parentheses:

8 – = <A/B>

8 <TDC> – = <A/B>

8 <TDC> – <A/B number>(– <A/B machine identity>)=(<A/B subscriber>)

Note 1 – <A/B> is the value of the answerback received from the telex terminal after the deletion of the CR LF characters, if present.

Note 2 – <TDC> is the value of the Recommendation F.69 "telex destination code" of the originating network.

Note 3 – <A/B number> is the subscriber number part of the received <A/B>, as described in Note 1, which conforms to Recommendation F.60.

Note 4 – <A/B machine identity> is the optional machine identity part of the received <A/B>, as described in Note 1, which conforms to Recommendation F.60.

Note 5 – <A/B subscriber name> is the optional subscriber name part of the received <A/B>, as described in Note 1, which conforms to Recommendation F.60. This may also include the telex network identification code.

TABLE 1/T.390

Mandatory parameter values during session establishment

T.62 parameters	Call type			
	Submission	Notification	Address validation	Delivery
Terminal identifier of the calling terminal (in CSS)	TTX's TID	CF's TID	CF's TID	See § 2.1.2
Terminal identifier of the called terminal (in RSSP)	CF's TID	TTX's TID	TTX's TID	TTX's TID
Date and time	Submission call originating date and time by TTX	Notification call originating date and time by CF	Address validation call originating date and time by CF	Delivery call originating date and time by CF
Service identifier	Teletex	Teletex	Teletex	Teletex

TID Terminal Identification
TTX Teletex Terminal
CF Conversion facility

2.1.2.2 When the CF and Teletex terminal are in different countries, it shall be the terminal identification of the CF in accordance to the format specified in Recommendation F.201. The calling telex answerback shall be contained in the text of the normal document in the format and at the point at which it was obtained by the CF.

3 Document elements of procedures

3.1 *General*

3.1.1 The document elements of procedure shall be in accordance with Recommendation T.62. In addition, the Teletex terminal and the CF shall be able to handle control documents defined in this Recommendation as shown in Table 2/T.390.

4 Control documents

4.1 For transferring additional control information between the Teletex terminal and the CF, control documents are used. The additional control information is contained in the user information part of the control document and is called "control text". A summary of the control documents is given in Table 2/T.390.

TABLE 2/T.390

Control documents for Teletex/telex interworking service

Type of call (see Figures 1/T.390 and 2/T.390)	Control document	Status of control document	
		CF	Teletex terminal
Teletex to telex message transfer			
Submission	Telex submission	Mandatory	Mandatory
Notification	Telex delivery notification (Note 2)	Mandatory	Selectable
	Telex non-delivery notification (Note 2)	Mandatory	Mandatory
Telex to teletex message transfer			
Address validation	Telex validation (Note 1)	Optional	Optional
Delivery	Telex message delivery	Optional	Optional

Mandatory: Shall always be implemented.

Optional: Shall be implemented according to national requirements.

Selectable: To be implemented as appropriate.

Note 1 – An address validation call need not contain a telex validation control document.

Note 2 – At present, this only relates to a single telex address.

4.2 *Telex submission control document*

4.2.1 The Telex submission control document shall be used by the Teletex terminal to indicate to the CF that the associated subsequent normal document(s) is to be transferred to a telex terminal.

4.2.2 Elements of the control text are:

a) control document identifier – mandatory;

b) submission control information – optional.

This element consists of the following parameters:

i) telex address – optional.

This is the address of the recipient. It need not be present if provided in lower layer procedures;

ii) answerback – optional.

This is the expected telex answerback. It is to be provided if automatic check by the CF is required;

iii) acknowledgement request – optional.

This is the request for telex delivery notification. This parameter is only present if the user requires advice of a successful telex call. It need not be present in cases where the CF always provides delivery notification.

4.2.3 This control document may be utilized in multi-addressing by providing several sets of the submission control information element in § 4.2.2 above. The provision of this capability in a CF is a national matter.

Note – The inclusion of Teletex addresses within the multi-addressing list is for further study.

4.2.4 Multiple telex submission control documents (each with associated normal documents) may be used within the same session. The provision of this capability in a CF is a national matter.

4.3 *Telex delivery notification control document*

4.3.1 If the Teletex user requires notification after the successful message transfer to the telex terminal [see § 4.2.2 b)], then the telex delivery notification control document shall be sent by the CF to the originator of the telex message. As a national matter, some CFs may always provide this delivery notification.

4.3.2 Elements of the control text are:

a) control document identifier – mandatory;

b) correlation information – mandatory.

This provides a unique reference to the corresponding telex submission control document. The element parameters which are provided by the CF are a national matter. These parameters are:

– CF TID

This is the terminal identification of the CF to which the corresponding telex submission control document was sent.

– TTX TID

This is the terminal identification of the Teletex terminal which sent the telex submission control document.

– Date and time

This is the date and time of the submission call.

– CD No.

This is the document reference number of the telex submission control document.

– Add'l session Ref No.

This is the additional session reference number if used in both the CSS and RSSP in the submission call;

c) submitted control information – mandatory.

This reflects the relevant parameters of the telex submission control document pertaining to a single address. These parameters are:

– telex address

Note – This parameter can be derived from lower layer procedures if not present in the telex submission control document.

– answerback

– acknowledgement request;

d) delivery information – mandatory.

This provides information concerning call establishment from CF to the called telex terminal. The element parameters which are provided by the CF are a national matter. These parameters are:

– telex address

This is the address derived from the telex submission control document and used by the CF to establish the call.

Note – This parameter can be derived from lower layer procedures if not present in the telex submission control document.

– received answerback

This is the complete telex answerback as received by the CF;

e) time of delivery – optional.

This is the time at which the CF delivered the telex message to the telex terminal;

f) telex transmission duration – optional;

g) note – optional.

This is used to convey additional information;

h) received recorded message – optional.

This is used to convey to the Teletex terminal any recorded message from the telex destination.

4.4 *Telex non-delivery notification control document*

4.4.1 The telex non-delivery notification control document shall be used in the following cases:

a) if a telex message cannot be delivered;

b) if a telex message was only partially delivered;

c) if a message is incompletely received in the CF and this partial message successfully delivered (see § 6.1.2).

4.4.2 Elements of the control text are:

a) control document identifier – mandatory;

b) correlation information – mandatory [see § 4.3.2 b)];

c) submitted control information – mandatory [see § 4.3.2 c)];

d) delivery information – mandatory.

This provides information concerning call attempt or call establishment. [For details of the parameters, see § 4.3.2 d)];

e) time of delivery – optional.

This is the time at which the CF delivered the partial telex message to the telex terminal if a partial delivery was achieved;

f) telex transmission duration – optional.

This may be provided if a partial delivery was achieved;

g) last page delivered – optional.

This shall identify the last page number, and its document reference number, successfully transmitted to telex;

h) failure cause – mandatory.

The cause may be one of the following examples:

– service signal from telex network;

– clearing before call connect;

– wrong answerback;

– clear or break during message transmission;

– submitted normal document not in the telex mode;

i) note – optional [see § 4.3.2 g)];

j) received recorded message – optional [see § 4.3.2 h)].

4.5 *Telex validation control document*

4.5.1 The telex validation control document is used by the CF to indicate to the called Teletex terminal that a message from telex will subsequently be sent from the CF. The use of this control document is a national matter (e.g. for unique message identification) and shall not be allowed over international connections.

4.5.2 Elements of the control text are:

a) control document identifier – mandatory;

b) reference – mandatory. This reference is assigned by the CF.

4.6 *Telex message delivery control document*

4.6.1 The telex message delivery control document is used by the CF to indicate to the Teletex terminal that the associated subsequent normal document was received from a telex terminal. The use of this control document is a national matter.

Note – The use of this control document over international connections requires further study.

4.6.2 Elements of the control text are:

a) control document identifier – mandatory;

b) reference – optional.

This reference is assigned by the CF. If the telex validation control document is used, then this reference must be quoted;

c) received time – mandatory.

This contains the time at which the telex was received by the CF;

d) received telex answerback – optional.

This is the complete telex answerback as received by the CF;

e) Note – optional [see § 4.3.2 g)].

5 General rules for control documents

5.1 *Control document utilization*

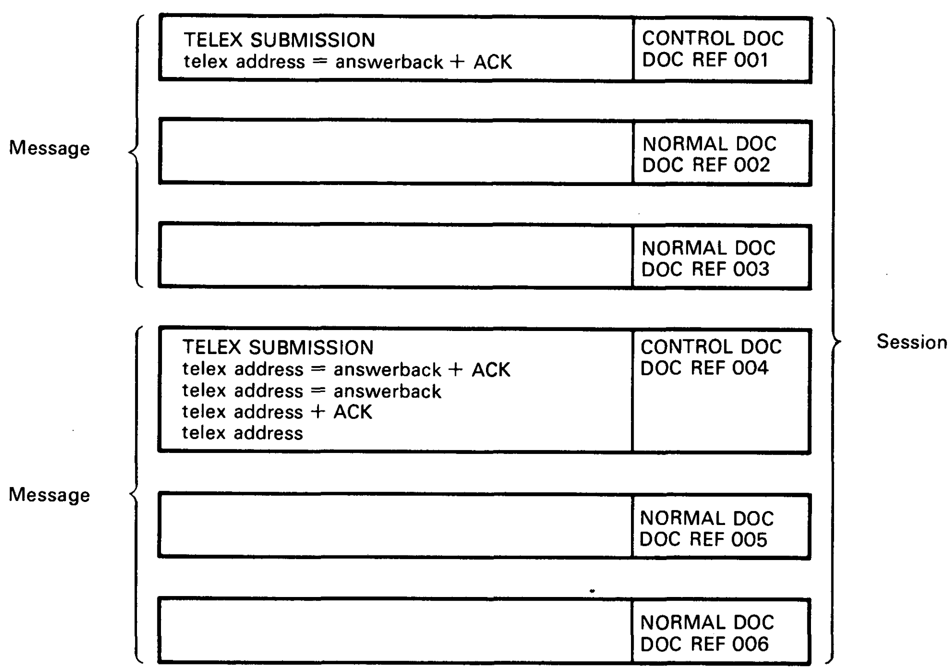
Teletex terminals shall be capable of constructing messages, each consisting of a control document linked with a sequence of normal documents. The message ends at either the start of the next control document, or at the normal end of the session.

5.1.1 The Teletex terminal will allow the following types of communication to a CF within a single session:

- one or more normal documents covered by the same control document ("normal" documents shall be in the "telex" mode);
- one or more addresses covered by the same control document. The provision of this capability in a CF is a national matter;
- more than one control document and related normal documents in the same session. The provision of this capability in a CF is a national matter.

5.1.2 When interworking with telex, the following rules shall apply (see Figure 3/T.390):

- a control document relates to the normal documents which follow it, up to the next control document (if any) within the session, or up to the normal end of the session;
- the presence of more than one address in a control document indicates multi-addressing of all related normal documents;
- within a session, document reference numbers will be assigned, as specified in Recommendation T.62, without distinction between control documents and normal documents.



CCITT-59090

FIGURE 3/T.390

Examples of types of communication (not exhaustive)

6 Error recovery in Teletex/telex interworking

6.1 *Message transfer from Teletex to telex*

6.1.1 During a session, the CF shall perform automatic linking using the procedures specified in T.62.

6.1.2 In the case where a session interruption or CF memory overflow occurs, the following shall apply:

a) All complete messages received and positively acknowledged in that session shall be handled by the CF as in error-free operation.

b) The CF shall handle the interrupted message in one of the following ways:

- The CF shall forward to the telex terminal all positively acknowledged documents and pages of the interrupted document, with an appropriate explanatory text.

  At the end of the text part which is transferred to telex an explanatory text is added, for example:

  "This is an incomplete telex message which may be continued later with the following reference information:

  <rearranged TTX terminal ID> <date and time>"

  At the end of the telex call, the CF may optionally transfer a telex non-delivery notification control document indicating the last page number, and its document reference number, which was successfully transferred to the telex network.

- The CF shall not forward the interrupted message, but shall return a telex non-delivery notification control document to the Teletex terminal to indicate that the complete message has to be retransmitted to the CF.

c) Additionally, in the case of the interrupted message:

- the reaction of the CF is independent of the acknowledgement request of the telex submission control document;
- if a CF in case of interrupted message always sends a telex non-delivery notification, a Teletex terminal or operator should not retransmit any part of that message before receiving that notification.

6.1.3 When the CF is not able to forward the telex message (e.g. because of line interruption) to the telex network, it shall transfer to the Teletex terminal a telex non-delivery notification control document.

6.1.4 If the Teletex terminal retransmits an interrupted message in a new session, it shall send as the first document the control document relating to the interrupted message. When the Teletex operator wants to indicate to the telex terminal that this transmission is a continuation of an interrupted one, the Teletex terminal shall always start the interrupted normal document with CDC. When delivering this message to the telex network, the CF shall precede the continuation of the transmitted message with an appropriate explanatory text, for example "This is a continuation of an incomplete message with the following reference information:

<rearranged TTX terminal ID> <date and time>"

Note – Both parameters of the reference information are taken from the linking information of the CDC command and are identical to the parameters described in § 6.1.2 b).

6.2 *Message transfer from telex to Teletex*

6.2.1 For error conditions arising during the Teletex call, normal error recovery procedures in Recommendation T.62 shall apply.

6.2.2 If an error was detected during telex input, the CF shall add to the end of the interrupted telex message an appropriate explanatory text. An example is "This may be an incomplete message".

This explanatory text may also be used if the telex call was cleared before the CF received the end of input signal (EOI).

7 Coding and formatting of control documents

7.1 *Control document structure*

7.1.1 Figure 4/T.390 illustrates the structure of control documents. The user information part of the control document user information (CDUI) in control documents is called *control text.*

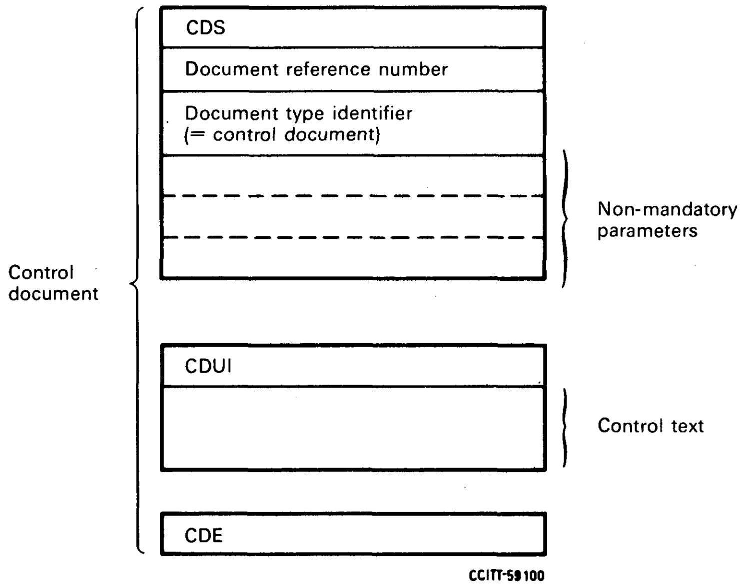


FIGURE 4/T.390

Structure of control documents

7.2 *Control document coding principles*

7.2.1 For the basic Teletex/telex interworking service, the control text shall be coded using explicit, human-readable graphic characters of Recommendation T.61 coding scheme.

7.3 *General formatting of control text*

7.3.1 The control text is subdivided in *elements* each consisting of a number of *fields.*

7.3.2 An element is uniquely identifiable by an *element number* field, which is language-independent, and an *element name* field, which is language dependent. These two fields are always closed by the character "colon" (:). In control documents sent by the CF to the Teletex terminal, at least one of the above fields must be present. For international communications, the element number field shall be mandatory.

7.3.2.1 The element number assigned to the control document identifier shall consist of two parts separated by a period. Each part is a sequentially assigned number. The first part identifies the application which uses the control document. The second part identifies the control document.

7.3.2.2 Each distinct element name, other than control document identifiers, shall be sequentially assigned a different number.

7.3.2.3 There shall be no restriction on the number of digits for element numbers. Any leading zeros in an element number are ignored.

7.3.2.4 Where national requirements dictate the use of non-standardized element numbers, Administrations may choose any value in the range 1000-1999 for the first part of non-standardized control document identifiers and for non-standardized elements.

7.3.3 Parameters to the elements shall be coded in separate *element value* fields.

The parameters will be illustrated in this Recommendation by enclosing them between angular brackets (< >).

7.3.4 In the text each element is contained in one or more lines. When more than one line is used, only the element number field and the element name field shall be present in the first line. See Figure 5/T.390.

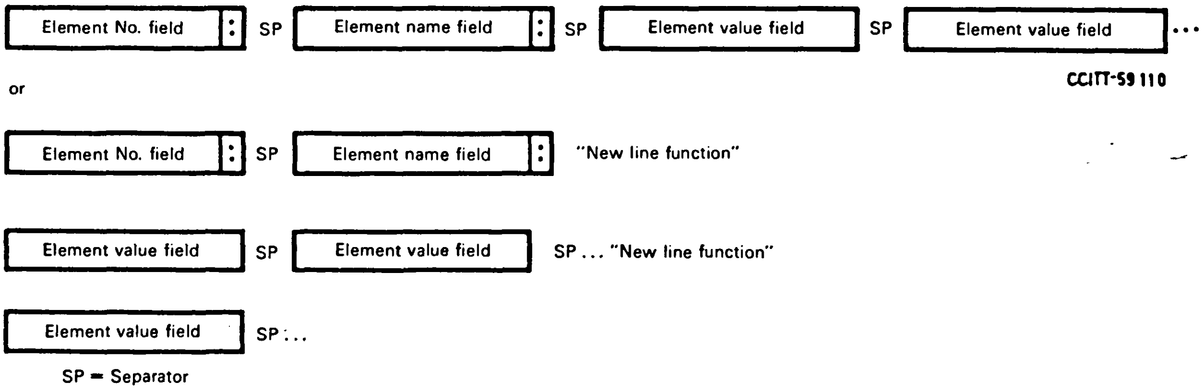


FIGURE 5/T.390

Formats of the elements

7.3.5 The first element present in the control text shall be the control document identifier. This element is mandatory and used to identify the function of the control document, e.g. submission of a telex message, acknowledgement of a previous message.

7.3.6 The element denoted by the name "Note" will be allowed for extending the format to accommodate national requirements.

7.3.7 An element name shall be represented by a *text string*, that is, a sequence of graphic characters. Some parameters may also consist of such text strings. All text strings are language-dependent.

7.3.8 When encoding the control text the following rules shall apply:

a) The control text shall always begin with the character sequence FF CR or CR FF preceded by applicable presentation control functions (see Recommendation T.61, § 3.3.1.4).

b) Each element following the control document identifier may be preceded by one or more empty lines.

c) Each field in an element shall be separated by the character "space" if no other special separator is defined for the element.

Note – It is allowed to have leading spaces or backspaces in a line.

7.3.9 When decoding the control text the following rules shall apply:

a) The first element in the text shall be recognized as the control document identifier. Leading New Line functions (CR LF or LF CR), LF and leading spaces shall be ignored.

b) Contiguous New Line functions (CR LF or LF CR) or LF shall be considered as *one* New Line function.

c) Contiguous embedded spaces are considered as one space. Leading spaces in a line shall be ignored.

Note – Leading backspaces in a line shall be ignored.

7.4 *Format of Teletex/telex interworking control documents*

In this section the formats of the different control documents are shown. Their control texts are illustrated by the use of four different syntax elements:

- The element number field is represented by a sequence of numeric graphic characters.
- 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.

– Separators [i.e.: (SP13), / (SP12), = (SA04)] are shown as they shall be represented in the actual control text.
– Element value fields are shown in angular brackets (< >). The actual parameter values are described in § 7.5.

The formats are illustrated in terms of layout and contents, but do not explicitly show the presentation control functions. These shall be inserted as appropriate (see § 7.3.8).

7.4.1 *Telex submission control document format*

7.4.1.1 Figure 6/T.390 illustrates the format of the control text in the telex submission control document.

```
1.1:    TELEX SUBMISSION:
        <TELEX ADDRESS> <=ANSWERBACK> <+ACKNOWLEDGEMENT REQUEST>
```

FIGURE 6/T.390

Telex submission control document

7.4.1.2 The submission control information element does not have an explicit identification.

7.4.2 *Telex delivery notification control document format*

7.4.2.1 Figure 7/T.390 illustrates the format of the control text in the telex delivery notification control document.

```
1.2:    TELEX DELIVERY NOTIFICATION:
1:      CORRELATION INFORMATION:
        <CF TID> <TTX TID> <DATE AND TIME> <CD No.> <ADD'L SESSION REF No.>
2:      SUBMITTED CONTROL INFORMATION:
        <TELEX ADDRESS> <=ANSWERBACK> <+ACKNOWLEDGEMENT REQUEST>
3:      DELIVERY INFORMATION:
        <TELEX ADDRESS> = <RECEIVED ANSWERBACK>
4:      TIME OF DELIVERY: <DATE AND TIME>
5:      TELEX TRANSMISSION DURATION: <DURATION>
6:      NOTE: <TEXT>
7:      RECEIVED RECORDED MESSAGE:
        <TEXT>
```

Note – If <CD No.> is 5 digits long, and <ADD'L SESSION REF No.> is used, the correlation information line will be 73 characters long. A backspace into the left margin is then necessary, in order not to exceed the printable area of a basic Teletex terminal.

FIGURE 7/T.390

Telex delivery notification control document

7.4.3 *Telex non-delivery nofitication control document format*

7.4.3.1 Figure 8/T.390 illustrates the format of the control text in the telex non-delivery notification control document.

1.3: TELEX NON-DELIVERY NOTIFICATION:

1: CORRELATION INFORMATION:

<CF TID> / <TTX TID> / <DATE AND TIME> / <CD No.> / <ADD'L SESSION REF No.>

2: SUBMITTED CONTROL INFORMATION:

<TELEX ADDRESS> <=ANSWERBACK> <+ACKNOWLEDGEMENT REQUEST>

3: DELIVERY INFORMATION:

<TELEX ADDRESS> = <RECEIVED ANSWERBACK>

4: TIME OF DELIVERY: <DATE AND TIME>

5: TELEX TRANSMISSION DURATION: <DURATION>

8: LAST PAGE DELIVERED: DOCUMENT = <DOC No.> PAGE = <PAGE No.>

9: FAILURE CAUSE: <CAUSE>

6: NOTE: <TEXT>

7: RECEIVED RECORDED MESSAGE:

<TEXT>

Note – See Note of Figure 7/T.390.

FIGURE 8/T.390

Telex non-delivery notification control document

7.4.4 *Telex validation control document format*

7.4.4.1 Figure 9/T.390 illustrates the format of the control text in the telex validation control document.

1.4: TELEX VALIDATION:

10: REFERENCE: <REFERENCE>

FIGURE 9/T.390

Telex validation control document

7.4.5 *Telex message delivery control document format*

7.4.5.1 Figure 10/T.390 illustrates the format of the control text in the telex message delivery control document.

1.5:	TELEX MESSAGE DELIVERY:
10:	REFERENCE: <REFERENCE>
11:	RECEIVED TIME: <DATE AND TIME>
12:	RECEIVED TELEX ANSWERBACK: <RECEIVED ANSWERBACK>
6:	NOTE: <TEXT>

FIGURE 10/T.390

Telex message delivery control document

7.5 *Description of parameter values in Teletex/telex interworking control documents*

7.5.1 <TELEX ADDRESS>

A sequence of numeric graphic characters. As a national option, a limited set of graphic characters may be allowed to be embedded in this field as punctuation marks.

7.5.2 <= ANSWERBACK>

An equals sign graphic character followed by a sequence of alphanumeric graphic characters which comprise part or all of the expected answerback code. The carriage return and line feed characters shall not occur in this parameter.

7.5.3 <+ ACKNOWLEDGEMENT REQUEST>

A plus sign graphic character followed by the text string "ACK".

7.5.4 <CF TID>

A sequence of graphic characters as defined in Figure 2/F.200.

7.5.5 <TTX TID>

A sequence of graphic characters as defined in Figure 2/F.200.

7.5.6 <DATE AND TIME>

A sequence of graphic characters as defined in Figure 1/F.200.

7.5.7 <CD No.>

A document reference number as specified in Recommendation T.62.

7.5.8 <ADD'L SESSION REF No.>

An additional session reference number as specified in Recommendation T.62.

7.5.9 <RECEIVED ANSWERBACK>

A sequence of graphic characters representing a telex answerback code from which CR LF characters have been deleted, if present.

7.5.10 <DURATION>

A sequence of graphic characters in the form "HH : MM : SS" representing the numeric values in hours, minutes and seconds.

7.5.11 <TEXT>

Any text string coded as in Recommendation T.61.

7.5.12 <DOC No.>

A document reference number as specified in Recommendation T.62 with a length not exceeding five digits.

7.5.13 <PAGE No.>

A page reference number as specified in Recommendation T.62 with a length not exceeding five digits.

7.5.14 <CAUSE>

This is a text string with the following possible values – the list is not exhaustive:

- network service signal as received from the telex network (see Recommendation U.12) or/and a corresponding text string in the appropriate language;
- error in submitted control document;
- invalid telex address;
- not delivered – interruption in the transmission;
- submitted text not prepared for telex;
- not permitted telex answerback;
- procedural error during the submission call.

7.5.15 <REFERENCE>

An alphanumeric text string.

ISBN 92-61-03631-7