

Annex C

Data Syntax II for international interactive Videotex service

(This annex forms an integral part of this Recommendation)

NOTE – This data syntax generally corresponds to the (CEPT T/CD 06-01) presentation layer data syntax adopted by some European countries.

General

C.1 Introduction

VIDEOTEX systems are text communication systems with 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.

Videotex services will be provided in different ways in different countries. The Videotex services may be a distributed network of independent computers or a hierarchy of computers with external databases or a mixture of both. It is probable that in all countries Videotex terminals will primarily access the Videotex services via the switched telephone network, over which data is transmitted to a terminal which generates displays. Three types of display have been identified and are described and defined in this Recommendation:

- 1) Alphamosaic;
- 2) Geometric;
- 3) Photographic.

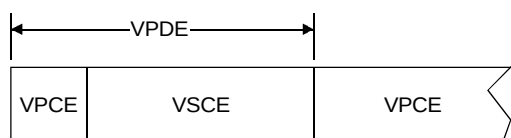
Other types of display may be defined in the future. Each type of display may be used simultaneously, though data for each type of display is separated during transmission. The way in which data is used to generate a display may be modified by 'management data'. Management data may affect more than one type of display.

C.1.1 Coding principles

C.1.1.1 Identification of Data Types

Different types of display data and management data are separated into different 'Videotex Presentation Data Elements' (VPDEs) during transmission.

'Videotex Presentation Data Elements' (VPDEs) are made up of two parts: 'Videotex Presentation Control Element' (VPCE) which identifies the type of data and 'Videotex Service Control Element' (VSCEs) which contains the data.



T0821650-95/d87

VPCEs are coded in the form US X where X is a character from:

- columns 4-7 for alphamosaic data;
- column 2 for management data;
- column 3 for other data.

The following VPCEs have been provisionally assigned:

TERMINAL FACILITY IDENTIFIER	US 2/0 and US 2/1
Define DRCS	US 2/3
Define COLOUR	US 2/6
Define FORMAT	US 2/13
TIMING CONTROL	US 2/14
RESET	US 2/15
ALPHAMOSAIC display data	US <any character from column 4-7>
Reserved ³⁾	US 3/0
GEOMETRIC display data (3D)	US 3/1
GEOMETRIC display data (2D)	US 3/2
PHOTOGRAPHIC pixel data	US 3/4
PHOTOGRAPHIC table data	US 3/5
SOUND	US 3/11
Reserved ⁴⁾	US 3/12
TELESOFTWARE (Processable) data	US 3/14
TRANSPARENT data	US 3/15

US is the UNIT SEPARATOR control and is coded 1/15.

C.1.1.2 Use of default values

Where data fields are used to describe parameters of the following data (e.g. the DRCS header) default values for these fields have been assigned. If the data field is not transmitted, then the terminal will apply the default value.

It is anticipated that some terminals will only be able to process data which conforms to these default values, to simplify their operation these terminals may ignore such data, unless the parameters describing that data are omitted (implying that the data conforms to the default).

It is therefore recommended that when a parameter is equal to the default value that field is not transmitted; if it is transmitted then the response of the terminal is not guaranteed.

C.1.2 Display principles

C.1.2.1 Defined display area

The defined display area is a rectangular area of the screen within which the text and pictorial information is displayed.

The ratio of the width to the height (aspect ratio) of this area should be 4:3.

For the alphamosaic display this area is composed of a defined number of rows each with a defined number of character positions. The default is 24 rows of 40 character positions.

For the geometric display the bottom left-hand corner of the defined display area is addressed as (0, 0) and the upper right-hand corner is defined as (1, 0.75).

The photographic display area also maps to the same area. The top left-hand pixel of the photographic display area is addressed as (1, 1).

The possibility of defining different aspect ratios is for further study.

C.1.2.2 Display structure

The theoretical structure of the display consists of the following layers in order of precedence:

- 1) Alphamosaic character foreground and background layers (see Part 1, C.1.2);
- 2) Geometric layer or layers (see Part 2);

³⁾ US 3/0 is reserved for Geometric display data.

⁴⁾ US 3/12 is reserved for private use.

- 3) Photographic layer (see Part 3);
- 4) Full screen background layer (see Part 1);
- 5) Any other video source.

Changing the display structure is for further study.

C.2 References

- ETS Standard 300 072, *Terminal Equipment (TE) – Videotex Presentation Layer protocol – Videotex Presentation Layer data Syntax*.
- ETS Standard 300 073, *Terminal Equipment (TE) – Videotex Presentation Layer protocol – Videotex Presentation Layer data Syntax – Geometric display*.
- ETS Standard 300 074, *Terminal Equipment (TE) – Videotex Presentation Layer protocol – Videotex Presentation Layer data Syntax – Transparent data*.
- ETS Standard 300 075, *Terminal Equipment (TE) – Videotex processable data*.
- ETS Standard 300 076, *Terminal Equipment (TE) – Videotex – Terminal Facility Identifier (TFI)*.
- CCITT Recommendation T.50, *International Alphabet No 5*.
- CCITT Recommendation T.51, *Latin based coded character sets for telematic services*.
- CCITT Recommendation T.52, *Non-latin based coded character sets for telematic services*.
- CCITT Recommendation F.300, *Videotex service*.
- CCITT Recommendation T.100, *International information exchange for interactive videotex*.
- ISO Standard 2022 (Rev.86), *Code Extension Techniques for use with the ISO 7-bit and 8-bit coded character set*.
- ISO Standard 6937, *Information processing – Coded character sets for text communication*.
- ISO Standard 6429.2, *Draft Standard – Additional control functions for character imaging devices*.
- ISO Standard 9036, *Information processing – 7-bit coded Arabic character set for information interchange (ASMO 449)*.
- CEPT Recommendation T/SF 59, *International Videotex Service*.
- Chinese Standard GB 2312-80, *Coded Chinese Graphic Character Set for Information Interchange*.

C.3 Definitions

bit-combination: Bit-combination is an ordered set of bits that represents a character.

border area: Border area is that part of the display screen (visible display) which is outside the defined display area (see Note and Figure C.1 below).

control character: Control character is a control function, the coded representation of which consists of a single bit-combination.

control function: Control function is an action that affects the recording, processing, transmission or interpretation of data. The coded representation of a control function consists of one or more bit-combinations.

defined display area: The defined display area is a rectangular area of the screen within which the text and pictorial information is displayed (see Figure C.1 and C.1.2.1).

NOTE – The default format of defined display area for the alphamosaic option is given in Part 1, C.1.1.2.

graphic code extension: Graphic code extension is the method of encoding graphic characters in excess of those which may be represented by the code combinations of the basic code table. Alternative sets of 94 characters may be designated by means of shift functions.

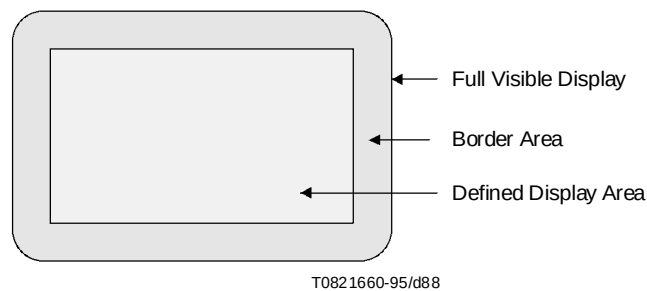


Figure C.1/T.101 – Full Visible Display – Border Area and Defined Display Area

Part 1 – Alphamosaic display

C.1 Description

C.1.1 Introduction

Data sent to the terminal are used to generate alphamosaic displays in which text and graphic characters are displayed, usually in a fixed format of rows and columns.

C.1.1.1 Definitions

active position: Active position is the position on the screen from which subsequent actions would take place if they were activated.

background colour: Background colour is the colour of that area of the character cell not occupied by the foreground colour.

character: Character is a member of a set of elements that is used for organization, control or representation of data. A character repertoire contains two types of elements: graphic characters and control functions.

coded character set: Coded character set is a set of unambiguous rules that establishes a character set and their one-to-one relationship between the characters of the set and the bit-combinations.

code table: Code table is a table showing the character corresponding to each bit-combination in a code. A code table is normally represented as a rectangular matrix of columns and rows.

foreground colour: Foreground colour is the colour of the graphics shape that is being displayed in a character cell.

format effectors: Format effectors are control functions that influence the positioning of text and pictorial images, within the defined display area on a presentation device.

graphic character: Graphic character is a character, other than a control function, that has a visual representation normally printed or displayed.

home position: Home position is the first character position of the first row of the defined display area.

markers: Markers are flags in a memory to show where attribute controls have been set; they are associated with the leading edge of the character position.

parallel attributes: Parallel attributes are the property of the active position and move with it under the action of format effectors or spacing display characters (including space). They apply to the displayed characters subsequently received until the attributes are changed by relevant controls including certain format effectors (CS, APA, APH). They also apply to spacing display characters (including space) inserted by control commands.

serial attributes: Serial attributes are set between markers on a row. They apply from the position of the active position at the time they are received to the end of the row or until a contradictory marker is reached.

wraparound controls: Wraparound controls comprise a set of rules which govern what happens when the active position attempts to move off the defined display area.

C.1.1.2 Format

The default format is 24 rows of 40 columns with automatic wraparound on rows and columns. The format and wraparound may be changed by the 'Define FORMAT' VPDE.

C.1.1.3 Characters

Alphanumeric, block mosaic, smoothed mosaic and line drawing characters are defined. Accented characters are coded using the composition method of coding. The fixed repertoire of characters may be extended with dynamically redefinable characters loaded via the 'Define DRCS' VPDE.

C.1.1.4 Format effectors

Characters may be positioned within the defined display area by means of format effector controls which move the active position, usually in units of one character position.

C.1.1.5 Attributes

The presentation of characters on the screen may be modified by the application of display attributes. Attributes may be applied to the full screen, full row, part of a row (serial) or to subsequently printed characters (parallel).

C.1.1.6 Device Control Functions

The action of scrolling, the display of the cursor and similar functions may be controlled by codes transmitted to the terminal.

C.1.2 Theoretical Terminal Model

The videotex service, alphamosaic option, may be described in the form of an ideally perfect theoretical terminal. This model is detailed hereafter.

C.1.2.1 Description

The theoretical terminal model is based on a separation between the visual content of the page and its structure. It can be described as if it were composed of three memories.

- 1) One character memory where one character address from the character generator is stored at every character location.
- 2) One attribute memory where all the attributes are set in parallel at every location of the screen plus registers for full screen and full row background. The number of registers in this memory is equal to the number of rows plus two. The last two registers refer to the top segment (above the defined display area) and the bottom segment (below the defined display area) of the full screen background.
- 3) One marker memory where every attribute or group of attributes or display functions may be flagged at any character location. When an attribute or function is modified according to the serial mode, this modification occurs between the current character location and the next flag related to this attribute or function (or up to the end of the row).

C.1.2.2 Operation of 'Parallel' and 'Serial' Mode Controls

Both the 'parallel' and 'serial' moded set only serial attributes in the terminal memory (which means that all attributes set, by either mode, are active between markers or up to the end of the row).

Parallel mode controls only apply attributes to the character locations where the cursor prints a character (including space), and remain with the cursor when it moves between rows except when the control codes CS, APA or APH are received. An attribute is copied into the attribute memory and markers are set wherever an attribute is changed. Whenever a continuous string of graphic characters, including SPACE, is written on a row under the parallel mode, then, if there is a change of attribute(s) between adjacent character locations, a marker(s) is created or moved. In addition, any existing markers within the overwritten part of the row are deleted.

Serial mode control codes insert or modify a marker into the marker memory and cause an attribute to be copied immediately into the attribute memory until a contradictory or complementary marker is encountered in the marker memory, or until the end of the row. When in the serial mode, the writing of a graphic character does not modify by itself the attribute in the attribute memory.

Parallel and Serial mode control codes are taken from different control sets and therefore may be unambiguously recognized by the terminal. This is achieved by invoking the appropriate Parallel or Serial C1 set.

The invocation of a Parallel or Serial C1 set will cause the mode of operation of the terminal to switch. Thus in the Serial mode any parallel attributes locked to the cursor will have no effect. Their effect will be restored when the Parallel mode is re-invoked.

Interaction of Serial and Parallel mode control codes: a subsequent (in time) Parallel mode control code will apply to all characters which the cursor writes while in the Parallel mode irrespective of how their attributes had been previously set.

A subsequent (in time) Serial mode control will propagate to the right of the cursor position at which it is received until it meets a contradictory marker.

A full row attribute (other than the background colour) has the effect of overwriting the defined attributes on all the positions of the row and has the effect of deleting all contradictory or complementary attribute markers. The full screen attribute has the same effect but written to all rows it does not delete markers.

C.1.2.3 Layered structure

The alphamosaic display area acts as if it were composed of 2 independent layers:

- A full screen background layer which may be partitioned into rows (with time-dependent precedence).
- A defined display area character layer. The colour of this layer is either BACKGROUND COLOUR or FOREGROUND COLOUR.

As indicated in Part 0 of this Recommendation, geometric and photographic layers may exist between the full screen background layer and the defined display area character layer.

C.1.2.4 Action of Attributes on Layers

The transparent colour in the defined display area character layer (foreground or background) allows see-through to the underlying full screen background layer or the geometric or photographic layers if present.

The character BACKGROUND COLOUR attribute, including the transparent value, applies only to the defined display area character layer.

The full screen and full row BACKGROUND COLOUR attribute affects only the full screen background layer. Its transparent value refers to the video picture.

The full screen, full row and parallel INVERT attribute-controls affect simultaneously and symmetrically the FOREGROUND and the BACKGROUND in the defined display area character layer.

All other full screen or defined display area attributes apply only to the foreground of the defined display area character layer (except the SIZE attribute which also affects the background of this layer). The actions of the combined effect of INVERT and the transparent colour are to be seen in Table C.1.

C.1.3 Defined Attributes and Qualified Areas

C.1.3.1 FOREGROUND COLOUR

This is the colour of the graphics shape being displayed. The colour may be any colour from the available colour tables including 'transparent' in which case the full screen background colour (or the geometric or photographic layers if present) is seen.

Table C.1/T.101 – Truth Table for Attribute Settings

ATTRIBUTE SETTINGS			Colour of resultant display	
			FOREGROUND and underline	BACKGROUND colour
FOREGROUND/BACKGROUND	FOREGROUND	BACKGROUND		
NORMAL	Normal (c)	Normal (b)	c	b
		Transp. (b)	c	a
	Transparent (c)	Normal (b)	a	b
		Transp. (b)	a	a
INVERT	Normal (c)	Normal (b)	b	c
		Transp. (b)	a	c
	Transparent (c)	Normal (b)	b	b
		Transp. (c)	a	a
Foreground of defined display area character layer (c) Background of the defined display area character layer (b) Full screen background layer (a)				

C.1.3.2 BACKGROUND COLOUR

– CHARACTER BACKGROUND COLOUR

This is the colour of the remaining area of the character cell. The colour may be any colour from the available colour tables or be transparent in which case the full screen background colour (or the geometric or photographic layers if present) is seen.

– FULL SCREEN or FULL ROW BACKGROUND COLOUR

This is the colour of layer (a) of the theoretical model (see C.1.2.3).

C.1.3.3 LINED

Alphanumeric characters are displayed with an underline in which the underline is considered to be part of the shape of the graphics character. Mosaic characters and line drawing characters are displayed in separated font (see C.2.1.2).

C.1.3.4 SIZE

There are four states of character size:

– NORMAL-SIZE

The extent of characters occupies the active position.

– DOUBLE-HEIGHT

The extent of characters occupies both the active position and the corresponding position of the adjacent row.

- **DOUBLE-WIDTH**

The extent of characters occupies both the active position and the next position of the same row.

- **DOUBLE-SIZE**

The extent of characters occupies the active position, the next position on the same row and the corresponding two positions on the adjacent row.

See C.1.4 for rules for the application of the SIZE attribute.

C.1.3.5 FLASH

The following attribute states are defined:

- **STEADY**

The characters are displayed normally.

- **FLASHING**

NORMAL FLASH

The characters are displayed alternately in the prevailing foreground colour and in the prevailing background colour.

INVERTED FLASH

This is as for FLASH but on the inverted phase of the flashing clock.

REDUCED INTENSITY FLASH (flash between colour tables)

The characters are displayed alternately in the prevailing foreground colour and in the equivalent colour of another colour table. Table 1 colours adopt table 2 colours, table 2 colours adopt table 1 colours, table 3 colours adopt table 4 colours, and table 4 colours adopt table 3 colours (see C.1.5.3).

States of FLASHING

Each of the above states may be displayed at either of the following rates:

- 50% ON/OFF ratio at about 1 Hz
 - 33% ON, 1st phase
 - 33% ON, 2nd phase
 - 33% ON, 3rd phase
- } at about 2 Hz

C.1.3.6 CONCEAL

The characters are displayed as spaces until the user chooses to make them appear.

C.1.3.7 INVERT

The characters are displayed as if the foreground and background colours had been exchanged. If FLASH is applied, the polarity of the flashing clock is also inverted.

C.1.3.8 WINDOW/BOX

The 'full screen background' of the character positions becomes transparent, i.e. the video picture is displayed.

C.1.3.9 MARKED

The characters are marked for further action at the terminal, e.g. to be transferred to an output device.

C.1.3.10 PROTECTED

The character positions are protected against alteration, manipulation or erasure. The protection is valid for attributes as well as characters.

Protected character positions may only be overwritten by the use of a specific code or by the action of the clear screen command (CS), which deletes both the characters and the protection.

Protected character positions may be scrolled and therefore may disappear from the screen, because the protection is always related to the particular information on the screen.

Protected characters must not be obscured by enlarged characters.

C.1.3.11 Scrolling Area

A scrolling area is an area within the defined display area, within which the characters and associated attributes move in increments of one character position under the action of format effectors or specific controls. The procedure of scrolling is defined by two processes:

- 1) the designation of the screen area inside which a scroll operation is to be executed;
- 2) the execution of the scrolling action.

The border of a scrolling area must not be crossed by an enlarged character. The action of a double-height command in the serial mode on the bottom row causes a scroll up, the writing of a double-height character in the parallel mode on the top row causes a scroll down.

The scrolling operation is applied to full rows; the scrolling of parts of rows is for further study.

The use of APA and APH allows the active positions to be moved across the borders of a scrolling area. The addressing of APA is relative to the defined display area and is independent of scrolling.

C.1.3.12 Colour Tables

Extension of the colour range is accomplished by providing a number of colour tables of 8 colours each. At a given instant only one table may be 'in use'; this table can be invoked using colour table controls.

The fixed repertoire of colours (plus transparent) may be extended with redefinable colours loaded via the 'Define COLOUR' VPDE.

C.1.4 Rules for the Action of the SIZE Attribute

In the parallel mode the application of the double-height control causes characters to be printed so that they occupy the character positions on the current row and on the row immediately above. The origin of the characters for subsequent attribute modification is the upper character position. The double-height and double-size controls are inactive on the top row of the defined display area. The writing of double-height character in the parallel mode on the top row of a scrolling area causes a scroll down.

In the serial mode double-height characters extend downwards, the origin of the character is the upper character position. The double-height and the double-size controls are inactive on the bottom row of the defined display area. The action of a double-height command in the serial mode on the bottom row of a scrolling area causes a scroll up.

Double-width characters extend to the right, the origin of the character is the left-hand character position. Alternate characters on the row are displayed.

The whole of an enlarged character is displayed with the attributes that apply to the origin of the character.

Parts of enlarged characters are not displayed, the double-width and double-size controls are inactive in the last character position of a row.

Attributes set at obscured character positions do not take effect if they would break any of the above rules.

The application of a double-width attribute or a double-size attribute causes the cursor to move two character positions forward in both the serial and parallel modes when a character is written. The action of cursor control functions such as APB, APF and spacing attribute controls is not affected.

The application of one SIZE attribute terminates the action of any other SIZE attribute.

NOTE – Attention is drawn to the fact that the retention of characters obscured by enlarged characters and the overwriting of parts of enlarged characters is for further study.

C.1.5 Defaults

C.1.5.1 Default Initiation

The occurrence of certain events causes the default settings to be set. Table C.2 below shows the events leading to the setting of a certain default. This is independent of the current mode of operation of the terminal.

Table C.2/T.101 – Default Initiation

Default set	Full screen background and device controls	Defined display area background	Defined display area foreground	Cursor parallel attributes	Markers	Colour table and Scrolling area
Session start	x	x	x ^{a)}	x	x	x
CS		x	x ^{a)}	x	x	x
APA				x		
APH				x		
Full Row Attribute controls					x ^{b)}	
CAN					x ^{c)}	
a) Default graphic character is SPACE. b) For the related attribute in the row. c) All the markers on the right side of the active position up to the end of the row.						

C.1.5.2 Default Setting of Attributes

Full screen attributes are used as default conditions for defined display area attributes (see Table C.3).

Table C.3/T.101 – Default Setting of Attributes

Full screen background	Defined display area background and cursor	Defined display area foreground and cursor	Markers	Colour table	Scrolling
Black	Transparent	Colour white Normal size Unboxed Not concealed Steady Non-lined Not inverted Non-protected Unmarked	Off	Colour Table 1	Implicit Scrolling active No defined scrolling area

C.1.5.3 Default Colour Look-up Tables

CLUT 1 addresses colours 0-7 of the colour map.

CLUT 2 addresses colours 8-15 of the colour map.

CLUT 3 addresses colours 16-23 of the colour map.

CLUT 4 addresses colours 24-31 of the colour map.

C.1.5.4 Default Colour Map

See Table C.4.

Table C.4/T.101 – RED, GREEN and BLUE components of Default Colour Map

Colour No.		R	G	B
0	BLACK	000000	000000	000000
1	RED	111111	000000	000000
2	GREEN	000000	111111	000000
3	YELLOW	111111	111111	000000
4	BLUE	000000	000000	111111
5	MAGENTA	111111	000000	111111
6	CYAN	000000	111111	111111
7	WHITE	111111	111111	111111
8 ^{a)}	TRANSPARENT	--	--	--
9	REDUCED INTENSITY RED	011111	000000	000000
10	REDUCED INTENSITY GREEN	000000	011111	000000
11	REDUCED INTENSITY YELLOW	011111	011111	000000
12	REDUCED INTENSITY BLUE	000000	000000	011111
13	REDUCED INTENSITY MAGENTA	011111	000000	011111
14	REDUCED INTENSITY CYAN	000000	011111	011111
15	GREY	011111	011111	011111
16	BLACK	000000	000000	000000
17	RED	111111	000000	000000
18	GREEN	000000	111111	000000
19	YELLOW	111111	111111	000000
20	BLUE	000000	000000	111111
21	MAGENTA	111111	000000	111111
22	CYAN	000000	111111	111111
23	WHITE	111111	111111	111111
24	BLACK	000000	000000	000000
25	RED	111111	000000	000000
26	GREEN	000000	111111	000000
27	YELLOW	111111	111111	000000
28	BLUE	000000	000000	111111
29	MAGENTA	111111	000000	111111
30	CYAN	000000	111111	111111
31	WHITE	111111	111111	111111
a) If this entry (No. 8) is defined as BLACK (as it is for default) it will be interpreted as TRANSPARENT.				

C.1.5.5 Default Device Controls

Cursor – Off

Recording Device – Stop

Hard Copy Device – Stop

Auxiliary Device – Off

Display Device – On

C.1.5.6 Default Graphic Sets

G0 set – The Latin primary set of graphic characters

G1 set – The second supplementary set of mosaic characters

G2 set – The Latin supplementary set of graphic characters

G3 set – The third supplementary set of mosaic characters

L-set – The first supplementary set of mosaic characters

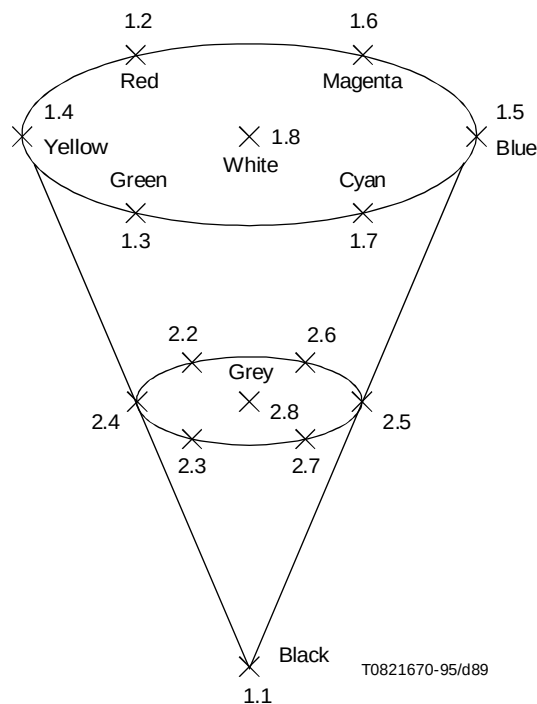


Figure C.2/T.101 – Colour cone

C.2 Repertory

Alphamosaic presentation data is identified by the transmission of the ALPHAMOSAIC VPCE. The data following the VPCE may consist of any of the following repertoire of characters, format effectors, code extension controls, device controls or attribute controls.

C.2.1 Character Repertoire

The character repertoire consists of a fixed repertoire of alphanumeric characters, mosaic characters and line drawing characters. This fixed repertoire may be extended by the use of the DRCS option as described in Part 4.

Characters of the fixed repertoire are identified according to the scheme described in Appendix I.

C.2.1.1 Alphanumeric Characters

The alphanumeric repertoire consists of a fixed repertoire of 232 Latin alphabetic characters, 69 Greek alphabetic characters, 44 Arabic alphabetic characters, 27 Hebrew alphabetic characters, 102 non-alphabetic alphanumeric characters, 204 mosaic characters and over 8000 characters associated with the Chinese language.

Latin alphabetic characters

NOTE – The G0 set is shown in Table C.8. The G2 set is shown in Table C.9.

ID	Graphic	Name or description	Set	Code		
				Position	Set	Position
LA01	a	Small a	G0	6/1		
LA02	A	Capital A	G0	4/1		
LA11	á	Small a with acute accent	G2	4/2	G0	6/1
LA12	Á	Capital A with acute accent	G2	4/2	G0	4/1
LA13	à	Small a with grave accent	G2	4/1	G0	6/1
LA14	À	Capital A with grave accent	G2	4/1	G0	4/1
LA15	â	Small a with circumflex	G2	4/3	G0	6/1
LA16	Â	Capital A with circumflex	G2	4/3	G0	4/1
LA17	ä	Small a with diaeresis or umlaut	G2	4/8	G0	6/1
LA18	Ä	Capital A with diaeresis or umlaut	G2	4/8	G0	4/1
LA19	ã	Small a with tilde	G2	4/4	G0	6/1
LA20	Ã	Capital A with tilde	G2	4/4	G0	4/1
LA23	ḁ	Small a with breve	G2	4/6	G0	6/1
LA24	Ḃ	Capital A with breve	G2	4/6	G0	4/1
LA27	å	Small a with ring	G2	4/10	G0	6/1
LA28	Å	Capital A with ring	G2	4/10	G0	4/1
LA31	ā	Small a with macron	G2	4/5	G0	6/1
LA32	Ā	Capital A with macron	G2	4/5	G0	4/1
LA43	ḃ	Small a with ogonek	G2	4/14	G0	6/1
LA44	Ḃ	Capital A with ogonek	G2	4/14	G0	4/1
LA51	æ	Small æ diphthong	G2	7/1		
LA52	Æ	Capital Æ diphthong	G2	6/1		
LB01	b	Small b	G0	6/2		
LB02	B	Capital B	G0	4/2		
LC01	c	Small c	G0	6/3		
LC02	C	Capital C	G0	4/3		
LC11	ć	Small c with acute accent	G2	4/2	G0	6/3
LC12	Ć	Capital C with acute accent	G2	4/2	G0	4/3
LC15	ċ	Small c with circumflex	G2	4/3	G0	4/1
LC16	Ċ	Capital C with circumflex	G2	4/3	G0	4/1
LC21	č	Small c with caron	G2	4/15	G0	6/3
LC22	Č	Capital C with caron	G2	4/15	G0	4/3
LC29	ċ	Small c with dot	G2	4/7	G0	6/3

LC30	●	Capital C with dot	G2	4/7	G0	4/3
LC41	ç	Small c with cedilla	G2	4/11	G0	6/3
LC42	Ç	Capital C with cedilla	G2	4/11	G0	4/3
LD01	d	Small d	G0	6/4		
LD02	D	Capital D	G0	4/4		
LD21	ḋ or d'	Small d with caron	G2	4/15	G0	6/4
LD22	Ḍ	Capital D with caron	G2	4/15	G0	4/4
LD61	ḏ	Small d with stroke	G2	7/2		
LD62	Ḑ	Capital D with stroke, Icelandic eth	G2	6/2		
LD63	ḑ	Small eth, Icelandic	G2	7/3		
LE01	e	Small e	G0	6/5		
LE02	E	Capital E	G0	4/5		
LE11	é	Small e with acute accent	G2	4/2	G0	6/5
LE12	É	Capital E with acute accent	G2	4/2	G0	4/5
LE13	è	Small e with grave accent	G2	4/1	G0	6/5
LE14	È	Capital E with grave accent	G2	4/1	G0	4/5
LE15	ê	Small e with circumflex	G2	4/3	G0	6/5
LE16	Ê	Capital E with circumflex	G2	4/3	G0	4/5
LE17	ë	Small e with diaeresis or umlaut	G2	4/8	G0	6/5
LE18	Ë	Capital E with diaeresis or umlaut	G2	4/8	G0	4/5
LE21	ě	Small e with caron	G2	4/15	G0	6/5
LE22	Ě	Capital E with caron	G2	4/15	G0	4/5
LE29	ċ	Small e with dot	G2	4/7	G0	6/5
LE30	Ċ	Capital E with dot	G2	4/7	G0	4/5
LE31	ē	Small e with macron	G2	4/5	G0	6/5
LE32	Ē	Capital E with macron	G2	4/5	G0	4/5
LE43	ė	Small e with ogonek	G2	4/14	G0	6/5
LE44	Ę	Capital E with ogonek	G2	4/14	G0	4/5
LF01	f	Small f	G0	6/6		
LF02	F	Capital F	G0	4/6		
LG01	g	Small g	G0	6/7		
LG02	G	Capital G	G0	4/7		
LG11	ǰ	Small g with acute accent	G2	4/2	G0	6/7
LG15	ɡ̸	Small g with circumflex	G2	4/3	G0	6/7
LG16	Ğ	Capital G with circumflex	G2	4/3	G0	4/7
LG23	ḡ	Small g with breve	G2	4/6	G0	6/7
LG24	Ḡ	Capital G with breve	G2	4/6	G0	4/7
LG29	ḡ	Small g with dot	G2	4/7	G0	6/7

LG30	⊕	Capital G with dot	G2	4/7	G0	4/7
LG42	Ɔ	Capital G with cedilla	G2	4/11	G0	4/7
LH01	h	Small h	G0	6/8		
LH02	H	Capital H	G0	4/8		
LH15	ĥ	Small h with circumflex	G2	4/3	G0	6/8
LH16	Ĥ	Capital H with circumflex	G2	4/3	G0	4/8
LH61	ꞥ	Small h with stroke	G2	7/4		
LH62	Ꞧ	Capital H with stroke	G2	6/4		
LI01	i	Small i	G0	6/9		
LI02	I	Capital I	G0	4/9		
LI11	í	Small i with acute accent	G2	4/2	G0	6/9
LI12	Í	Capital I with acute accent	G2	4/2	G0	4/9
LI13	ì	Small i with grave accent	G2	4/1	G0	6/9
LI14	Ì	Capital I with grave accent	G2	4/1	G0	4/9
LI15	î	Small i with circumflex	G2	4/3	G0	6/9
LI16	Î	Capital I with circumflex	G2	4/3	G0	4/9
LI17	ï	Small i with diaeresis or umlaut	G2	4/8	G0	6/9
LI18	Ï	Capital I with diaeresis or umlaut	G2	4/8	G0	4/9
LI19	ɨ	Small i with tilde	G2	4/4	G0	6/9
LI20	Ꝁ	Capital I with tilde	G2	4/4	G0	4/9
LI30	ⱱ	Capital I with dot	G2	4/7	G0	4/9
LI31	ḭ	Small i with macron	G2	4/5	G0	6/9
LI32	Ī	Capital I with macron	G2	4/5	G0	4/9
LI43	ḯ	Small i with ogonek	G2	4/14	G0	6/9
LI44	Ꞩ	Capital I with ogonek	G2	4/14	G0	4/9
LI51	ɶ	Small ij ligature	G2	7/6		
LI52	Ɑ	Capital IJ ligature	G2	6/6		
LI61	Ɱ	Small i without dot	G2	7/5		
LJ01	j	Small j	G0	6/10		
LJ02	J	Capital J	G0	4/10		
LJ15	ĵ	Small j with circumflex	G2	4/3	G0	6/10
LJ16	Ĵ	Capital J with circumflex	G2	4/3	G0	4/10
LK01	k	Small k	G0	6/11		
LK02	K	Capital K	G0	4/11		
LK41	ƙ	Small k with cedilla	G2	4/11	G0	6/11
LK42	Ƙ	Capital K with cedilla	G2	4/11	G0	4/11
LK61	K	Small k, Greenlandic	G2	7/0		
LL01	l	Small l	G0	6/12		

LL02	L	Capital L	G0	4/12		
LL11	Ⓛ	Small l with acute accent	G2	4/2	G0	6/12
LL12	Ⓛ	Capital L with acute accent	G2	4/2	G0	4/12
LL21	l' or Ⓛ	Small l with caron	G2	4/15	G0	6/12
LL22	L' or Ⓛ	Capital L with caron	G2	4/15	G0	4/12
LL41	Ⓛ	Small l with cedilla	G2	4/11	G0	6/12
LL42	Ⓛ	Capital L with cedilla	G2	4/11	G0	4/12
LL61	●	Small l with stroke	G2	7/8		
LL62	Ⓛ	Capital L with stroke	G2	6/8		
LL63	ℓ	Small l with middle dot	G2	7/7		
LL64	ℓ	Capital L with middle dot	G2	6/7		
LM01	m	Small m	G0	6/13		
LM02	M	Capital M	G0	4/13		
LN01	n	Small n	G0	6/14		
LN02	N	Capital N	G0	4/14		
LN11	■	Small n with acute accent	G2	4/2	G0	6/14
LN12	Ⓝ	Capital N with acute accent	G2	4/2	G0	4/14
LN19	ñ	Small n with tilde	G2	4/4	G0	6/14
LN20	Ñ	Capital N with tilde	G2	4/4	G0	4/14
LN21	œ	Small n with caron	G2	4/15	G0	6/14
LN22	Ⓝ	Capital N with caron	G2	4/15	G0	4/14
LN41	Ⓝ	Small n with cedilla	G2	4/11	G0	6/14
LN42	Ⓝ	Capital N with cedilla	G2	4/11	G0	6/14
LN61	⇒	Small eng, Lapp	G2	7/14		
LN62	↑	Capital eng, Lapp	G2	6/14		
LN63	n'	Small n with apostrophe	G2	6/15		
LO01	o	Small o	G0	6/15		
LO02	O	Capital O	G0	4/15		
LO11	ó	Small o with acute accent	G2	4/2	G0	6/15
LO12	Ó	Capital O with acute accent	G2	4/2	G0	4/15
LO13	ò	Small o with grave accent	G2	4/1	G0	6/15
LO14	Ò	Capital O with grave accent	G2	4/1	G0	4/15
LO15	ô	Small o with circumflex	G2	4/3	G0	6/15
LO16	Ô	Capital O with circumflex	G2	4/3	G0	4/15
LO17	ö	Small o with diaeresis or umlaut	G2	4/8	G0	6/15
LO18	Ö	Capital O with diaeresis or umlaut	G2	4/8	G0	4/15
LO19	õ	Small o with tilde	G2	4/4	G0	6/15
LO20	Õ	Capital O with tilde	G2	4/4	G0	4/15

LO25	œ	Small o with double acute accent	G2	4/13	G0	6/15
LO26	Ŏ	Capital O with double acute accent	G2	4/13	G0	4/15
LO31	ō	Small o with macron	G2	4/5	G0	6/15
LO32	Ō	Capital O with macron	G2	4/5	G0	4/15
LO51	æ	Small ligature	G2	7/10		
LO52	Œ	Capital ligature	G2	6/10		
LO61	ø	Small o with slash	G2	7/9		
LO62	Ø	Capital O with slash	G2	6/9		
LP01	p	Small p	G0	7/0		
LP02	P	Capital P	G0	5/0		
LQ01	q	Small q	G0	7/1		
LQ02	Q	Capital Q	G0	5/1		
LR01	r	Small r	G0	7/2		
LR02	R	Capital R	G0	5/2		
LR11	ŕ	Small r with acute accent	G2	4/2	G0	7/2
LR12	Ŗ	Capital R with acute accent	G2	4/2	G0	5/2
LR21	ŗ	Small r with caron	G2	4/15	G0	7/2
LR22	Ŕ	Capital R with caron	G2	4/15	G0	5/2
LR41	ŗ̣	Small r with cedilla	G2	4/11	G0	7/2
LR42	Ŗ̣	Capital R with cedilla	G2	4/11	G0	5/2
LS01	s	Small s	G0	7/3		
LS02	S	Capital S	G0	5/3		
LS11	š	Small s with acute accent	G2	4/2	G0	7/3
LS12	Š	Capital S with acute accent	G2	4/2	G0	5/3
LS15	ș	Small s with circumflex	G2	4/3	G0	7/3
LS16	Ș	Capital S with circumflex	G2	4/3	G0	5/3
LS21	ŝ	Small s with caron	G2	4/15	G0	7/3
LS22	Ŝ	Capital S with caron	G2	4/15	G0	5/3
LS41	ṣ̂	Small s with cedilla	G2	4/11	G0	7/3
LS42	Ṣ̂	Capital S with cedilla	G2	4/11	G0	5/3
LS61	ß	Small sharp s, German	G2	7/11		
LT01	t	Small t	G0	7/4		
LT02	T	Capital T	G0	5/4		
LT21	ţ	Small t with caron	G2	4/15	G0	7/4
LT22	Ț	Capital T with caron	G2	4/15	G0	5/4
LT41	ţ̣	Small t with cedilla	G2	4/11	G0	7/4
LT42	Ț̣	Capital T with cedilla	G2	4/11	G0	5/4
LT61	⤀	Small t with stroke	G2	7/13		

LT62	⌘	Capital T with stroke	G2	6/13		
LT63	✞	Small thorn, Icelandic	G2	7/12		
LT64	⌚	Capital thorn, Icelandic	G2	6/12		
LU01	u	Small u	G0	7/5		
LU02	U	Capital U	G0	5/5		
LU11	ú	Small u with acute accent	G2	4/2	G0	7/5
LU12	Ú	Capital U with acute accent	G2	4/2	G0	5/5
LU13	ù	Small u with grave accent	G2	4/1	G0	7/5
LU14	Ù	Capital U with grave accent	G2	4/1	G0	5/5
LU15	û	Small u with circumflex	G2	4/3	G0	7/5
LU16	Û	Capital U with circumflex	G2	4/3	G0	5/5
LU17	ü	Small u with diaeresis or umlaut	G2	4/8	G0	7/5
LU18	Ü	Capital U with diaeresis or umlaut	G2	4/8	G0	5/5
LU19	𝄞	Small u with tilde	G2	4/4	G0	7/5
LU20	𝄟	Capital U with tilde	G2	4/4	G0	5/5
LU23	𝄢	Small u with breve	G2	4/6	G0	7/5
LU24	Ɱ	Capital U with breve	G2	4/6	G0	5/5
LU25	•	Small u with double acute accent	G2	4/13	G0	7/5
LU26	◌̧	Capital U with double acute accent	G2	4/13	G0	5/5
LU27	𝄠	Small u with ring	G2	4/10	G0	7/5
LU28	⌘	Capital U with ring	G2	4/10	G0	5/5
LU31	·	Small u with macron	G2	4/5	G0	7/5
LU32	Ū	Capital U with macron	G2	4/5	G0	5/5
LU43	ꞥ	Small u with ogonek	G2	4/14	G0	7/5
LU44	Ɑ	Capital U with ogonek	G2	4/14	G0	5/5
LV01	v	Small v	G0	7/6		
LV02	V	Capital V	G0	5/6		
LW01	w	Small w	G0	7/7		
LW02	W	Capital W	G0	5/7		
LW15	ⱥ	Small w with circumflex	G2	4/3	G0	7/7
LW16	ⱦ	Capital W with circumflex	G2	4/3	G0	5/7
LX01	x	Small x	G0	7/8		
LX02	X	Capital X	G0	5/8		
LY01	y	Small y	G0	7/9		
LY02	Y	Capital Y	G0	5/9		
LY11	ÿ	Small y with acute accent	G2	4/2	G0	7/9
LY12	Ỳ	Capital Y with acute accent	G2	4/2	G0	5/9
LY15	ÿ	Small y with circumflex	G2	4/3	G0	7/9

LY16	Ÿ	Capital Y with circumflex	G2	4/3	G0	5/9
LY17	ÿ	Small y with diaeresis or umlaut	G2	4/8	G0	7/9
LY18	Ÿ	Capital Y with diaeresis or umlaut	G2	4/8	G0	5/9
LZ01	z	Small z	G0	7/10		
LZ02	Z	Capital Z	G0	5/10		
LZ11	Ʒ	Small z with acute accent	G2	4/2	G0	7/10
LZ12	Ʒ	Capital Z with acute accent	G2	4/2	G0	5/10
LZ21	ž	Small z with caron	G2	4/15	G0	7/10
LZ22	Ž	Capital Z with caron	G2	4/15	G0	5/10
LZ29	ẏ	Small z with dot	G2	4/7	G0	7/10
LZ30	Ẑ	Capital Z with dot	G2	4/7	G0	5/10

Greek alphabetic characters

NOTE – The G0 set is shown in Table C.13. The G2 set is shown in Table C.9.

ID	Graphic	Name or description	Set	Code		
				Position	Set	Position
GA01	α	Small letter Alpha	G0	6/1		
GA02	Α	Capital letter Alpha	G0	4/1		
GA11	α̂	Small letter Alpha with accent	G2	4/2	G0	6/1
GA12	Α̂	Capital letter Alpha with accent	G2	4/2	G0	4/1
GB01	β	Small letter Beta	G0	6/2		
GB02	Β	Capital letter Beta	G0	4/2		
GG01	γ	Small letter Gamma	G0	6/3		
GG02	Γ	Capital letter Gamma	G0	4/3		
GD01	δ	Small letter Delta	G0	6/4		
GD02	Δ	Capital letter Delta	G0	4/4		
GE01	ε	Small letter Epsilon	G0	6/5		
GE02	Ε	Capital letter Epsilon	G0	4/5		
GE11	ε̂	Small letter Epsilon with accent	G2	4/2	G0	6/5
GE12	Ε̂	Capital letter Epsilon with accent	G2	4/2	G0	4/5
GZ01	ζ	Small letter Zeta	G0	6/6		
GZ02	Ζ	Capital letter Zeta	G0	4/6		
GE61	η	Small letter Eta	G0	6/7		
GE62	Η	Capital letter Eta	G0	4/7		
GE63	η̂	Small letter Eta with accent	G2	4/2	G0	6/7
GE64	Η̂	Capital letter Eta with accent	G2	4/2	G0	4/7
GT61	θ	Small letter Theta	G0	6/8		

GT62	➤	Capital letter Theta	G0	4/8		
GI01	ⱥ	Small letter Iota	G0	6/9		
GI02	ⱦ	Capital letter Iota	G0	4/9		
GI11	Ⱨ	Small letter Iota with accent	G2	4/2	G0	6/9
GI12	ⱨ	Capital letter Iota with accent	G2	4/2	G0	4/9
GI17	Ⱪ	Small letter Iota with diaeresis	G2	4/8	G0	6/9
GI18	ⱪ	Capital letter Iota with diaeresis	G2	4/8	G0	4/9
GI33	ⱬ	Small letter Iota with accent and diaeresis	G2	4/0	G0	6/9
GK01	Ɱ	Small letter Kappa	G0	6/10		
GK02	Ɐ	Capital letter Kappa	G0	4/10		
GL01	Ⱳ	Small letter Lambda	G0	6/11		
GL02	ⱳ	Capital letter Lambda	G0	4/11		
GM01	Ⱶ	Small letter Mu	G0	6/12		
GM02	ⱶ	Capital letter Mu	G0	4/12		
GN01	ⱷ	Small letter Nu	G0	6/13		
GN02	ⱸ	Capital letter Nu	G0	4/13		
GX01	ⱺ	Small letter Xi	G0	6/14		
GX02	ⱻ	Capital letter Xi	G0	4/14		
GO01	ⱽ	Small letter Omicron	G0	6/15		
GO02	Ȿ	Capital letter Omicron	G0	4/15		
GO11	Ɀ	Small letter Omicron with accent	G2	4/2	G0	6/15
GO12	Ⲁ	Capital letter Omicron with accent	G2	4/2	G0	4/15
GP01	ⲁ	Small letter Pi	G0	7/0		
GP02	Ⲃ	Capital letter Pi	G0	5/0		
GR01	ⲃ	Small letter Rho	G0	7/1		
GR02	Ⲅ	Capital letter Rho	G0	5/1		
GS01	Ⲇ	Small letter Sigma	G0	7/3		
GS02	ⲇ	Capital letter Sigma	G0	5/3		
GS03	Ⲉ	Small letter final Sigma	G0	7/2		
GT01	Ⲋ	Small letter Tau	G0	7/4		
GT02	ⲋ	Capital letter Tau	G0	5/4		
GY01	Ⲍ	Small letter Upsilon	G0	7/5		
GY02	ⲍ	Capital letter Upsilon	G0	5/5		
GY11	ⲏ	Small letter Upsilon with accent	G2	4/2	G0	7/5
GY12	Ⲑ	Capital letter Upsilon with accent	G2	4/2	G0	5/5
GY17	ⲑ	Small letter Upsilon with diaeresis	G2	4/8	G0	7/5

GY18	Ⲑ	Capital letter Upsilon with diaeresis	G2	4/8	G0	5/5
GY33	·	Small letter Upsilon with accent and diaeresis	G2	4/0	G0	7/5
GF01	ϕ	Small letter Phi	G0	7/6		
GF02	ϕ	Capital letter Phi	G0	5/6		
GH01	ϣ	Small letter Khi	G0	7/7		
GH02	ϣ	Capital letter Khi	G0	5/7		
GP61	ϙ	Small letter Psi	G0	7/8		
GP62	ϙ	Capital letter Psi	G0	5/8		
GO61	Ω	Small letter Omega	G0	7/9		
GO62	Ω	Capital letter Omega	G0	5/9		
GO63	Ω	Small letter Omega with accent	G2	4/2	G0	7/9
GO64	Ω	Capital letter Omega with accent	G2	4/2	G0	5/9

Arabic alphabetic characters

NOTE – The G0 set is shown in Table C.14.

ID	Graphic	Name or description	Code	
			Set	Position
AH61	ﺀ	Arabic letter hamzah	G0	4/01
	ﺀ	Arabic letter maddah on alef	G0	4/02
	ﺀ	Arabic letter hamzah on alef	G0	4/03
	ﺀ	Arabic letter hamzah on waw	G0	4/04
	ﺀ	Arabic letter hamzah under alef	G0	4/05
	ﺀ	Arabic letter hamzah on ya	G0	4/06
AX33	ﺀ	Arabic letter alef	G0	4/07
AB01	ﺀ	Arabic letter ba	G0	4/08
	ﺀ	Arabic letter ta marbutah	G0	4/09
AT01	ﺀ	Arabic letter ta	G0	4/10
AT61	ﺀ	Arabic letter tha	G0	4/11
AG23	ﺀ	Arabic letter jeem	G0	4/12
AH29	ﺀ	Arabic letter ha	G0	4/13
AH61	ﺀ	Arabic letter kha	G0	4/14
AD01	ﺀ	Arabic letter dal	G0	4/15
AD61	ﺀ	Arabic letter thal	G0	5/00
AR01	ﺀ	Arabic letter ra	G0	5/01
AZ01	ﺀ	Arabic letter zain	G0	5/02

AS01	ﺶ	Arabic letter seen	G0	5/03
AS21	ﺺ	Arabic letter sheen	G0	5/04
AS29	ﺺ	Arabic letter sad	G0	5/05
AD29	ﺪ	Arabic letter dad	G0	5/06
AT29	ﺖ	Arabic letter tah	G0	5/07
AZ29	ﺰ	Arabic letter dhah	G0	5/08
	ﺀ	Arabic letter ain	G0	5/09
AG61	ﻎ	Arabic letter ghain	G0	5/10
SM49	*	Arabic tatweel	G0	6/00
AF01	ﻑ	Arabic letter fa	G0	6/01
AQ01	ﻕ	Arabic letter qaf	G0	6/02
AK01	ﻙ	Arabic letter kaf	G0	6/03
AL01	ﻝ	Arabic letter lam	G0	6/04
AM01	ﻡ	Arabic letter meem	G0	6/05
AN01	ﻥ	Arabic letter noon	G0	6/06
AH01	ﻩ	Arabic letter ha	G0	6/07
AW01	ﻭ	Arabic letter waw	G0	6/08
	ﺀ	Arabic letter alef maqsurah	G0	6/09
AY01	ﻱ	Arabic letter ya	G0	6/10
AA31	ﺀ	Arabic letter fathatan	G0	6/11
AU31	ﺀ	Arabic letter dammatan	G0	6/12
AI31	ﺀ	Arabic letter kasratan	G0	6/13
AA01	ﺀ	Arabic letter fathah	G0	6/14
AU01	ﺀ	Arabic letter dammah	G0	6/15
AI01	ﺀ	Arabic letter kasrah	G0	7/00
	ﻩ	Arabic letter shaddah	G0	7/01
	ﻩ	Arabic letter sukun	G0	7/02

Hebrew alphabetic characters

NOTE – The G0 set is shown in Table C.14.

ID	Graphic	Name or description	Code	
			Set	Position
HX33	ﺀ	Hebrew letter alef	G1	6/00
HB01	ﺀ	Hebrew letter bet	G1	6/01
HG01	ﺀ	Hebrew letter gimel	G1	6/02

HD01	↖	Hebrew letter dalet	G1	6/03
HH01	↗	Hebrew letter he	G1	6/04
HW01	↙	Hebrew letter vav	G1	6/05
HZ01	↘	Hebrew letter zayin	G1	6/06
HH45	←	Hebrew letter het	G1	6/07
HT45	→	Hebrew letter tet	G1	6/08
HY01	↑	Hebrew letter yod	G1	6/09
HK61	↓	Hebrew letter final kaf	G1	6/10
HK01	↖	Hebrew letter kaf	G1	6/11
HL01	↗	Hebrew letter lamed	G1	6/12
HM61	↙	Hebrew letter final mem	G1	6/13
HM01	↘	Hebrew letter mem	G1	6/14
HN61	↩	Hebrew letter final nun	G1	6/15
HN01	↪	Hebrew letter nun	G1	7/00
HS01	↱	Hebrew letter samekh	G1	7/01
HX35	↲	Hebrew letter ayin	G1	7/02
HP61	↔	Hebrew letter final pe	G1	7/03
HP01	↕	Hebrew letter pe	G1	7/04
HS61	↖	Hebrew letter final tsadi	G1	7/05
HS45	↗	Hebrew letter tsadi	G1	7/06
HQ01	↘	Hebrew letter qof	G1	7/07
HR01	↙	Hebrew letter resh	G1	7/08
HS21	◻	Hebrew letter shin	G1	7/09
HT01	◻	Hebrew letter tav	G1	7/10

Non-alphabetic characters

Chinese Ideographic characters

These characters are not independently listed in the repertoire; they are shown only in the code table (see Table C.16).

Decimal digits

ID	Graphic	Name or description	Code	
			Set	Position
ND01	1	Digit 1	G0	3/1
ND02	2	Digit 2	G0	3/2
ND03	3	Digit 3	G0	3/3
ND04	4	Digit 4	G0	3/4

ND05	5	Digit 5	G0	3/5
ND06	6	Digit 6	G0	3/6
ND07	7	Digit 7	G0	3/7
ND08	8	Digit 8	G0	3/8
ND09	9	Digit 9	G0	3/9
ND10	0	Digit 0	G0	3/0

Currency signs

ID	Graphic	Name or description	Code	
			Set	Position
SC01	₪	General currency sign	G2	2/3
			(G0	2/4)
SC02	£	Pound sign	G2	2/3
SC03	\$	Dollar sign	G2	2/4
SC04	¢	Cent sign	G2	2/2
SC05	¥	Yen sign	G2	2/5

Punctuation marks⁵⁾

ID	Graphic	Name or description	Code	
			Set	Position
SP01		Space	Gx	2/0
SP02	!	Exclamation mark	G0	2/1
SP03	¡	Inverted exclamation mark	G2	2/1
SP04	"	Quotation mark	G0	2/2
SP05	'	Apostrophe	G0	2/7
SP06	(⁶⁾	Left parenthesis	G0	2/8
SP07	)	Right parenthesis	G0	2/9
SP08	, ⁷⁾	Comma	G0	2/12
SP10	-	Hyphen or minus sign	G0	2/13
SP11	.	Full stop, period	G0	2/14
SP12	/	Solidus	G0	2/15
SP13	:	Colon	G0	3/10
SP14	; ⁷⁾	Semicolon	G0	3/11
SP15	? ⁷⁾	Question mark	G0	3/15
SP16	¿	Inverted question mark	G2	3/15

⁵⁾ In videotex 'quotation mark', 'apostrophe' and 'comma' are independent characters which cannot have the meaning of diacritical marks.

⁶⁾ The characters in pair like open parenthesis and close parenthesis, greater than and smaller than, square or curly brackets, may have two presentation variants depending on the writing direction. In Latin type of writing, opening character of the pair is the left one while in Arabic way of writing, opening character is the right one.

⁷⁾ The characters 'quotation marks', 'comma' and 'semicolon' may have two presentation variants depending on the writing direction.

SP17	«	Angle quotation mark left	G2	2/11
SP18	»	Angle quotation mark right	G2	3/11
SP19	‘	Single quotation mark left	G2	2/9
SP20	’	Single quotation mark right	G2	3/9
SP21	“	Double quotation mark left	G2	2/10
SP22	”	Double quotation mark right	G2	3/10

Arithmetic signs

ID	Graphic	Name or description	Code	
			Set	Position
SA01	+	Plus sign	G0	2/11
SA02	±	Plus/minus sign	G2	3/1
SA03	< ⁸⁾	Less-than sign	G0	3/12
SA04	=	Equals sign	G0	3/13
SA05	> ⁸⁾	Greater-than sign	G0	3/14
SA06	÷	Divide sign	G2	3/8
SA07	×	Multiply sign	G2	3/4

Subscripts and Superscripts

ID	Graphic	Name or description	Code	
			Set	Position
NS01	¹	Superscript 1	G2	5/1
NS02	²	Superscript 2	G2	3/2
NS03	³	Superscript 3	G2	3/3

Fractions

ID	Graphic	Name or description	Code	
			Set	Position
NF01	$\frac{1}{2}$	Fraction one half	G2	3/13
NF04	$\frac{1}{4}$	Fraction one quarter	G2	3/12
NF05	$\frac{3}{4}$	Fraction three quarters	G2	3/14

⁸⁾ See Footnote 6.

Miscellaneous symbols

ID	Graphic	Name or description	Code	
			Set	Position
SM01	#	Number sign	G0	2/3
			G2	2/6
SM02	%	Percent sign	G0	2/5
SM03	&	Ampersand	G0	2/6
SM04	*	Star	G0	2/10
SM05	@	Commercial at	G0	4/0
SM06	[⁹⁾	Left square bracket	G0	5/11
SM07	\	Reverse solidus	G0	5/12
SM08	] ⁹⁾	Right square bracket	G0	5/13
SM11	{ ⁹⁾	Left curly bracket	G0	7/11
SM12	—	Central horizontal bar jointive	G2	5/0
SM13		Central vertical bar jointive	G0	7/12
SM14	} ⁹⁾	Right curly bracket	G0	7/13
SM17	μ	Micro sign	G2	3/5
SM18	Ω	Ohm sign	G2	6/0
SM19	°	Degree sign	G2	3/0
SM20	♂	Ordinal indicator, masculine	G2	6/11
SM21	♀	Ordinal indicator, feminine	G2	6/3
SM24	¶	Section sign	G2	2/7
SM25	÷	Paragraph sign, pilcrow	G2	3/6
SM26	•	Middle dot	G2	3/7
SM30	←	Leftward arrow	G2	2/12
SM31	→	Rightward arrow	G2	2/14
SM32	↑	Upward arrow	G2	2/13
SM33	↓	Downward arrow	G2	2/15
SM34	■	Delete	Gx	7/15
SM35	®	Registered mark symbol	G2	5/2
SM36	©	Copyright symbol	G2	5/3
SM37	™	Trade mark symbol	G2	5/4
SM38	🎵	Musical symbol	G2	5/5

⁹⁾ See Footnote 6.

SM39		One eighth	G2	5/12
SM40		Three eighths	G2	5/13
SM41		Five eighths	G2	5/14
SM42		Seven eighths	G2	5/15
SM43	^	Arrowhead upwards	G0	5/14
SM44	\	Upper reverse solidus	G0	6/0
SM45		Left vertical bar jointive	G1	4/14
SM46		Right vertical bar jointive	G1	5/14
SM47	—	Upper bar	G0	7/14
SM48	—	Lower bar	G0	5/15

Diacritical marks (as displayed when used in conjunction with SPACE)

ID	Graphic	Name or description	Code	
			Set	Position
SD11	´	Acute accent	G2	4/2
SD13	`	Grave accent	G2	4/1
SD15	^	Circumflex	G2	4/3
SD17	¨	Umlaut or diaeresis	G2	4/8
SD19	~	Tilde	G2	4/4
SD21	̣	Caron	G2	4/15
SD23	◌◌	Breve	G2	4/6
SD25	ˆ	Double acute accent	G2	4/13
SD27	×	Ring	G2	4/10
SD29	˙	Dot	G2	4/7
SD31	◌◌	Macron	G2	4/5
SD33	¨	Diaeresis with acute accent	G2	4/0
SD41	¸	Cedilla	G2	4/11
SD43	◌◌	Ogonek	G2	4/14

Arabic diacritics (as displayed in conjunction with SPACE)

ID	Graphic	Name or description	Code	
			Set	Position
DA10	◌◌	Arabic fathatan	G0	6/11
DA11	◌◌	Arabic dammatan	G0	6/12
DA12	◌◌	Arabic kasratan	G0	6/13
DA13	◌◌	Arabic fathah	G0	6/14
DA14	◌◌	Arabic dammah	G0	6/15

DA15	⇒	Arabic kasrah	G0	7/00
DA16	↑	Arabic shaddah	G0	7/01
DA17	↓	Arabic sukun	G0	7/02

C.2.1.2 Mosaic Graphics

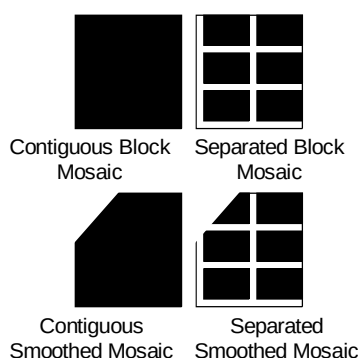
In addition to the alphanumeric repertoire, it is possible to make simple pictures using characters from the mosaic graphic repertoire defined below. Each mosaic character completely fills the area of a character cell on the screen.

The repertoire consists of:

- 63 graphics (block mosaic characters) consisting of a combination of six rectangular elements;
- 48 graphics (smoothed mosaic characters) where the shapes are bounded by lines between corners of six rectangular elements;
- 8 graphics (smoothed mosaic characters) where the shapes are bounded by lines between the corners of the character cell and the centre of the character cell;
- 48 line drawing graphics;
- 4 jointive arrows;
- 13 miscellaneous drawing graphics including one graphic with a dot-pattern where approximately 40% of the character cell area has the foreground colour and the remaining area has the background colour.

The shaded areas in the representations of the mosaic character are to be displayed in the defined foreground colour and the unshaded areas are to be displayed in the defined background colour.

The mosaic graphic character may be displayed in two fonts, 'contiguous' and 'separated' as shown in the examples below. For the 'separated' font the 'separation space' is on the left and lower edge of the blocks. The actual dimension of the space is not defined.



T0821680-95/d90

Block Mosaic Graphics

<i>ID</i>	<i>Graphic</i>	<i>Code</i>	
		<i>Set</i>	<i>Position</i>
MG01		G1	2/1
MG02		G1	2/2
MG03		G1	2/3
MG04		G1	2/4
MG05		G1	2/5
MG06		G1	2/6
MG07		G1	2/7
MG08		G1	2/8
MG09		G1	2/9
MG10		G1	2/10
MG11		G1	2/11
MG12		G1	2/12
MG13		G1	2/13
MG14		G1	2/14
MG15		G1	2/15
MG16		G1	3/0
MG17		G1	3/1
MG18		G1	3/2
MG19		G1	3/3

T0823330-95/d91

<i>ID</i>	<i>Graphic</i>	<i>Code</i>	
		<i>Set</i>	<i>Position</i>
MG20		G1	3/4
MG21		G1	3/5
MG22		G1	3/6
MG23		G1	3/7
MG24		G1	3/8
MG25		G1	3/9
MG26		G1	3/10
MG27		G1	3/11
MG28		G1	3/12
MG29		G1	3/13
MG30		G1	3/14
MG31		G1	3/15
MG32		G1	6/0
MG33		G1	6/1
MG34		G1	6/2
MG35		G1	6/3
MG36		G1	6/4
MG37		G1	6/5
MG38		G1	6/6
MG39		G1	6/7
MG40		G1	6/8
MG41		G1	6/9

T0823340-95/d92

<i>ID</i>	<i>Graphic</i>	<i>Code</i>	
		<i>Set</i>	<i>Position</i>
MG42		G1	6/10
MG43		G1	6/11
MG44		G1	6/12
MG45		G1	6/13
MG46		G1	6/14
MG47		G1	6/15
MG48		G1	7/0
MG49		G1	7/1
MG50		G1	7/2
MG51		G1	7/3
MG52		G1	7/4
MG53		G1	7/5
MG54		G1	7/6
MG55		G1	7/7
MG56		G1	7/8
MG57		G1	7/9
MG58		G1	7/10
MG59		G1	7/11
MG60		G1	7/12
MG61		G1	7/13
MG62		G1	7/14
MG63		G1	5/15

T0823350-95/d93

Smoothed Mosaic Graphics

ID	Graphic	Code	
		Set	Position
SG01		G1	4/0
SG02		G1	4/1
SG03		G1	4/2
SG04		G1	4/3
SG05		G1	4/4
SG06		G1	4/5
SG07		G1	4/6
SG08		G1	4/7
SG09		G1	4/8
SG10		G1	4/9
SG11		G1	4/10
SG12		G1	4/11
SG13		G1	4/12
SG14		G1	4/13
SG15		G1	5/13
SG16		G1	5/12
SG17		G1	5/11

T0821730-95/d94

ID	Graphic	Code	
		Set	Position
SG18		G1	5/10
SG19		G1	5/9
SG20		G1	5/8
SG21		G1	5/7
SG22		G1	5/6
SG23		G1	5/5
SG24		G1	5/4
SG25		G1	5/3
SG26		G1	5/2
SG27		G1	5/1
SG28		G1	5/0
SG29		G3	6/0
SG30		G3	6/1
SG31		G3	6/2
SG32		G3	6/3
SG33		G3	6/4
SG34		G3	6/5
SG35		G3	6/6
SG36		G3	6/7
SG37		G3	6/8

T0821740-95/d95

<i>ID</i>	<i>Graphic</i>	<i>Code</i>	
		<i>Set</i>	<i>Position</i>
SG38		G3	6/9
SG39		G3	6/10
SG40		G3	6/11
SG41		G3	6/12
SG42		G3	6/13
SG43		G3	7/0
SG44		G3	7/1
SG45		G3	7/2
SG46		G3	7/3
SG47		G3	7/4
SG48		G3	7/5
SG49		G3	7/6
SG50		G3	7/7
SG51		G3	7/8
SG52		G3	7/9
SG53		G3	7/10
SG54		G3	7/11
SG55		G3	7/12
SG56		G3	7/13

T0821750-95/d96

Line Drawing Graphics

<i>ID</i>	<i>Graphic</i>	<i>Code</i>	
		<i>Set</i>	<i>Position</i>
DG01		G3	4/0
DG02		G3	4/1
DG03		G3	4/2
DG04		G3	4/3
DG05		G3	4/4
DG06		G3	4/5
DG07		G3	4/6
DG08		G3	4/7
DG09		G3	4/8
DG10		G3	4/9
DG11		G3	4/10
DG12		G3	4/11
DG13		G3	4/12
DG14		G3	5/0
DG15		G3	5/1
DG16		G3	5/2
DG17		G3	5/3
DG18		G3	5/4
DG19		G3	5/5
DG20		G3	5/6

T0821760-95/d97

<i>ID</i>	<i>Graphic</i>	<i>Code</i>	
		<i>Set</i>	<i>Position</i>
DG21		G3	5/7
DG22		G3	5/8
DG23		G3	5/9
DG24		G3	5/10
DG25		G3	5/11
DG26		G3	5/12
DG27		G3	5/13
DG28		G3	5/14
DG29		G3	4/13
DG30		G3	4/14
DG31		G3	4/15
DG32		G1	4/15
DG33		G3	2/1
DG34		G3	2/2
DG35		G3	2/3
DG36		G3	2/4
DG37		G3	2/5
DG38		G3	2/6
DG39		G3	2/7
DG40		G3	2/8

T0821770-95/d98

<i>ID</i>	<i>Graphic</i>	<i>Code</i>	
		<i>Set</i>	<i>Position</i>
DG41		G3	2/9
DG42		G3	2/10
DG43		G3	2/11
DG44		G3	2/12
DG45		G3	2/13
DG46		G3	2/15
DG47		G3	3/15
DG48		G3	3/14
DG49		G3	3/13
DG50		G3	3/12
DG51		G3	3/11
DG52		G3	3/0
DG53		G3	3/1
DG54		G3	3/2
DG55		G3	3/3
DG56		G3	3/4
DG57		G3	3/5
DG58		G3	3/6
DG59		G3	3/7
DG60		G3	3/8

T0821780-95/d99

<i>ID</i>	<i>Graphic</i>	<i>Code</i>	
		<i>Set</i>	<i>Position</i>
DG61		G3	3/9
DG62		G3	3/10
DG63		G3	5/15
DG64		G3	6/14
DG65		G3	6/15

T0821790-95/d100

C.2.2 Format Effector Repertoire

<i>Abbreviation</i>	<i>Name and definition</i>
---------------------	----------------------------

APA	ACTIVE POSITION ADDRESSING
-----	----------------------------

A format effector which causes the active position to move to a defined position on the screen in accordance with parameters following.

APB	ACTIVE POSITION BACK
-----	----------------------

A format effector which causes the active position to move backwards one character position 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 of the first row moves the active position to the last character position of the last row in the defined display area.

APF	ACTIVE POSITION FORWARD
-----	-------------------------

A format effector which causes the active position to move forward to the next character position on the same row. At the last position on the row, this format effector moves the active position to the first character position on the following row. APF on the last character of the last row moves the active position to the first character position on the first row in the defined display area.

APD	ACTIVE POSITION DOWN
-----	----------------------

A format effector which causes the active position to move to the equivalent character position on the following row. APD on the last row moves the active position to the equivalent character position on the first row in the defined display area.

APU	ACTIVE POSITION UP
-----	--------------------

A format effector which causes the active position to move 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 in the defined display area.

APR	ACTIVE POSITION RETURN
-----	------------------------

A format effector which causes the active position to move to the first character position of the same row.

APH	ACTIVE POSITION HOME
	A format effector which causes the active position to be moved to the first character position of the first row of the defined display area.
CS	CLEAR SCREEN
	A format effector which causes the active position to be moved to the first character position of the first row in the defined display area and causes all character positions to be filled with spaces with all attributes set to the default conditions as described in C.1.5.
CAN	CANCEL
	A control function which fills all character positions from the active position to the end of the row inclusive with spaces. The active position is then returned to its previous location.
SP	SPACE
	A format effector which advances the active position one character-width forward on the same row. (It is also regarded as a graphic character with no foreground. In those systems which 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.) SPACE on the last character position of a row moves the active position to the first position of the next row. SPACE on the last character position of a frame moves the active position to the first character position of the frame.
RPT	REPEAT
	A format effector which causes the immediately preceding complete graphic character, including SPACE and DEL, to be displayed a number of times as defined by a parameter.
DEL	DELETE
	In the mosaic graphics mode, the use of DEL moves the active position one space forward, with the vacated space obliterated with the foreground colour. Attributes (double-height, colour, etc.) remain in force.
	In the alphanumeric mode, the use of DEL moves the active position one space forward and displays the DELETE graphics character in the vacated position.
	DEL on the last character position of a row moves the active position to the first position of the next row. DEL on the last character position of a frame moves the active position to the first character position of the frame.
HMS	HOLD MOSAIC
	When the mosaic graphics set is activated, this function causes the last received mosaic graphic character to be displayed in its previously defined rendition when a serial attribute control function is transmitted.
RMS	RELEASE MOSAIC
	Causes the action of HOLD MOSAIC to be stopped.

C.2.3 Attribute Control Repertoire

An attribute control causes the desired display attribute to be applied to the display graphic characters referenced. Four types of attribute control are defined:

- *Full screen attribute controls*

These affect all the character positions on the screen, except the full screen background colour control which affects the full screen background layer.

- *Full row attribute controls*

These affect all the character positions on the defined row, except the full row background colour control which affects the defined row of the full screen background layer.

- *Serial attribute controls*

These apply between markers on a row. They apply from the location of the active position at the time they are received to the end of the row or until a contradictory marker is reached. Each of the control functions of this repertoire causes the active position to be advanced one character-width forwards; the position thus vacated is to be generally displayed as a SPACE. The control HOLD MOSAICS may modify this display. Combinations of control functions may be applied at one character location.

- *Parallel attribute controls*

These are the property of the active position and move with it under the action of format effectors or spacing display characters (including space). They apply to the displayed characters subsequently received until the attributes are changed by relevant controls including certain format effectors (CS, APA, APH). They also apply to spacing display characters (including space) inserted by control commands.

C.2.3.1 FOREGROUND COLOUR Controls

- a) *Full screen and full row controls*

The following controls are available as either full screen or full row controls.

The FOREGROUND COLOUR may be set to any one of the eight colours of the currently invoked colour table using the following controls:

<i>Abbreviation</i>	<i>Name and definition</i>
BKF	BLACK FOREGROUND Invokes 1st colour of the colour table.
RDF	RED FOREGROUND Invokes 2nd colour of the colour table.
GRF	GREEN FOREGROUND Invokes 3rd colour of the colour table.
YLF	YELLOW FOREGROUND Invokes 4th colour of the colour table.
BLF	BLUE FOREGROUND Invokes 5th colour of the colour table.
MGF	MAGENTA FOREGROUND Invokes 6th colour of the colour table.
CNF	CYAN FOREGROUND Invokes 7th colour of the colour table.
WHF	WHITE FOREGROUND Invokes 8th colour of the colour table.

- b) *Serial controls*

The FOREGROUND COLOUR may be set to any one of the eight colours of the currently invoked colour table. The same controls are also used to shift into or out of the first mosaic set (the L-set).

The following 'alpha' foreground colour controls cause the appropriate foreground colour to be applied and a locking shift from the first mosaic set (the L-set) back to the previously invoked G-set:

<i>Abbreviation</i>	<i>Name and definition</i>
ABK	ALPHA BLACK Invokes 1st colour of the colour table.
ANR	ALPHA RED Invokes 2nd colour of the colour table.
ANG	ALPHA GREEN Invokes 3rd colour of the colour table.
ANY	ALPHA YELLOW Invokes 4th colour of the colour table.
ANB	ALPHA BLUE Invokes 5th colour of the colour table.
ANM	ALPHA MAGENTA Invokes 6th colour of the colour table.
ANC	ALPHA CYAN Invokes 7th colour of the colour table.
ANW	ALPHA WHITE Invokes 8th colour of the colour table.

The following 'mosaic' foreground colour controls cause the appropriate foreground colour to be applied and a locking shift to the first mosaic set (the L-set). A shift back from the first mosaic set (the L-set) to the previously invoked G-set is implicit at the start of each row:

<i>Abbreviation</i>	<i>Name and definition</i>
MBK	MOSAIC BLACK Invokes 1st colour of the colour table.
MSR	MOSAIC RED Invokes 2nd colour of the colour table.
MSG	MOSAIC GREEN Invokes 3rd colour of the colour table.
MSY	MOSAIC YELLOW Invokes 4th colour of the colour table.
MSB	MOSAIC BLUE Invokes 5th colour of the colour table.
MSM	MOSAIC MAGENTA Invokes 6th colour of the colour table.
MSC	MOSAIC CYAN Invokes 7th colour of the colour table.
MSW	MOSAIC WHITE Invokes 8th colour of the colour table.

c) *Parallel controls*

The foreground colour may be set to any one of the eight colours of the currently invoked colour table using the following controls:

<i>Abbreviation</i>	<i>Name and definition</i>
BKF	BLACK FOREGROUND Invokes 1st colour of the colour table.
RDF	RED FOREGROUND Invokes 2nd colour of the colour table.

GRF	GREEN FOREGROUND
	Invokes 3rd colour of the colour table.
YLF	YELLOW FOREGROUND
	Invokes 4th colour of the colour table.
BLF	BLUE FOREGROUND
	Invokes 5th colour of the colour table.
MGF	MAGENTA FOREGROUND
	Invokes 6th colour of the colour table.
CNF	CYAN FOREGROUND
	Invokes 7th colour of the colour table.
WHF	WHITE FOREGROUND
	Invokes 8th colour of the colour table.

C.2.3.2 BACKGROUND COLOUR Controls

a) Full screen and full row controls

The following controls are available as either full screen or full row controls. They cause the full screen background layer to adopt one of the eight colours of the currently invoked colour table or transparency:

<i>Abbreviation</i>	<i>Name and definition</i>
BKB	BLACK BACKGROUND
	Invokes 1st colour of the colour table.
RDB	RED BACKGROUND
	Invokes 2nd colour of the colour table.
GRB	GREEN BACKGROUND
	Invokes 3rd colour of the colour table.
YLB	YELLOW BACKGROUND
	Invokes 4th colour of the colour table.
BLB	BLUE BACKGROUND
	Invokes 5th colour of the colour table.
MGB	MAGENTA BACKGROUND
	Invokes 6th colour of the colour table.
CNB	CYAN BACKGROUND
	Invokes 7th colour of the colour table.
WHB	WHITE BACKGROUND
	Invokes 8th colour of the colour table.
TRB	TRANSPARENT BACKGROUND
	Invokes transparent background (the underlying video picture).

b) Serial controls

The following controls affect the character background:

<i>Abbreviation</i>	<i>Name and definition</i>
NBD	NEW BACKGROUND
	Causes the BACKGROUND COLOUR to adopt the current foreground colour as defined by previous colour controls. The foreground colour is unchanged.
BBD	BLACK BACKGROUND
	Causes the BACKGROUND COLOUR to invoke the first colour of the colour table.

c) *Parallel controls*

The following controls cause the character background layer (layer b) to adopt one of the eight colours of the currently invoked colour table or transparency:

<i>Abbreviation</i>	<i>Name and definition</i>
BKB	BLACK BACKGROUND Invokes 1st colour of the colour table.
RDB	RED BACKGROUND Invokes 2nd colour of the colour table.
GRB	GREEN BACKGROUND Invokes 3rd colour of the colour table.
YLB	YELLOW BACKGROUND Invokes 4th colour of the colour table.
BLB	BLUE BACKGROUND Invokes 5th colour of the colour table.
MGB	MAGENTA BACKGROUND Invokes 6th colour of the colour table.
CNB	CYAN BACKGROUND Invokes 7th colour of the colour table.
WHB	WHITE BACKGROUND Invokes 8th colour of the colour table.
TRB	TRANSPARENT BACKGROUND Invokes transparent background.

C.2.3.3 LINING Controls

The following controls are available as full screen, full row, serial or parallel controls:

<i>Abbreviation</i>	<i>Name and definition</i>
STL	START LINING Applies the LINED attribute.
SPL	STOP LINING Stops the application of the LINED attribute.

C.2.3.4 SIZE Controls

a) *Full screen and full row controls*

The following control is available either as a full screen or full row control:

<i>Abbreviation</i>	<i>Name and definition</i>
NSZ	NORMAL-SIZE Applies the NORMAL-SIZE attribute.

b) *Serial and parallel controls*

The following controls are available in both the serial and parallel modes:

<i>Abbreviation</i>	<i>Name and definition</i>
NSZ	NORMAL-SIZE Applies the NORMAL-SIZE attribute.
DBH	DOUBLE-HEIGHT Applies the DOUBLE-HEIGHT attribute.

DBW	DOUBLE-WIDTH
	Applies the DOUBLE-WIDTH attribute.
DBS	DOUBLE-SIZE
	Applies the DOUBLE-SIZE attribute.

NOTE – As described in C.1.4, the action of the DOUBLE-HEIGHT control is different in the serial and parallel modes.

C.2.3.5 FLASH Controls

a) Full screen and full row controls

The following controls are available as either full screen or full row controls:

<i>Abbreviation</i>	<i>Name and definition</i>
FSH	FLASH
	Applies the normal (50%) FLASH attribute.
STD	STEADY
	Cancels the application of any FLASH attribute.

b) Serial and parallel controls

The following controls are available in both the serial and parallel modes:

State controls:

<i>Abbreviation</i>	<i>Name and definition</i>
FSH	FLASH
	Applies the normal flash state.
STD	STEADY
	Cancels the application of any FLASH attribute.
IVF	INVERTED FLASH
	Applies the inverted flash state.
RIF	REDUCED INTENSITY FLASH (flash between colour tables)
	Applies the reduced intensity flash state.

Rate controls:

<i>Abbreviation</i>	<i>Name and definition</i>
FF1	FAST FLASH 1
	Applies the 1st phase of three-phase flash.
FF2	FAST FLASH 2
	Applies the 2nd phase of three-phase flash.
FF3	FAST FLASH 3
	Applies the 3rd phase of three-phase flash.

NOTE – The application of any of the state controls defaults to the normal 50% 1 Hz rate.

<i>Abbreviation</i>	<i>Name and definition</i>
ICF	INCREMENT FLASH
	Three-phase fast flash is applied to characters so that the phase is sequentially changed for every character (enlarged characters count as single characters) in a string of three adjacent characters to produce an apparent movement to the right.
DCF	DECREMENT FLASH
	Three-phase fast flash is applied to characters so that the phase is sequentially changed for every character (enlarged characters count as single characters) in a string of three adjacent characters to produce an apparent movement to the left.

C.2.3.6 CONCEAL Controls

a) Full screen and full row attributes

The following controls are available as either full screen or full row controls:

<i>Abbreviation</i>	<i>Name and definition</i>
CDY	CONCEAL DISPLAY Applies the CONCEAL attribute.
STC	STOP CONCEAL Causes the concealed characters to be revealed.

b) Serial and parallel controls

The following controls are available in both the serial and parallel modes:

<i>Abbreviation</i>	<i>Name and definition</i>
CDY	CONCEAL DISPLAY Applies the CONCEAL attribute.
STC	STOP CONCEAL Stops the application of the CONCEAL attribute.

At full screen and full row level there is no need for a 'non-concealed' control; the 'stop conceal' control is interpreted as 'reveal' and also resets the character positions addressed to the 'not concealed' state.

C.2.3.7 INVERT Controls

a) Full screen, full row and parallel controls

The following controls are available as either full screen, full row or parallel controls:

<i>Abbreviation</i>	<i>Name and definition</i>
IPO	INVERTED POLARITY Applies the INVERT attribute.
NPO	NORMAL POLARITY Stops the application of the INVERT attribute.

b) Serial controls

None.

C.2.3.8 WINDOW/BOX controls

The following controls are available as either full screen, full row, serial or parallel controls.

Full screen controls affect the whole of the defined display area.

Full row controls affect whole rows within the defined display area.

Serial and parallel controls affect parts of rows and individual characters respectively.

<i>Abbreviation</i>	<i>Name and Definition</i>
SBX	START BOX Applies the WINDOW/BOX attribute.
EBX	END BOX Stops the application of the WINDOW/BOX attribute.

C.2.3.9 MARKING controls

The following controls are available as either full screen, full row, serial or parallel controls.

Full screen controls affect the whole of the defined display area.

Full row controls affect whole rows within the defined display area.

Serial and parallel controls affect parts of rows and individual characters respectively.

<i>Abbreviation</i>	<i>Name and Definition</i>
MMS	MARKED MODE START Applies the MARKED attribute.
MMT	MARKED MODE STOP Stops the application of the MARKED attribute.

C.2.3.10 PROTECTING controls

a) *Full screen, full row, serial and parallel controls*

The following controls are available as either full screen, full row, serial or parallel controls.

Full screen controls affect the whole of the defined display area.

Full row controls affect whole rows within the defined display area.

Serial and parallel controls affect parts of rows and individual characters respectively.

<i>Abbreviation</i>	<i>Name and Definition</i>
PMS	PROTECTED MODE START Applies the PROTECTED attribute.
PMC	PROTECTED MODE CANCEL Cancels (removes) the PROTECTED attribute. (allows overwriting).

(b) *Additional serial and parallel controls*

The following controls may be applied in either the serial or parallel mode.

<i>Abbreviation</i>	<i>Name and Definition</i>
PMI	PROTECTED MODE IDLE Stops the application of the PROTECTED attribute.

C.2.3.11 Definition of scrolling area

<i>Abbreviation</i>	<i>Name and Definition</i>
CSA	CREATE SCROLLING AREA Creates a scrolling area
DSA	DELETE SCROLLING AREA Deletes all or part of a scrolling area.

C.2.3.12 Execution of scrolling

a) *Implicit scrolling*

Scroll up

APF, or the printing of a character or spacing attribute control on the last character position, or APD in the lowest row of the selected part of the screen, copies the contents of row *i* to *i*-1. The contents of the uppermost row of the selected part of the screen will be discarded. The lowest row of the selected part of the screen is filled with spaces (2/0) but the off-screen row-defined attributes remain unchanged. Thus the lowest row will show spaces in the row-defined background colour.

Scroll down

APB on the first character position, or APU in the uppermost row of the selected part of the screen, copies the contents of row *i* to row *i* + 1. The contents of the lowest row of the selected part of the screen will be discarded. The uppermost row of the selected part of the screen is filled with spaces (2/0) but the off-screen row-defined attributes remain unchanged. Thus the uppermost row will show spaces in the row-defined background colour.

<i>Abbreviation</i>	<i>Name and Definition</i>
DIS	DEACTIVATE IMPLICIT SCROLLING This deactivates the implicit scrolling, allowing the active position in to move across the border of a scrolling area.
AIS	ACTIVATE IMPLICIT SCROLLING This restores the implicit scrolling effect of format effectors.

b) *Explicit scrolling*

These controls affect the scrolling area.

<i>Abbreviation</i>	<i>Name and Definition</i>
SCU	SCROLL UP This causes a scrolling up of the designated scrolling area.
SCD	SCROLL DOWN This causes a scroll down of the designated area. The active position does not move relative to the defined display area.

C.2.3.13 Colour table controls

The following controls invoke the selected colour table.

<i>Abbreviation</i>	<i>Name and Definition</i>
CT1	COLOUR TABLE 1 Invokes 1st colour table.
CT2	COLOUR TABLE 2 Invokes 2nd colour table.
CT3	COLOUR TABLE 3 Invokes 3rd colour table.
CT4	COLOUR TABLE 4 Invokes 4th colour table.

These controls are Locking controls and are reset by a contradictory control or clear screen (CS). Invoking a colour table has no effect on an attribute until that attribute is changed.

C.2.3.14 Miscellaneous presentation functions

SPD Select Presentation Direction

SPD is used to establish for subsequent text the direction of the active position movement for graphic characters and for the control functions which initiate movement of the active position. The established directions remain in effect until the next occurrence of SPD in the data stream. The direction of the character path may be temporarily reversed by START REVERSED STRING (SRS).

SRS Start Reversed String

SRS is used to delimit a string of graphic characters which is to be presented in a direction that will not necessarily be that of the immediately preceding text. For example, SRS may temporarily reverse the direction of the established character path until the end of the delimited string.

A string delimited by the SRS may contain n (where n is zero or more) further occurrences of SRS with a parameter value not equal to 0; the end of the string is indicated by the $(n + 1)^{\text{th}}$ following occurrence of SRS with a parameter value of 0, the active position is moved to the character position following the graphic character presented

SAPV Select Alternative Presentation Variant

SAPV is used to specify one or more variants for the presentation of subsequent text.

FNT Font Selection

FNT is used to identify the character font to be selected as primary or alternative font by subsequent occurrences of SELECT GRAPHIC RENDITION (SGR) in the data stream.

SGR Select Graphic Rendition

SGR is used to establish one or more graphic rendition aspects for subsequent text. The established aspects remain in effect until the next occurrence of SGR in the data stream, depending on the setting of the GRAPHIC RENDITION COMBINATION MODE (GRCM).

C.2.4 Device control function repertoire

C.2.4.1 Cursor controls

<i>Abbreviation</i>	<i>Name and Definition</i>
CON	CURSOR ON A device control function which causes the active position to be indicated.
COF	CURSOR OFF A device control function which terminates the action of CON.

C.2.4.2 Recording device controls

<i>Abbreviation</i>	<i>Name and Definition</i>
RDS	RECORDING DEVICE START Causes the associated recording device to start recording data subsequently received by the terminal.
RDT	RECORDING DEVICE STOP Causes the associated recording device to stop.
RDW	RECORDING DEVICE WAIT Causes the associated recording device to wait.

C.2.4.3 Hard copy device controls

<i>Abbreviation</i>	<i>Name and Definition</i>
HCS	HARD COPY START Causes the associated hard copy device to start copying data subsequently received by the terminal.

HCT	HARD COPY STOP Causes the associated hard copy device to stop.
HCW	HARD COPY WAIT Causes the associated hard copy device to wait.

C.2.4.4 Display device controls

<i>Abbreviation</i>	<i>Name and Definition</i>
DDO	DISPLAY DEVICE ON Data subsequently received by the terminal is displayed.
DDF	DISPLAY DEVICE OFF Data subsequently received by the terminal is not displayed .

C.2.4.5 Auxiliary device controls

<i>Abbreviation</i>	<i>Name and Definition</i>
ADO	AUXILIARY DEVICE ON Data subsequently received by the terminal is passed to the auxiliary device.
ADF	AUXILIARY DEVICE OFF Data subsequently received by the terminal is not passed to the auxiliary device.

C.2.4.6 Miscellaneous device controls

<i>Abbreviation</i>	<i>Name and Definition</i>
EBU	EMPTY BUFFER Causes the contents of the terminal buffer to be transmitted to the line.

C.3 Coding Structure

The coding structure defined allows for both 7- and 8-bit coding of presentation data.

Control functions are coded using primary and supplementary control sets and by using combinations of control codes and following parameters.

Characters are coded in five character-sets.

In the 7-bit environment only one of these character-sets may be invoked into the 'in use' code table.

In the 8-bit environment two of these character-sets may be invoked into the 'in use' code table.

In order to invoke the character-sets, locking shift functions are required for all sets (G0, G1, G2, G3 and L). To enable access to the sets not invoked, single shift functions are also incorporated.

The designation of the sets from a library to the G0, G1, G2 and G3 sets is, in accordance with ISO 2022, the same for both the 8-bit and 7-bit environment.

C.3.1 Code extension and invocation

C.3.1.1 Common code extension control functions

<i>Abbreviation</i>	<i>Name and Definition</i>
ESC	ESCAPE 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.

CSI	CONTROL SEQUENCE INTRODUCER
	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.
	This control is mainly used in the context of bidirectionnal text for Latin/Arabic Videotex applications or for functions associated with ideographic writing.

The coding of CSI-introduce functions is described below:

C.3.1.2 Invocation functions (7-bit environment)

<i>Abbreviation</i>	<i>Name and Definition</i>
SO	SHIFT OUT Invokes the G1 set into columns 2-7 of the code table.
SI	SHIFT IN Invokes the G0 set into columns 2-7 of the code table.
LS2	LOCKING SHIFT 2 Invokes the G2 set into columns 2-7 of the code table.
LS3	LOCKING SHIFT 3 Invokes the G3 set into columns 2-7 of the code table.
SS2	SINGLE SHIFT 2 Invokes a single character from the G2 set.
SS3	SINGLE SHIFT 3 Invokes a single character from the G3 set.

NOTE – L-set activation is by serial C1-controls 5/0 to 5/7. L-set deactivation is by any one of the following:

- serial C1-controls 4/0 to 4/7;
- invocation of parallel C1-set;
- entering new line;
- invocation of a G-set into columns 2 to 7 of the code table.

C.3.1.3 Invocation functions (8-bit environment)

<i>Abbreviation</i>	<i>Name and Definition</i>
LS0	LOCKING SHIFT 0 Invokes the G0 set into columns 2-7 of the code table.
LS1	LOCKING SHIFT 1 Invokes the G1 set into columns 2-7 of the code table.
LS1R	LOCKING SHIFT 1 RIGHT Invokes the G1 set into columns 10-15 of the code table.
LS2	LOCKING SHIFT 2 Invokes the G2 set into columns 2-7 of the code table.
LS2R	LOCKING SHIFT 2 RIGHT Invokes the G2 set into columns 10-15 of the code table.
LS3	LOCKING SHIFT 3 Invokes the G3 set into columns 2-7 of the code table.
LS3R	LOCKING SHIFT 3 RIGHT Invokes the G3 set into columns 10-15 of the code table.

SS2	SINGLE SHIFT 2 Invokes a single character from the G2 set.
-----	---

SS3	SINGLE SHIFT 3 Invokes a single character from the G3 set.
-----	---

NOTE – L-set activation is by serial C1-controls 5/0 to 5/7. L-set deactivation is by any one of the following:

- serial C1-controls 4/0 to 4/7;
- invocation of parallel C1-set;
- entering new line;
- invocation of a G-set into columns 2 to 7 of the code table.

C.3.1.4 Default code sets

The primary control function set is designated the CO set. Either of the supplementary control function sets may be designated as the default C1 set.

The primary set of characters is designated the G0 set. The supplementary set of alphanumeric characters is designated the G2 set.

The first supplementary set of mosaic characters is designated the L set and is invoked by controls in the serial C1 set.

The second supplementary set of mosaic characters is designated the G1 set.

The third supplementary set of mosaic characters is designated the G3 set. In the 8-bit environment the G0 set is invoked into columns 2-7 and the G2 set is invoked into columns 10-14 of the 'in use' code table.

C.3.2 The Primary control function set (see Table C.5)

This set contains two types of elements: those which consist of a single bit combination and those which are used in conjunction with following parameters (RPT and APA).

C.3.2.1 Parameters for format effectors

Repeat RPT (char)

The parameter (char) indicates the number of repetitions of the immediately preceding graphic character. The representation is in binary form by the 6 least significant bits of the parameter which is taken from columns 4 to 7. The character itself is not included in the count. This function does not apply to control characters.

Active Position Address APA (char) (char)

A control function with a two or four character parameter. All the characters are within the range 4/0¹⁰⁾ to 7/15, and they represent respectively the row address and the column address in binary form, with 6 useful bits (bit 6 being the most significant bit) of the first character to be displayed.

The first character received shall be displayed on the designated character location of the addressed row.

The default address range of the defined display area is 1 to 24 vertically and 1 to 40 horizontally. The location addressed by APA, 4/1, 4/1 (or APA 4/0, 4/1, 4/0, 4/1 if the format exceeds either 63 rows or 63 columns) is the top left-hand location of the defined display area.

If the format exceeds either 63 rows or 63 columns, then the relevant parameters, i.e. the row and the column address, is coded as a two byte sequence with 12 useful bits, the first byte carrying the most significant bits.

¹⁰⁾ Addressing row 0 is for private use.

Table C.5/T.101 – The primary control function set (default C0 set)

				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	NUL							
0	0	0	1	1		CON						
0	0	1	0	2		RPT						
0	0	1	1	3								
0	1	0	0	4		COF						
0	1	0	1	5								
0	1	1	0	6								
0	1	1	1	7								
1	0	0	0	8	APB	CAN						
1	0	0	1	9	APF	SS2						
1	0	1	0	10	APD							
1	0	1	1	11	APU	ESC						
1	1	0	0	12	CS							
1	1	0	1	13	APR	SS3						
1	1	1	0	14	SO	APH						
1	1	1	1	15	SI	APA ^{a)}						

a) This code is also used for the Unit Separator (US) control.

 Empty positions in the table denote bit combinations reserved for future standardization and shall not be used.

 Shaded code positions are reserved for G sets and shall not be used for control characters.

C.3.3 The supplementary control function sets

Two supplementary control function sets are defined; one for applying 'serial' attribute controls and one for applying 'parallel' attribute controls.

In the 7-bit environment individual characters of these sets are represented by two-byte combinations of the form ESC, Fe where Fe lies in the range 4/0 to 5/15.

In the 8-bit environment individual characters of these sets are represented by the combinations in the range 8/0 to 9/15.

C.3.3.1 The serial supplementary control function set (see Table C.6)

This set is designated by the sequence ESC 2/2 4/0.

C.3.3.2 The parallel supplementary control function set (see Table C.7)

This set is designated by the sequence ESC 2/2 4/1.

C.3.3.3 CSI introduced functions

The functions introduced by CSI are common to both supplementary control function sets.

SPD SELECT PRESENTATION DIRECTION

The utilized sequence is: CSI Ps 2/0 5/0

default value of Ps = 0;

Ps Call take the following values:

- 0 The Direction of character path is from left to right; the direction of the line progression is from top to bottom. This value corresponds to the Latin way of writing.
- 1 The Direction of character path is from top to bottom (up down); the direction of the line progression is from right to left.
- 2 The direction of character path is from top to bottom (up down); the direction of the line progression is left to right.
- 3 The direction of the character path is from right to left; the direction of the line progression is from top to bottom. This value corresponds to Arabic way of writing.
- 4 The direction of the character path is from bottom to top (down up); the direction of the line progression is from left to right.
- 5 The direction of the character path is from right to left; the direction of the line progression is from bottom to top.
- 6 The direction of the character path is from left to right; the direction of the line progression is from bottom to top.
- 7 The direction of the character path is from bottom to top (down up); the direction of the line is from right to left.

SRS Start Reversed String

A string delimited by the SRS may contain n (where n is zero or more) further occurrences of SRS with a parameter value not equal to 0; the end of the string is indicated by the (n + 1)th following occurrence of SRS with a parameter value of 0, the active position is moved to the character position following the graphic character presented.

Representation: CSI Ps 5/1 1

where Ps:

- 0 end of the string
- 1 start of the string in a direction opposite to that of the immediately preceding text.

SAPV Select Alternative Presentation Variant

Representation: CSI Ps 2/0 5/13

SAPV is used to specify one or more variants for the presentation of subsequent text, the parameter values are:

- 0 Default presentation (implementation-defined); cancels the effect of any preceding occurrence of SAPV in the data stream.
- 1 The decimal digits are presented by means of the graphic symbols used in latin script.
- 2 The decimal digits are presented by means of the graphic symbols used in Arabic script, i.e. the Hindi symbols.
- 3 Each of the graphic characters in the graphic character set(s) in use which is one of left/right handed pair (parentheses, square brackets, curly brackets, greater than/less-than signs, etc.) is presented as “mirrored”, i.e. as the other member of the pair. For example, the coded graphic character given the name LEFT PARENTHESIS is presented as RIGHT PARENTHESIS, and vice versa.
- 5 The following graphic character is presented in its isolated form.
- 6 The following graphic character is presented in its initial form.
- 7 The following graphic character is presented in its final form.
- 8 The following graphic character is presented in its medial form.
- 9 Where the bit combination 02/14 is intended to represent a decimal mark in a decimal number it shall be presented by means of the graphic symbol FULL STOP.
- 10 Where the bit combination 02/14 is intended to represent a decimal mark in a decimal number it shall be presented by means of the graphic symbol COMMA.
- 11 Vowels are presented above or below the preceding character.
- 12 Vowels are presented after the preceding character.
- 13 Contextual shape determination of Arabic scripts, including the LAM-ALEPH ligature but excluding all other Arabic ligatures.
- 14 Contextual shape determination of Arabic scripts excluding all Arabic ligatures.
- 15 Cancels the effect of parameter values 3 and 4.
- 16 Vowels are not presented.

FNT Font Selection

FNT is used to identify the character font to be selected as primary or alternative font by subsequent occurrences of SELECT GRAPHIC RENDITION (SGR) in the data stream.

Representation: CSI Psl; Ps2 2/0 4/4

parameter default values Psl = 0; Ps2 = 0

Ps1 specifies the primary or alternative font concerned:

- 0 Primary font
- 1 First alternative font
- 2 Second alternative font
- 3 Third alternative font
- 4 Fourth alternative font
- 5 Fifth alternative font
- 6 Sixth alternative font
- 7 Seventh alternative font

- 8 Eighth alternative font
- 9 Ninth alternative font

Ps2 identifies the character font according to some register which is to be established.

SGR Select Graphic Rendition

Representation: CSI Ps 6/13

Each graphic rendition aspect is specified by a parameter value:

- 0 Default rendition (implementation-defined), cancels the effect of any preceding occurrence of SGR in the data stream, depending on the setting of the GRAPHIC RENDITION COMBINATION (GRCM)
- 1 Bold or increased intensity
- 2 Faint, decreased intensity or second color
- 3 Italicized
- 4 Underlined
- 5 Slowly blinking (less than 150 per minute)
- 6 Rapidly blinking (150 per minute or more)
- 7 Negative image
- 8 Concealed characters
- 9 Crossed-out (characters still legible but marked as to be deleted)
- 10 Primary (default) font
- 11 First alternative font
- 12 Second alternative font
- 13 Third alternative font
- 14 Fourth alternative font
- 15 Fifth alternative font
- 16 Sixth alternative font
- 17 Seventh alternative font
- 18 Eighth alternative font
- 19 Ninth alternative font
- 20 Fraktur (Gothic)
- 21 Doubly underlined
- 22 Normal colour or normal intensity (neither bold nor faint)
- 23 Not italicized, not fraktur
- 24 Not underlined (neither singly nor doubly)
- 25 Steady (not blinking)
- 26 (Reserved for proportional spacing as specified in CCITT Recommendation T.61, 1984)
- 27 Positive image
- 28 Revealed characters
- 29 Not crossed out

- 30 Black display
- 31 Red display
- 32 Green display
- 33 Yellow display
- 34 Blue display
- 35 Magenta display
- 36 Cyan display
- 37 White display
- 38 (Reserved for future standardization)
- 39 Default display colour (implementation defined)
- 40 Black background
- 41 Red background
- 42 Green background
- 43 Yellow background
- 44 Blue background
- 45 Magenta background
- 46 Cyan background
- 47 White background
- 48 (Reserved for future standardization)
- 49 Default background colour (implementation defined)
- 50 (Reserved for cancelling the effect of the rendering aspect established by parameter 26)
- 51 Framed
- 52 Encircled
- 53 Overlined
- 54 Not framed, not encircled
- 55 Not overlined
- 56 (Reserved for future standardization)
- 57 (Reserved for future standardization)
- 58 (Reserved for future standardization)
- 59 (Reserved for future standardization)
- 60 Ideogram underline or right side line
- 61 Ideogram double underline or double right side line
- 62 Ideogram overline or left side line
- 63 Ideogram double overline or left side line
- 64 Ideogram stress marking
- 65 Cancels the effect of the rendition aspects established by parameters 60 to 64

Table C.6/T.101 – The serial supplementary control function set

				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					ABK	MBK		
0	0	0	1	1					ANR	MSR		
0	0	1	0	2					ANG	MSG		
0	0	1	1	3					ANY	MSY		
0	1	0	0	4					ANB	MSB		
0	1	0	1	5					ANM	MSN		
0	1	1	0	6					ANC	MSC		
0	1	1	1	7					ANW	MSW		
1	0	0	0	8					FSH	CDY		
1	0	0	1	9					STD	SPL		
1	0	1	0	10					EBX	STL		
1	0	1	1	11					SBX	CSI		
1	1	0	0	12					NSZ	BBO		
1	1	0	1	13					DBH	NBD		
1	1	1	0	14					DBW	HMS		
1	1	1	1	15					DBS	RMS		

Table C.7/T.101 – The parallel supplementary control function set

					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						BKF	BKB		
0	0	0	1	1						RDF	RDB		
0	0	1	0	2						GRF	GRB		
0	0	1	1	3						YLF	YLB		
0	1	0	0	4						BLF	BLB		
0	1	0	1	5						MGF	MGB		
0	1	1	0	6						CNF	CNB		
0	1	1	1	7						WHF	WHB		
1	0	0	0	8						FSH	CDY		
1	0	0	1	9						STD	SPL		
1	0	1	0	10						EBX	STL		
1	0	1	1	11						SBX	CSI		
1	1	0	0	12						NSZ	NPO		
1	1	0	1	13						DBH	IPO		
1	1	1	0	14						DBW	TRB		
1	1	1	1	15						DBS	STC		

C.3.4 The coding of graphic characters

C.3.4.1 Code sets

Nine code sets are used to encode the graphic characters. These are:

- 1) The *primary set of characters* (see Table C.8)
This consists of the most frequently used alphanumeric characters and punctuation marks. The bit combination 2/0 is used for SPACE and 7/15 is used for DELETE.
- 2) The *supplementary set of alphanumeric characters* (see Table C.9)
This set contains three types of characters:
 - 4/0 to 4/15
Diacritical marks which are used in combination with the letters of the basic Latin or Greek alphabet in the primary set to constitute the coded representations of characters with diacritical marks. Each of these characters acts as a modifier indicating that the immediately following letter is to be transformed.
 - 6/0 to 7/14
Alphabetic characters which are used in addition to the basic Latin alphabet in the primary set and which are not composed by combining diacritical marks and basic letters.
 - 2/1 to 3/15 and 5/0 to 5/15
Non-alphabetic characters which are used in addition to those in the primary set.
- 3) The *first supplementary set of mosaic characters* (see Table C.10)
This set consists of 63 block mosaic characters and 32 text characters, the representation of which is identical to that of the characters of columns 4 and 5 of the primary set of characters.
- 4) The *second supplementary set of mosaic characters* (see Table C.11)
This set consists of 63 block mosaic characters, 28 smoothed mosaic characters, two line vertical bars and one shading character.
- 5) The *third supplementary set of mosaic characters* (see Table C.12)
This set consists of 28 smoothed mosaic characters, 24 line drawing characters and 7 miscellaneous characters.
- 6) The *Greek primary set* (see Table C.13)
This consists of the most frequently used Greek alphanumeric characters and punctuation marks. The bit combination 2/0 is used for SPACE and 7/15 is used for DELETE.
- 7) The *Arabic primary set* (see Table C.14)
This consists of the isolated form of the Arabic alphabetic characters and one of the presentation variants of the bit combination 2/0 is used for SPACE and 7/15 is used for DELETE.
- 8) The *Hebrew supplementary set* (see Table C.15)
This consists of the non dotted Hebrew characters.
- 9) The *Chinese primary set* (see Table C.16)
This set is defined as a combination of Chinese national standards GB 2312 and GB 8565.2. It is composed of general characters, ordinal numbers, numerical characters, latin, greek and russian alphabets, japanese kanas, phonetic chinese symbols, letters for phonetic chinese annotation and chinese ideograms, making a total of 8443 graphic characters. This set is a multibyte graphic set where each character is represented by a pair of bytes in the range 2/0 to 7/14.

General characters

There are 224 general characters including space, punctuation marks, arithmetic signs, unit marks and background characters.

Ordinal Numbers

There are 60 ordinal numbers, 20 ranging from 1. to 20., 20 from (1) to (20) 10 from ① to ⑩ and 10 Chinese numbers from (–) to (+).

Numeric characters

There are 22 numerical characters, 10 ranging from 0 to 9 and 12 from I to XII.

Latin alphabet

There are 52 latin letters, including 26 upper case from A to Z and 26 lower case from a to z.

Syllabic japanese kanas

There are 169 Japanese Kanas including 83 Hiragana and 86 Katakana.

Greek alphabet

There are 48 greek letters, including 24 upper case from Α to Ω and 24 lower case from α to ω.

Russian alphabet

There are 66 russian letters, including 33 upper case from А to Я and 33 lower case from а to я.

Chinese Phonetic symbols

There are 26 Chinese phonetic symbols and 6 syllabic special characters.

Chinese phonetic-annotated letters

There are 37 letters for Chinese-annotated letters, from ㄅ to ㄨ.

Half length graphic characters

There are 128 half length graphic characters, including 94 general characters and 34 Chinese phonetic symbols.

Chinese characters

There are 7607 chinese characters, 6763 of them from the Chinese national standard GB 2312, 705 from the Chinese national standard GB 8565.2 and 139 general Chinese character and variants.

Empty positions

The empty positions in rows 88 and 89 may be used to define the correspondance of one simplified form of a Chinese character with the multiple complex forms of Chinese characters.

Presentation of the coded set in Table C.16.

Table C.16 is presented for practical reasons on 25 different pages, the complete table being subdivided into 25 sections to accomodate the size of the printed ISO A4 page as shown in the following table.

Columns	1 to 19	20 to 38	39 to 57	58 to 76	77 to 94
Rows					
1 to 19	1	2	3	4	5
20 to 38	6	7	8	9	10
39 to 57	11	12	13	14	15
58 to 76	16	17	18	19	20
77 to 94	21	22	23	24	25

Table C.8/T.101 – The primary set of graphic characters (default G0 set)

				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			SP	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	# / _	o	

/ _

The characters allocated to positions 5/15 may be displayed either as _ (LOWER BAR) or # (SQUARE) to represent the terminator function required by existing Videotex services.

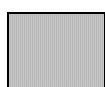


The representation of these characters is not guaranteed in international communication and may be replaced by national application oriented variants.

Table C.9/T.101 – The supplementary set of graphic characters (default G2 set)

				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				◦	”	—	Ω	Κ
0	0	0	1	1			ı	±	Π	ı	Æ	æ
0	0	1	0	2			¢	2	★	®	Ð	Ω
0	0	1	1	3			£	3	↗	©	ä	ð
0	1	0	0	4			\$	×	↗	™	☞	≡
0	1	0	1	5			¥	μ	↗	\		↗
0	1	1	0	6			#	¶	□		☺	er
0	1	1	1	7			§	•	□		☹	●
1	0	0	0	8			¤	÷	★		⌘	⌘
1	0	0	1	9			‘	’	★		Ø	ø
1	0	1	0	10			“	”	✕		Œ	œ
1	0	1	1	11			«	»	✓		º	β
1	1	0	0	12			←	¼		📄	🏳️	↓
1	1	0	1	13			↑	½	☒	📄	❄️	◆
1	1	1	0	14			→	¾	☑	🕒	↑	⇒
1	1	1	1	15			↓	¿	🏠	🏠	’n	

 4/8 is diaeresis and this character is used for compatibility with other text communication services which may need to distinguish between umlaut and diaeresis.

 Empty positions in the table denote bit combinations reserved for future standardization and shall not be used.

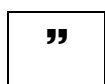
 ” The diacritical mark "dieresis with accent" in position 4/0 should only be used with the Greek primary set of graphic characters.

Table C.10/T.101 – The first supplementary set of mosaic characters (L set)

					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						@	P		
0	0	0	1	1						A	Q		
0	0	1	0	2						B	R		
0	0	1	1	3						C	S		
0	1	0	0	4						D	T		
0	1	0	1	5						E	U		
0	1	1	0	6						F	V		
0	1	1	1	7						G	W		
1	0	0	0	8						H	X		
1	0	0	1	9						I	Y		
1	0	1	0	10						J	Z		
1	0	1	1	11						K	←		
1	1	0	0	12						L	1		
1	1	0	1	13						M	→		
1	1	1	0	14						N	↑		
1	1	1	1	15						O	#/_		



The characters allocated to positions 5/15 may be displayed either as _ (LOWER BAR) or # (SQUARE) to represent the terminator function required by existing Videotex services.



The representation of these characters is not guaranteed in international communication and may be replaced by national application oriented variants.

Table C.11/T.101 – The second supplementary set of mosaic characters (default G1 set)

					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	0								
0	0	0	1	1	1								
0	0	1	0	2	2								
0	0	1	1	3	3								
0	1	0	0	4	4								
0	1	0	1	5	5								
0	1	1	0	6	6								
0	1	1	1	7	7								
1	0	0	0	8	8								
1	0	0	1	9	9								
1	0	1	0	10	10								
1	0	1	1	11	11								
1	1	0	0	12	12								
1	1	0	1	13	13								
1	1	1	0	14	14								
1	1	1	1	15	15								

Table C.12/T.101 – The third supplementary set of mosaic characters (default G3 set)

					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									
0	0	0	1	1									
0	0	1	0	2									
0	0	1	1	3									
0	1	0	0	4									
0	1	0	1	5									
0	1	1	0	6									
0	1	1	1	7									
1	0	0	0	8									
1	0	0	1	9									
1	0	1	0	10									
1	0	1	1	11									
1	1	0	0	12									
1	1	0	1	13									
1	1	1	0	14									
1	1	1	1	15									



Empty positions in the table denote bit combinations reserved for future standardization and shall not be used.

NOTE – In order to preserve the compatibility with Recommendation T.101 (1988), the use of the following code positions is optional: 2/1 to 2/13, 2/15, 3/0 to 3/15, 5/15, 6/14 and 6/15.

Table C.13/T.101 – The Greek primary set of graphic characters

					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					0	✋	☞	‘	◻
0	0	0	1	1				!	1	✌	⚙	☞	◻
0	0	1	0	2				"	2	✋		☞	✚
0	0	1	1	3				#	3	✋	💧	☞	♦
0	1	0	0	4				☞	4	✋	☞	☞	♦
0	1	0	1	5				%	5	✋	✚	☞	♦
0	1	1	0	6				&	6	☞	✋	☞	☞
0	1	1	1	7				☞	7	✋	☞	☞	☞
1	0	0	0	8				(	8	✈	☞	◻	☞
1	0	0	1	9				)	9	✋	☞	☞	♦
1	0	1	0	10				*	:	☞		☞	
1	0	1	1	11				+	;	☞	☞	●	{
1	1	0	0	12				,	<	☞	☞	☞	
1	1	0	1	13				-	=	☞	☞	■	}
1	1	1	0	14				.	>	☞	☞	☞	—
1	1	1	1	15				/	?	☞	☞	◻	

Telematic terminals should not transmit this code. however, to ensure compatibility with some conversion equipments, when a telematic terminal receives code 5/2, it shall interpret it as a capital Greek letter Sigma (Σ).

These code positions are reserved for further standardization.

Table C.14/T.101 – The Arabic primary set of graphic characters

				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				0	✎	✎	←	⇒
0	0	0	1	1			!	1	🕒	✎	↑	⇧
0	0	1	0	2			"	2	🕒	✎	↓	⇩
0	0	1	1	3			#	3	↩	✎	↖	
0	1	0	0	4			α	4	↩	✎	↗	
0	1	0	1	5			%	5	↩	✎	↙	
0	1	1	0	6			&	6	✎	✎	↘	
0	1	1	1	7			💧	7	↩	✎	←	
1	0	0	0	8			(	8	↩	✎	→	
1	0	0	1	9			)	9	↩	✎	↑	
1	0	1	0	10			*	:	↩	✎	↓	
1	0	1	1	11			+	;	✎	✎	↖	}
1	1	0	0	12			,	>	✎	✎	↗	
1	1	0	1	13			-	=	✎	☯	↙	{
1	1	1	0	14			.	<	✎	^	↘	~
1	1	1	1	15			/	?	✎	✎	⇧	



These code positions are reserved for further standardization.

Table C.15/T.101 – The Hebrew supplementary set of graphic characters

				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							→	⇒
0	0	0	1	1							↑	⇑
0	0	1	0	2							↓	⇓
0	0	1	1	3							↖	↔
0	1	0	0	4							↗	↕
0	1	0	1	5							↙	↗
0	1	1	0	6							↘	↖
0	1	1	1	7							←	⇐
1	0	0	0	8							→	⇨
1	0	0	1	9							↑	◻
1	0	1	0	10							↓	◻
1	0	1	1	11							↖	
1	1	0	0	12							↗	
1	1	0	1	13							↙	
1	1	1	0	14							↘	
1	1	1	1	15							↖	



These code positions are reserved for further standardization.

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 1

File name:

1区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01-19																			
20-39	<	>	<<	>>	「	」	『	』	【	】	±	×	÷	:	^	∇	Σ	Π	
40-59	U	∩	∈	::	√	⊥	//	∠	∪	⊙	∫	≡	≈	≈	∞	≠	≠	≠	≠
60-79	≤	≥	∞	::	∴	♂	♀	°	'	"	℃	\$	⊙	⊙	£	‰	§	No	☆
80-94	○	●	◎	◇	◆	□	■	△	▲	※	→	←	↑	↓	≡				
	2121	2122	2123	2124	2125	2126	2127	2128	2129	213A	212B	212C	212D	212E	212F	2130	2131	2132	2133
	2134	2135	2136	2137	2138	2139	213A	213B	213C	213D	213E	213F	2140	2141	2142	2143	2144	2145	2146
	2148	2149	214A	214B	214C	214D	214E	214F	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	215A
	215C	215D	215E	215F	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	216A	216B	216C	216D	216E
	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	217A	217B	217C	217D	217E				

2区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01-19																	1.	2.	3.
20-39	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.	17.	18.	19.	20.	(1)	(2)
40-59	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)	①	②
60-79	④	⑤	⑥	⑦	⑧	⑨	⑩		(-)	(=)	(≡)	(四)	(五)	(六)	(七)	(八)	(九)	(十)	(H)
80-94	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII							
	2221	2222	2223	2224	2225	2226	2227	2228	2229	222A	222B	222C	222D	222E	222F	2230	2231	2232	2233
	2234	2235	2236	2237	2238	2239	223A	223B	223C	223D	223E	223F	2240	2241	2242	2243	2244	2245	2246
	2248	2249	224A	224B	224C	224D	224E	224F	2250	2251	2252	2253	2254	2255	2256	2257	2258	2259	225A
	225C	225D	225E	225F	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	226A	226B	226C	226D	226E
	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	227A	227B	227C	227D	227E				

3区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01-19	!	"	#	¥	%	&	'	(	)	*	+	,	-	.	/	0	1	2	3
20-39	4	5	6	7	8	9	:	;	<	=	>	?	@	A	B	C	D	E	F
40-59	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
60-79	\	]	^	_	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n
80-94	p	q	r	s	t	u	v	w	x	y	z	{		}					
	2321	2322	2323	2324	2325	2326	2327	2328	2329	232A	232B	232C	232D	232E	232F	2330	2331	2332	2333
	2334	2335	2336	2337	2338	2339	233A	233B	233C	233D	233E	233F	2340	2341	2342	2343	2344	2345	2346
	2348	2349	234A	234B	234C	234D	234E	234F	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359	235A
	235C	235D	235E	235F	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	236A	236B	236C	236D	236E
	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	237A	237B	237C	237D	237E				

4区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01-19	あ	あ	い	い	う	う	え	え	お	お	か	が	き	ぎ	く	ぐ	け	げ	こ
20-39	ご	さ	ざ	し	じ	す	ず	せ	ぜ	そ	ぞ	た	だ	ち	ぢ	つ	づ	て	で
40-59	と	ど	な	に	ぬ	ね	の	は	ば	び	び	ふ	ぶ	ぶ	へ	べ	べ	ほ	ほ
60-79	ぼ	ぼ	ま	み	む	め	も	や	や	ゆ	ゆ	よ	よ	ら	り	る	れ	ろ	わ
80-94	ゐ	ゑ	を	ん															
	2421	2422	2423	2424	2425	2426	2427	2428	2429	242A	242B	242C	242D	242E	242F	2430	2431	2432	2433
	2434	2435	2436	2437	2438	2439	243A	243B	243C	243D	243E	243F	2440	2441	2442	2443	2444	2445	2446
	2448	2449	244A	244B	244C	244D	244E	244F	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	245A
	245C	245D	245E	245F	2460	2461	2462	2463	2464	2465	2466	2467	2468	2469	246A	246B	246C	246D	246E
	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479	247A	247B	247C	247D	247E				

T0814430-93/d104

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 2

5区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	ア	ア	イ	イ	ウ	ウ	エ	エ	オ	オ	カ	ガ	キ	ギ	ク	グ	ケ	ゲ	コ
20/39	ゴ	サ	ザ	シ	ジ	ス	ズ	セ	ゼ	ソ	ゾ	タ	チ	ヂ	ツ	ヅ	テ	デ	
40/59	ト	ド	ナ	ニ	ヌ	ネ	ノ	ハ	バ	パ	ヒ	ビ	フ	ブ	ヘ	ベ	ペ	ホ	
60/79	ボ	ポ	マ	ミ	ム	メ	モ	ヤ	ユ	ヨ	ヨ	ラ	リ	ル	レ	ロ	ワ	ワ	
80/94	ヰ	ヱ	ヲ	ン	ヴ	カ	ケ												
	2570	2571	2572	2573	2574	2575	2576	2577	2578	2579	257A	257B	257C	257D	257E				
6区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M	N	Ξ	O	Π	P	Σ	T
20/39	Φ	X	Ψ	Ω										α	β	γ	δ	ε	ζ
40/59	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω	ς	η
60/79	⏏	⏐	⏑	⏒	⏓	⏔	⏕	⏖	⏗	⏘	⏙	⏚	⏛	⏜	⏝	⏞	⏟	⏠	⏡
80/94	⏢	⏣	⏤	⏥	⏦	⏧	⏨	⏩	⏪	⏫	⏬	⏭	⏮	⏯	⏰	⏱	⏲	⏳	⏴
	2670	2671	2672	2673	2674	2675	2676	2677	2678	2679	267A	267B	267C	267D	267E				
7区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	A	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С
20/39	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я					
40/59									а	б	в	г	д	е	ё	ж	з	и	й
60/79	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь
80/94	ю	я																	
	2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	277A	277B	277C	277D	277E				
8区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	ā	á	ǎ	à	ē	é	ě	è	ī	í	ǐ	ì	ō	ó	ǒ	ò	ū	ú	ǔ
20/39	û	ū	ǘ	ǚ	ü	ê	ɑ	ɑ̃	ɲ	ɲ̃	ɳ	g					ɥ	ɥ̃	ɰ
40/59	ㄥ	ㄆ	ㄇ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ
60/79	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ
80/94	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ
	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879	287A	287B	287C	287D	287E				

T0814440-93/d105

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 3

9区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	
01/19		2921	2922	2923	2924	2925	2926	2927	2928	2929	292A	292B	292C	292D	292E	292F	2930	2931	2932	2933
20/39	2934	2935	2936	2937	2938	2939	293A	293B	293C	293D	293E	293F	2940	2941	2942	2943	2944	2945	2946	2947
40/59	2948	2949	294A	294B	294C	294D	294E	294F	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959	295A	295B
60/79	295C	295D	295E	295F	2960	2961	2962	2963	2964	2965	2966	2967	2968	2969	296A	296B	296C	296D	296E	296F
80/94	2970	2971	2972	2973	2974	2975	2976	2977	2978	2979	297A	297B	297C	297D	297E					
10区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	
01/19	!	"	#	¥	%	&	'	(	)	*	+	,	-	.	/	0	1	2	3	
20/39	4	5	6	7	8	9	:	;	<	=	>	?	@	A	B	C	D	E	F	G
40/59	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	[
60/79	\	]	^	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
80/94	p	q	r	s	t	u	v	w	x	y	z	{		}						
	2A70	2A71	2A72	2A73	2A74	2A75	2A76	2A77	2A78	2A79	2A7A	2A7B	2A7C	2A7D	2A7E					
11区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	
01/19	ā	á	ǎ	à	ē	é	ě	è	í	í	ǐ	ì	ō	ó	ǒ	ò	ū	ú	ǔ	
20/39	à	ā	ǎ	ǎ	ū	ē	ā	m	n	n	h	g								
40/59	2B48	2B49	2B4A	2B4B	2B4C	2B4D	2B4E	2B4F	2B50	2B51	2B52	2B53	2B54	2B55	2B56	2B57	2B58	2B59	2B5A	2B5B
60/79	2B5C	2B5D	2B5E	2B5F	2B60	2B61	2B62	2B63	2B64	2B65	2B66	2B67	2B68	2B69	2B6A	2B6B	2B6C	2B6D	2B6E	2B6F
80/94	2B70	2B71	2B72	2B73	2B74	2B75	2B76	2B77	2B78	2B79	2B7A	2B7B	2B7C	2B7D	2B7E					
12区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	
01/19	丑	弹	怀	惊	惨	鄙	泚	泚	沟	涪	激	涪	涪	涪	涪	涪	涪	涪	涪	
20/39	弁	怀	弹	弹	弹	弹	弹	弹	弹	弹	弹	弹	弹	弹	弹	弹	弹	弹	弹	
40/59	虞	裸	场	场	场	场	场	场	场	场	场	场	场	场	场	场	场	场	场	
60/79	疹	被	被	被	被	被	被	被	被	被	被	被	被	被	被	被	被	被	被	
80/94	踏	被	被	被	被	被	被	被	被	被	被	被	被	被	被	被	被	被	被	

T0814450-93/d106

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 4

13区 01-19	01 钇 2D21	02 铈 2D22	03 铈 2D23	04 铈 2D24	05 铈 2D25	06 铈 2D26	07 铈 2D27	08 铈 2D28	09 铈 2D29	10 铈 2D2A	11 铈 2D2B	12 铈 2D2C	13 铈 2D2D	14 铈 2D2E	15 铈 2D2F	16 铈 2D30	17 铈 2D31	18 铈 2D32	19 铈 2D33	
20-39	20 铈 2D34	21 铈 2D35	22 铈 2D36	23 铈 2D37	24 铈 2D38	25 铈 2D39	26 铈 2D3A	27 铈 2D3B	28 铈 2D3C	29 铈 2D3D	30 铈 2D3E	31 铈 2D3F	32 铈 2D40	33 铈 2D41	34 铈 2D42	35 铈 2D43	36 铈 2D44	37 铈 2D45	38 铈 2D46	39 铈 2D47
40-59	40 铈 2D48	41 铈 2D49	42 铈 2D4A	43 铈 2D4B	44 铈 2D4C	45 铈 2D4D	46 铈 2D4E	47 铈 2D4F	48 铈 2D50	49 铈 2D51	50 铈 2D52	51 铈 2D53	52 铈 2D54	53 铈 2D55	54 铈 2D56	55 铈 2D57	56 铈 2D58	57 铈 2D59	58 铈 2D5A	59 铈 2D5B
60-79	60 铈 2D5C	61 铈 2D5D	62 铈 2D5E	63 铈 2D5F	64 铈 2D60	65 铈 2D61	66 铈 2D62	67 铈 2D63	68 铈 2D64	69 铈 2D65	70 铈 2D66	71 铈 2D67	72 铈 2D68	73 铈 2D69	74 铈 2D6A	75 铈 2D6B	76 铈 2D6C	77 铈 2D6D	78 铈 2D6E	79 铈 2D6F
80-94	80 铈 2D70	81 铈 2D71	82 铈 2D72	83 铈 2D73	84 铈 2D74	85 铈 2D75	86 铈 2D76	87 铈 2D77	88 铈 2D78	89 铈 2D79	90 铈 2D7A	91 铈 2D7B	92 铈 2D7C	93 铈 2D7D	94 铈 2D7E					
14区 01-19	01 铈 2E21	02 铈 2E22	03 铈 2E23	04 铈 2E24	05 铈 2E25	06 铈 2E26	07 铈 2E27	08 铈 2E28	09 铈 2E29	10 铈 2E2A	11 铈 2E2B	12 铈 2E2C	13 铈 2E2D	14 铈 2E2E	15 铈 2E2F	16 铈 2E30	17 铈 2E31	18 铈 2E32	19 铈 2E33	
20-39	20 铈 2E34	21 铈 2E35	22 铈 2E36	23 铈 2E37	24 铈 2E38	25 铈 2E39	26 铈 2E3A	27 铈 2E3B	28 铈 2E3C	29 铈 2E3D	30 铈 2E3E	31 铈 2E3F	32 铈 2E40	33 铈 2E41	34 铈 2E42	35 铈 2E43	36 铈 2E44	37 铈 2E45	38 铈 2E46	39 铈 2E47
40-59	40 铈 2E48	41 铈 2E49	42 铈 2E4A	43 铈 2E4B	44 铈 2E4C	45 铈 2E4D	46 铈 2E4E	47 铈 2E4F	48 铈 2E50	49 铈 2E51	50 铈 2E52	51 铈 2E53	52 铈 2E54	53 铈 2E55	54 铈 2E56	55 铈 2E57	56 铈 2E58	57 铈 2E59	58 铈 2E5A	59 铈 2E5B
60-79	60 铈 2E5C	61 铈 2E5D	62 铈 2E5E	63 铈 2E5F	64 铈 2E60	65 铈 2E61	66 铈 2E62	67 铈 2E63	68 铈 2E64	69 铈 2E65	70 铈 2E66	71 铈 2E67	72 铈 2E68	73 铈 2E69	74 铈 2E6A	75 铈 2E6B	76 铈 2E6C	77 铈 2E6D	78 铈 2E6E	79 铈 2E6F
80-94	80 铈 2E70	81 铈 2E71	82 铈 2E72	83 铈 2E73	84 铈 2E74	85 铈 2E75	86 铈 2E76	87 铈 2E77	88 铈 2E78	89 铈 2E79	90 铈 2E7A	91 铈 2E7B	92 铈 2E7C	93 铈 2E7D	94 铈 2E7E					
15区 01-19	01 铈 2F21	02 铈 2F22	03 铈 2F23	04 铈 2F24	05 铈 2F25	06 铈 2F26	07 铈 2F27	08 铈 2F28	09 铈 2F29	10 铈 2F2A	11 铈 2F2B	12 铈 2F2C	13 铈 2F2D	14 铈 2F2E	15 铈 2F2F	16 铈 2F30	17 铈 2F31	18 铈 2F32	19 铈 2F33	
20-39	20 铈 2F34	21 铈 2F35	22 铈 2F36	23 铈 2F37	24 铈 2F38	25 铈 2F39	26 铈 2F3A	27 铈 2F3B	28 铈 2F3C	29 铈 2F3D	30 铈 2F3E	31 铈 2F3F	32 铈 2F40	33 铈 2F41	34 铈 2F42	35 铈 2F43	36 铈 2F44	37 铈 2F45	38 铈 2F46	39 铈 2F47
40-59	40 铈 2F48	41 铈 2F49	42 铈 2F4A	43 铈 2F4B	44 铈 2F4C	45 铈 2F4D	46 铈 2F4E	47 铈 2F4F	48 铈 2F50	49 铈 2F51	50 铈 2F52	51 铈 2F53	52 铈 2F54	53 铈 2F55	54 铈 2F56	55 铈 2F57	56 铈 2F58	57 铈 2F59	58 铈 2F5A	59 铈 2F5B
60-79	60 铈 2F5C	61 铈 2F5D	62 铈 2F5E	63 铈 2F5F	64 铈 2F60	65 铈 2F61	66 铈 2F62	67 铈 2F63	68 铈 2F64	69 铈 2F65	70 铈 2F66	71 铈 2F67	72 铈 2F68	73 铈 2F69	74 铈 2F6A	75 铈 2F6B	76 铈 2F6C	77 铈 2F6D	78 铈 2F6E	79 铈 2F6F
80-94	80 铈 2F70	81 铈 2F71	82 铈 2F72	83 铈 2F73	84 铈 2F74	85 铈 2F75	86 铈 2F76	87 铈 2F77	88 铈 2F78	89 铈 2F79	90 铈 2F7A	91 铈 2F7B	92 铈 2F7C	93 铈 2F7D	94 铈 2F7E					
16区 01-19	01 铈 3021	02 铈 3022	03 铈 3023	04 铈 3024	05 铈 3025	06 铈 3026	07 铈 3027	08 铈 3028	09 铈 3029	10 铈 302A	11 铈 302B	12 铈 302C	13 铈 302D	14 铈 302E	15 铈 302F	16 铈 3030	17 铈 3031	18 铈 3032	19 铈 3033	
20-39	20 铈 3034	21 铈 3035	22 铈 3036	23 铈 3037	24 铈 3038	25 铈 3039	26 铈 303A	27 铈 303B	28 铈 303C	29 铈 303D	30 铈 303E	31 铈 303F	32 铈 3040	33 铈 3041	34 铈 3042	35 铈 3043	36 铈 3044	37 铈 3045	38 铈 3046	39 铈 3047
40-59	40 铈 3048	41 铈 3049	42 铈 304A	43 铈 304B	44 铈 304C	45 铈 304D	46 铈 304E	47 铈 304F	48 铈 3050	49 铈 3051	50 铈 3052	51 铈 3053	52 铈 3054	53 铈 3055	54 铈 3056	55 铈 3057	56 铈 3058	57 铈 3059	58 铈 305A	59 铈 305B
60-79	60 铈 305C	61 铈 305D	62 铈 305E	63 铈 305F	64 铈 3060	65 铈 3061	66 铈 3062	67 铈 3063	68 铈 3064	69 铈 3065	70 铈 3066	71 铈 3067	72 铈 3068	73 铈 3069	74 铈 306A	75 铈 306B	76 铈 306C	77 铈 306D	78 铈 306E	79 铈 306F
80-94	80 铈 3070	81 铈 3071	82 铈 3072	83 铈 3073	84 铈 3074	85 铈 3075	86 铈 3076	87 铈 3077	88 铈 3078	89 铈 3079	90 铈 307A	91 铈 307B	92 铈 307C	93 铈 307D	94 铈 307E					

T0814460-93/d107

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 5

17区 01-19	01 薄 3121	02 雹 3122	03 保 3123	04 堡 3124	05 饱 3125	06 宝 3126	07 抱 3127	08 报 3128	09 暴 3129	10 豹 312A	11 鲍 312B	12 爆 312C	13 杯 312D	14 碑 312E	15 悲 312F	16 卑 3130	17 北 3131	18 辈 3132	19 背 3133
20-39	贝 3134	钡 3135	倍 3136	狈 3137	惫 3139	焙 313A	被 313B	奔 313C	苯 313D	本 313E	笨 313F	崩 3140	绷 3141	甬 3142	泵 3143	蹯 3144	进 3145	逼 3146	鼻 3147
40-59	比 3148	鄙 3149	笔 314A	彼 314B	蓖 314D	蔽 314E	毕 314F	毙 3150	恣 3151	币 3152	底 3153	痹 3154	闭 3155	敝 3156	弊 3157	必 3158	辟 3159	壁 315A	臂 315B
60-79	避 315C	陛 315D	鞭 315E	边 315F	编 3160	贬 3161	便 3163	变 3164	卞 3165	辨 3166	辩 3167	辩 3168	遍 3169	标 316A	彪 316B	膘 316C	表 316D	璧 316E	憋 316F
80-94	别 3170	瘪 3171	彬 3172	斌 3173	濒 3174	滨 3175	宾 3176	宾 3177	冰 3178	柄 317A	丙 317B	秉 317C	饼 317D	炳 317E					
18区 01-19	01 病 3221	02 并 3222	03 玻 3223	04 菠 3224	05 播 3225	06 拨 3226	07 钵 3227	08 波 3228	09 博 3229	10 勃 322A	11 搏 322B	12 铂 322C	13 箔 322D	14 伯 322E	15 帛 322F	16 舶 3230	17 脖 3231	18 膊 3232	19 渤 3233
20-39	泊 3234	驳 3235	捕 3236	卜 3237	补 3239	埠 323A	不 323B	布 323C	步 323D	簿 323E	部 323F	怖 3240	擦 3241	猜 3242	裁 3243	材 3244	才 3245	财 3246	睬 3247
40-59	睬 3248	采 3249	彩 324A	菜 324B	餐 324C	参 324E	蚕 324F	残 3250	惭 3251	惨 3252	灿 3253	苍 3254	舱 3255	仓 3256	沧 3257	藏 3258	操 3259	糙 325A	槽 325B
60-79	曹 325C	草 325D	厕 325E	策 325F	册 3260	测 3261	层 3262	蹭 3263	插 3264	又 3265	茬 3267	茶 3268	查 3269	碴 326A	捺 326B	察 326C	岔 326D	差 326E	诧 326F
80-94	拆 3270	柴 3271	豺 3272	搀 3273	掺 3274	蝉 3275	馋 3276	讪 3277	缠 3278	铲 3279	产 327A	阐 327B	颤 327C	昌 327D	猖 327E				
19区 01-19	01 场 3321	02 尝 3322	03 常 3323	04 长 3324	05 偿 3325	06 肠 3326	07 厂 3327	08 敞 3328	09 畅 3329	10 唱 332A	11 倡 332B	12 超 332C	13 抄 332D	14 钞 332E	15 朝 332F	16 嘲 3330	17 潮 3331	18 巢 3332	19 吵 3333
20-39	炒 3334	车 3335	扯 3336	掣 3338	彻 3339	澈 333A	郴 333B	臣 333C	辰 333D	尘 333E	晨 333F	忱 3340	沉 3341	陈 3342	趁 3343	衬 3344	撑 3345	称 3346	城 3347
40-59	橙 3348	成 3349	呈 334A	乘 334B	惩 334C	澄 334E	诚 334F	承 3350	逞 3351	骋 3352	秤 3353	吃 3354	痴 3355	持 3356	匙 3357	池 3358	迟 3359	弛 335A	驰 335B
60-79	耻 335C	齿 335D	侈 335E	尺 335F	赤 3360	斥 3361	炽 3363	充 3364	冲 3365	虫 3366	崇 3367	宠 3368	抽 3369	酬 336A	畴 336B	踌 336C	稠 336D	愁 336E	筹 336F
80-94	仇 3370	绸 3371	瞅 3372	丑 3373	臭 3374	初 3375	橱 3377	厨 3378	躇 3379	锄 337A	雏 337B	滁 337C	除 337D	楚 337E					
20区 01-19	01 础 3421	02 储 3422	03 矗 3423	04 搐 3424	05 触 3425	06 处 3426	07 揣 3427	08 川 3428	09 穿 3429	10 椽 342A	11 传 342B	12 船 342C	13 喘 342D	14 串 342E	15 疮 342F	16 窗 3430	17 幢 3431	18 床 3432	19 闯 3433
20-39	创 3434	吹 3435	炊 3436	捶 3437	垂 3439	春 343A	椿 343B	醇 343C	唇 343D	淳 343E	纯 343F	蠢 3440	戳 3441	绰 3442	疵 3443	茨 3444	磁 3445	雌 3446	辞 3447
40-59	慈 3448	瓷 3449	词 344A	此 344B	赐 344C	次 344E	聪 344F	葱 3450	囱 3451	匆 3452	从 3453	丛 3454	凑 3455	粗 3456	醋 3457	簇 3458	促 3459	蹕 345A	篡 345B
60-79	窜 345C	摧 345D	崔 345E	催 345F	脆 3460	粹 3461	淬 3463	翠 3464	村 3465	存 3466	寸 3467	磋 3468	撮 3469	搓 346A	措 346B	挫 346C	错 346D	搭 346E	达 346F
80-94	答 3470	瘩 3471	打 3472	大 3473	呆 3474	歹 3475	傣 3476	戴 3477	殆 3478	代 347A	贷 347B	袋 347C	待 347D	逮 347E					

T0814470-93/d108

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 6

21区 01/19	荡 3534	01 怠 3521	02 耽 3522	03 担 3523	04 丹 3524	05 单 3525	06 郸 3526	07 掸 3527	08 胆 3528	09 旦 3529	10 氮 352A	11 但 352B	12 憚 352C	13 淡 352D	14 诞 352E	15 弹 352F	16 蛋 3530	17 当 3531	18 挡 3532	19 党 3533
20/39	等 3548	档 3535	刀 3536	捣 3537	蹈 3538	倒 3539	岛 353A	祷 353B	导 353C	到 353D	稻 353E	悼 353F	道 3540	盗 3541	德 3542	得 3543	的 3544	蹬 3545	灯 3546	登 3547
40/59	弟 355C	瞪 3549	凳 354A	邓 354B	堤 354C	低 354D	滴 354E	迪 354F	敌 3550	笛 3551	狄 3552	涤 3553	翟 3554	嫡 3555	抵 3556	底 3557	地 3558	蒂 3559	第 355A	帝 355B
60/79	呀 357D	递 355D	缔 355E	颠 355F	掂 3560	滇 3561	碘 3562	点 3563	典 3564	靛 3565	垫 3566	电 3567	佃 3568	甸 3569	店 356A	惦 356B	奠 356C	淀 356D	殿 356E	碉 356F
80/94		雕 357L	凋 3572	刁 3573	掉 3574	吊 3575	钓 3576	调 3577	跌 3578	爹 3579	碟 357A	蝶 357B	迭 357C	谍 357D	叠 357E					
22区 01/19	洞 3634	01 丁 3621	02 盯 3622	03 叮 3623	04 钉 3624	05 顶 3625	06 鼎 3626	07 铎 3627	08 定 3628	09 订 3629	10 丢 362A	11 东 362B	12 冬 362C	13 董 362D	14 懂 362E	15 动 362F	16 栋 3630	17 侗 3631	18 恫 3632	19 冻 3633
20/39	度 3648	兜 3635	抖 3636	斗 3637	陡 3638	豆 3639	逗 363A	痘 363B	都 363C	督 363D	毒 363E	怯 363F	独 3640	读 3641	堵 3642	睹 3643	赌 3644	杜 3645	镀 3646	肚 3647
40/59	盾 365C	渡 3649	妒 364A	端 364B	短 364C	段 364D	段 364E	断 364F	缎 3650	堆 3651	兑 3652	队 3653	对 3654	墩 3655	吨 3656	蹲 3657	敦 3658	顿 3659	囤 365A	钝 365B
60/79	娥 367D	遁 365D	掇 365E	哆 365F	多 3660	夺 3661	垛 3662	躲 3663	朵 3664	蹂 3665	舵 3666	剥 3667	惰 3668	堕 3669	蛾 366A	峨 366B	鹅 366C	俄 366D	额 366E	讹 366F
80/94		恶 3671	厄 3672	扼 3673	遏 3674	鄂 3675	饿 3676	恩 3677	而 3678	儿 3679	耳 367A	尔 367B	饵 367C	洱 367D	二 367E					
23区 01/19	反 3734	01 贰 3721	02 发 3722	03 罚 3723	04 筏 3724	05 伐 3725	06 乏 3726	07 阙 3727	08 法 3728	09 珐 3729	10 藩 372A	11 帆 372B	12 番 372C	13 翻 372D	14 樊 372E	15 矾 372F	16 钒 3730	17 繁 3731	18 凡 3732	19 烦 3733
20/39	啡 3748	返 3735	范 3736	贩 3737	犯 3738	饭 3739	泛 373A	坊 373B	芳 373C	方 373D	肪 373E	房 373F	防 3740	妨 3741	仿 3742	访 3743	纺 3744	放 3745	菲 3746	非 3747
40/59	奋 375C	飞 3749	肥 374A	匪 374B	诽 374C	吠 374D	肺 374E	废 374F	沸 3750	费 3751	芬 3752	汾 3753	吩 3754	氛 3755	分 3756	纷 3757	坟 3758	焚 3759	汾 375A	粉 375B
60/79	佛 377D	份 375D	忿 375E	愤 375F	粪 3760	丰 3761	封 3762	枫 3763	蜂 3764	峰 3765	锋 3766	风 3767	疯 3768	氛 3769	逢 376A	冯 376B	缝 376C	讽 376D	奉 376E	风 376F
80/94		否 3771	夫 3772	敷 3773	肤 3774	孵 3775	扶 3776	拂 3777	辐 3778	幅 3779	氟 377A	符 377B	伏 377C	俘 377D	服 377E					
24区 01/19	复 3834	01 浮 3821	02 涪 3822	03 福 3823	04 袱 3824	05 弗 3825	06 甫 3826	07 抚 3827	08 辅 3828	09 俯 3829	10 釜 382A	11 斧 382B	12 脯 382C	13 腑 382D	14 府 382E	15 腐 382F	16 赴 3830	17 副 3831	18 覆 3832	19 赋 3833
20/39	溉 3848	傅 3835	付 3836	阜 3837	父 3838	腹 3839	负 383A	富 383B	讷 383C	附 383D	妇 383E	缚 383F	咐 3840	腑 3841	嘎 3842	该 3843	改 3844	概 3845	钙 3846	盖 3847
40/59	杠 385C	干 3849	甘 384A	杆 384B	柑 384C	竿 384D	肝 384E	赶 384F	感 3850	秆 3851	敢 3852	赣 3853	冈 3854	刚 3855	钢 3856	缸 3857	肛 3858	纲 3859	岗 385A	港 385B
60/79	葛 387D	篙 385D	皋 385E	高 385F	膏 3860	羔 3861	糕 3862	搞 3863	搞 3864	稿 3865	告 3866	哥 3867	歌 3868	搁 3869	戈 386A	鸽 386B	胳 386C	疙 386D	割 386E	革 386F
80/94		格 3871	蛤 3872	阁 3873	隔 3874	铬 3875	个 3876	各 3877	给 3878	根 3879	跟 387A	耕 387B	更 387C	庚 387D	羹 387E					

T0814480-93/d109

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 7

25区 01/19		01 埂 3921	02 耿 3922	03 梗 3923	04 工 3924	05 攻 3925	06 功 3926	07 恭 3927	08 龚 3928	09 供 3929	10 躬 392A	11 公 392B	12 宫 392C	13 弓 392D	14 巩 392E	15 汞 392F	16 拱 3930	17 贡 3931	18 共 3932	19 钩 3933
20/39	勾 3934	沟 3935	苟 3936	狗 3937	垢 3938	构 3939	购 393A	够 393B	辜 393C	菇 393D	咕 393E	箍 393F	估 3940	沽 3941	孤 3942	姑 3943	鼓 3944	古 3945	蛊 3946	骨 3947
40/59	谷 3948	股 3949	故 394A	顾 394B	固 394C	雇 394D	刮 394E	瓜 394F	刚 3950	寡 3951	挂 3952	褂 3953	乖 3954	拐 3955	怪 3956	棺 3957	关 3958	官 3959	冠 395A	观 395B
60/79	管 395C	馆 395D	罐 395E	惯 395F	灌 3960	贯 3961	光 3962	广 3963	逛 3964	瑰 3965	规 3966	圭 3967	硅 3968	归 3969	龟 396A	轨 396B	鬼 396D	诡 396E	癸 396F	
80/94	桂 3970	柜 3971	跪 3972	责 3973	剑 3974	辊 3975	滚 3976	棍 3977	锅 3978	郭 3979	国 397A	果 397B	裹 397C	过 397D	哈 397E					
26区 01/19		01 骸 3A21	02 孩 3A22	03 海 3A23	04 氦 3A24	05 亥 3A25	06 害 3A26	07 骇 3A27	08 酣 3A28	09 憨 3A29	10 邯 3A2A	11 韩 3A2B	12 含 3A2C	13 涵 3A2D	14 寒 3A2E	15 函 3A2F	16 喊 3A30	17 罕 3A31	18 翰 3A32	19 撼 3A33
20/39	捍 3A34	旱 3A35	憾 3A36	悍 3A37	焊 3A38	汗 3A39	汉 3A3A	夯 3A3B	杭 3A3C	航 3A3D	壕 3A3E	壕 3A3F	豪 3A40	毫 3A41	郝 3A42	好 3A43	耗 3A44	号 3A45	浩 3A46	呵 3A47
40/59	喝 3A48	荷 3A49	荷 3A4A	核 3A4B	禾 3A4C	和 3A4D	何 3A4E	合 3A4F	盒 3A50	貉 3A51	阍 3A52	河 3A53	涸 3A54	赫 3A55	褐 3A56	鹤 3A57	贺 3A58	嘿 3A59	黑 3A5A	痕 3A5B
60/79	很 3A5C	狠 3A5D	恨 3A5E	哼 3A5F	亨 3A60	横 3A61	衡 3A62	恒 3A63	轰 3A64	哄 3A65	烘 3A66	虹 3A67	鸿 3A68	洪 3A69	宏 3A6A	弘 3A6B	喉 3A6D	侯 3A6E	猴 3A6F	
80/94	吼 3A70	厚 3A71	候 3A72	后 3A73	呼 3A74	乎 3A75	忽 3A76	瑚 3A77	壶 3A78	葫 3A79	胡 3A7A	蝴 3A7B	狐 3A7C	糊 3A7D	湖 3A7E					
27区 01/19		01 弧 3B21	02 虎 3B22	03 唬 3B23	04 护 3B24	05 互 3B25	06 沪 3B26	07 户 3B27	08 花 3B28	09 哗 3B29	10 华 3B2A	11 猾 3B2B	12 滑 3B2C	13 画 3B2D	14 划 3B2E	15 化 3B2F	16 话 3B30	17 槐 3B31	18 徊 3B32	19 怀 3B33
20/39	淮 3B34	坏 3B35	欢 3B36	环 3B37	桓 3B38	还 3B39	缓 3B3A	换 3B3B	患 3B3C	唤 3B3D	痪 3B3E	獾 3B3F	焕 3B40	涣 3B41	宦 3B42	幻 3B43	荒 3B44	慌 3B45	黄 3B46	磺 3B47
40/59	蝗 3B48	簧 3B49	皇 3B4A	凰 3B4B	惶 3B4C	煌 3B4D	晃 3B4E	幌 3B4F	恍 3B50	谎 3B51	灰 3B52	挥 3B53	辉 3B54	徽 3B55	恢 3B56	蛔 3B57	回 3B58	毁 3B59	悔 3B5A	慧 3B5B
60/79	卉 3B5C	惠 3B5D	晦 3B5E	贿 3B5F	秽 3B60	会 3B61	烩 3B62	汇 3B63	讳 3B64	海 3B65	绘 3B66	荤 3B67	昏 3B68	婚 3B69	魂 3B6A	浑 3B6B	混 3B6C	豁 3B6D	活 3B6E	伙 3B6F
80/94	火 3B70	获 3B71	或 3B72	惑 3B73	霍 3B74	货 3B75	祸 3B76	击 3B77	圾 3B78	基 3B79	机 3B7A	畸 3B7B	稽 3B7C	积 3B7D	箕 3B7E					
28区 01/19		01 肌 3C21	02 饥 3C22	03 迹 3C23	04 激 3C24	05 讥 3C25	06 鸡 3C26	07 姬 3C27	08 绩 3C28	09 缉 3C29	10 吉 3C2A	11 极 3C2B	12 棘 3C2C	13 辑 3C2D	14 籍 3C2E	15 集 3C2F	16 及 3C30	17 急 3C31	18 疾 3C32	19 汲 3C33
20/39	即 3C34	嫉 3C35	级 3C36	挤 3C37	几 3C38	脊 3C39	己 3C3A	薏 3C3B	技 3C3C	冀 3C3D	季 3C3E	伎 3C3F	祭 3C40	剂 3C41	悸 3C42	济 3C43	寄 3C44	寂 3C45	计 3C46	记 3C47
40/59	既 3C48	忌 3C49	际 3C4A	妓 3C4B	继 3C4C	纪 3C4D	嘉 3C4E	枷 3C4F	夹 3C50	佳 3C51	家 3C52	加 3C53	荚 3C54	颊 3C55	贾 3C56	甲 3C57	钾 3C58	假 3C59	稼 3C5A	价 3C5B
60/79	架 3C5C	驾 3C5D	嫁 3C5E	歼 3C5F	监 3C60	坚 3C61	尖 3C62	笺 3C63	间 3C64	煎 3C65	兼 3C66	肩 3C67	艰 3C68	奸 3C69	缄 3C6A	茧 3C6B	检 3C6C	束 3C6D	碱 3C6E	硷 3C6F
80/94	拣 3C70	捡 3C71	简 3C72	俭 3C73	剪 3C74	减 3C75	荐 3C76	槛 3C77	鉴 3C78	践 3C79	贱 3C7A	见 3C7B	键 3C7C	箭 3C7D	件 3C7E					

T0814490-93/d110

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 8

29区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	健 3D21	舰 3D22	剑 3D23	钱 3D24	渐 3D25	溅 3D26	涧 3D27	建 3D28	僵 3D29	姜 3D2A	将 3D2B	浆 3D2C	江 3D2D	疆 3D2E	蒋 3D2F	浆 3D30	奖 3D31	讲 3D32	匠 3D33
20/39	酱 3D34	降 3D35	蕉 3D36	椒 3D37	焦 3D38	胶 3D3A	交 3D3B	郊 3D3C	浇 3D3D	骄 3D3E	娇 3D3F	嚼 3D40	搅 3D41	绞 3D42	矫 3D43	饶 3D44	脚 3D45	狡 3D46	角 3D47
40/59	饺 3D48	缴 3D49	绞 3D4A	剿 3D4B	教 3D4C	轿 3D4E	较 3D4F	叫 3D50	窖 3D51	揭 3D52	接 3D53	皆 3D54	桔 3D55	街 3D56	阶 3D57	截 3D58	劫 3D59	节 3D5A	桔 3D5B
60/79	杰 3D5C	捷 3D5D	睫 3D5E	竭 3D5F	洁 3D60	结 3D61	姐 3D63	戒 3D64	藉 3D65	芥 3D66	界 3D67	借 3D68	介 3D69	疥 3D6A	诫 3D6B	届 3D6C	巾 3D6D	筋 3D6E	斤 3D6F
80/94	金 3D70	今 3D71	津 3D72	襟 3D73	紧 3D74	锦 3D75	仅 3D76	谨 3D77	进 3D78	靳 3D79	晋 3D7A	禁 3D7B	近 3D7C	烬 3D7D	浸 3D7E				
30区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	尽 3E21	劲 3E22	荆 3E23	兢 3E24	茎 3E25	睛 3E26	晶 3E27	鲸 3E28	京 3E29	惊 3E2A	精 3E2B	梗 3E2C	经 3E2D	井 3E2E	警 3E2F	景 3E30	颈 3E31	静 3E32	境 3E33
20/39	敬 3E34	镜 3E35	径 3E36	痉 3E37	竟 3E38	竞 3E3A	净 3E3B	炯 3E3C	窘 3E3D	揪 3E3E	究 3E3F	纠 3E40	玖 3E41	韭 3E42	久 3E43	灸 3E44	九 3E45	酒 3E46	厩 3E47
40/59	救 3E48	旧 3E49	臼 3E4A	舅 3E4B	咎 3E4C	疚 3E4E	鞠 3E4F	拘 3E50	狙 3E51	疽 3E52	居 3E53	驹 3E54	菊 3E55	局 3E56	咀 3E57	矩 3E58	举 3E59	沮 3E5A	聚 3E5B
60/79	拒 3E5C	据 3E5D	巨 3E5E	具 3E5F	距 3E60	锯 3E62	俱 3E63	句 3E64	惧 3E65	炬 3E66	剧 3E67	捐 3E68	鹃 3E69	娟 3E6A	倦 3E6B	眷 3E6C	卷 3E6D	绢 3E6E	撅 3E6F
80/94	攫 3E70	抉 3E71	掘 3E72	倔 3E73	爵 3E74	觉 3E75	决 3E76	诀 3E77	绝 3E78	均 3E79	钧 3E7B	军 3E7C	君 3E7D	峻 3E7E					
31区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	俊 3F21	竣 3F22	浚 3F23	郡 3F24	骏 3F25	咯 3F26	咖 3F27	卡 3F28	咯 3F29	开 3F2A	揩 3F2B	楷 3F2C	凯 3F2D	慨 3F2E	刊 3F2F	堪 3F30	勘 3F31	坎 3F32	砍 3F33
20/39	看 3F34	康 3F35	慷 3F36	糠 3F37	抗 3F38	亢 3F3A	炕 3F3B	考 3F3C	拷 3F3D	烤 3F3E	靠 3F3F	坷 3F40	苛 3F41	柯 3F42	棵 3F43	磕 3F44	颗 3F45	科 3F46	壳 3F47
40/59	咳 3F48	可 3F49	渴 3F4A	克 3F4B	客 3F4C	课 3F4E	肯 3F4F	啃 3F50	垦 3F51	恳 3F52	坑 3F53	吭 3F54	空 3F55	恐 3F56	孔 3F57	控 3F58	扼 3F59	口 3F5A	扣 3F5B
60/79	寇 3F5C	枯 3F5D	哭 3F5E	窟 3F5F	酷 3F60	库 3F62	裤 3F63	夸 3F64	垮 3F65	垮 3F66	跨 3F67	胯 3F68	块 3F69	筷 3F6A	佻 3F6B	快 3F6C	宽 3F6D	款 3F6E	匡 3F6F
80/94	筐 3F70	狂 3F71	框 3F72	矿 3F73	旷 3F75	况 3F76	亏 3F77	盔 3F78	岗 3F79	窥 3F7A	葵 3F7B	奎 3F7C	魁 3F7D	傀 3F7E					
32区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	馈 4021	愧 4022	溃 4023	坤 4024	昆 4025	捆 4026	困 4027	括 4028	扩 4029	廓 402A	阔 402B	垃 402C	拉 402D	喇 402E	蜡 402F	腊 4030	辣 4031	啦 4032	莱 4033
20/39	来 4034	赖 4035	蓝 4036	婪 4037	栏 4038	拦 4039	阑 403B	兰 403C	澜 403D	斓 403E	揽 403F	览 4040	懒 4041	缆 4042	烂 4043	滥 4044	琅 4045	榔 4046	狼 4047
40/59	廊 4048	郎 4049	朗 404A	浪 404B	捞 404C	劳 404D	老 404F	佬 4050	姥 4051	酪 4052	烙 4053	涝 4054	勒 4055	乐 4056	雷 4057	镭 4058	蕾 4059	磊 405A	累 405B
60/79	儡 405C	垒 405D	擂 405E	肋 405F	类 4060	泪 4061	楞 4062	冷 4064	厘 4065	梨 4066	犁 4067	黎 4068	篱 4069	狸 406A	离 406B	漓 406C	理 406D	李 406E	里 406F
80/94	鲤 4070	礼 4071	莉 4072	荔 4073	吏 4074	栗 4075	丽 4076	厉 4077	励 4078	砾 4079	历 407A	栗 407B	例 407C	俐 407E					

T0814500-93/d111

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 9

33区 01/19	01 痢 4121	02 立 4122	03 粒 4123	04 沥 4124	05 隶 4125	06 力 4126	07 璃 4127	08 哩 4128	09 俩 4129	10 联 412A	11 莲 412B	12 连 412C	13 镰 412D	14 廉 412E	15 伶 412F	16 涟 4130	17 帘 4131	18 敛 4132	19 脸 4133
20/39	链 4134	恋 4135	炼 4136	练 4137	粮 4138	凉 4139	梁 413A	梁 413B	两 413D	辆 413E	量 413F	晾 4140	亮 4141	谅 4142	撩 4143	聊 4144	僚 4145	疗 4146	燎 4147
40/59	寥 4148	辽 4149	潦 414A	了 414B	摺 414C	镣 414D	廖 414E	料 414F	裂 4151	烈 4152	劣 4153	猎 4154	琳 4155	林 4156	磷 4157	霖 4158	临 4159	邻 415A	鳞 415B
60/79	淋 415C	凛 415D	赁 415E	吝 415F	拎 4160	玲 4161	菱 4162	零 4163	龄 4164	伶 4166	羚 4167	凌 4168	灵 4169	陵 416A	岭 416B	领 416C	另 416D	令 416E	溜 416F
80/94	琉 4170	榴 4171	硫 4172	溜 4173	留 4174	刘 4175	瘤 4176	流 4177	柳 4178	六 4179	龙 417A	聋 417B	笼 417D	隆 417E					
34区 01/19	01 隆 4221	02 垄 4222	03 拢 4223	04 陇 4224	05 楼 4225	06 娄 4226	07 搂 4227	08 篓 4228	09 漏 4229	10 陋 422A	11 芦 422B	12 卢 422C	13 颅 422D	14 庐 422E	15 炉 422F	16 掬 4230	17 卤 4231	18 虏 4232	19 鲁 4233
20/39	麓 4234	碌 4235	路 4237	赂 4238	鹿 4239	潞 423A	禄 423B	录 423C	陆 423D	戮 423E	驴 423F	吕 4240	铝 4241	侣 4242	旅 4243	履 4244	屡 4245	缕 4246	虑 4247
40/59	氯 4248	律 4249	率 424A	绿 424B	峦 424C	率 424E	李 424F	漆 4250	卵 4251	乱 4252	掠 4253	略 4254	抡 4255	抡 4256	仑 4257	仑 4258	沦 4259	纶 425A	论 425B
60/79	萝 425C	螺 425D	罗 425E	锣 425F	箩 4260	骡 4262	裸 4263	落 4264	洛 4265	骆 4266	络 4267	妈 4268	麻 4269	玛 426A	码 426B	蚂 426C	马 426D	骂 426E	嘛 426F
80/94	吗 4270	埋 4271	买 4272	麦 4273	卖 4274	迈 4275	脉 4276	慢 4278	蛮 4279	满 427A	蔓 427B	曼 427C	慢 427D	漫 427E					
35区 01/19	01 漫 4321	02 芒 4322	03 茫 4323	04 盲 4324	05 氓 4325	06 忙 4326	07 莽 4327	08 猫 4328	09 茅 4329	10 锚 432A	11 毛 432B	12 矛 432C	13 柳 432D	14 卯 432E	15 茂 432F	16 冒 4330	17 帽 4331	18 貌 4332	19 贸 4333
20/39	么 4334	玫 4335	枚 4336	梅 4337	霉 4338	煤 433A	没 433B	眉 433C	媒 433D	镁 433E	每 433F	美 4340	味 4341	寐 4342	妹 4343	媚 4344	门 4345	闷 4346	们 4347
40/59	萌 4348	蒙 4349	檬 434A	盟 434B	猛 434C	梦 434E	孟 434F	眯 4350	醚 4351	靡 4352	糜 4353	迷 4354	谜 4355	弥 4356	秘 4358	秘 4358	觅 4359	泌 435A	蜜 435B
60/79	密 435C	幂 435D	棉 435E	眠 435F	绵 4360	免 4362	勉 4363	婉 4364	缅 4365	面 4366	苗 4367	描 4368	瞄 4369	藐 436A	秒 436B	渺 436C	庙 436D	妙 436E	蔑 436F
80/94	灭 4370	民 4371	抿 4372	皿 4373	敏 4374	悯 4375	明 4377	螟 4378	鸣 4379	铭 437A	名 437B	命 437C	谬 437D	摸 437E					
36区 01/19	01 摹 4421	02 磨 4422	03 模 4423	04 膜 4424	05 磨 4425	06 摩 4426	07 魔 4427	08 抹 4428	09 末 4429	10 莫 442A	11 墨 442B	12 默 442C	13 沫 442D	14 漠 442E	15 寞 442F	16 陌 4430	17 谋 4431	18 牟 4432	19 某 4433
20/39	拇 4434	牡 4435	姆 4437	母 4438	墓 4439	暮 443A	幕 443B	募 443C	慕 443D	木 443E	目 443F	睦 4440	牧 4441	穆 4442	拿 4443	哪 4444	呐 4445	钠 4446	那 4447
40/59	娜 4448	纳 4449	氛 444A	奶 444C	耐 444D	奈 444E	南 444F	男 4450	难 4451	囊 4452	挠 4453	脑 4454	恼 4455	闹 4456	凇 4457	呢 4458	馁 4459	内 445A	嫩 445B
60/79	能 445C	妮 445D	霓 445E	泥 445F	尼 4460	拟 4462	你 4463	匿 4464	臧 4465	逆 4466	溺 4467	蔫 4468	拈 4469	年 446A	碾 446B	撵 446C	捻 446D	念 446E	娘 446F
80/94	酿 4470	鸟 4471	尿 4472	捏 4473	聂 4474	孽 4475	啮 4476	镊 4477	镍 4478	您 4479	柠 447A	狞 447C	凝 447D	宁 447E					

T0814510-93/gd112

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 10

37区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	拧	泞	牛	扭	钮	组	脓	浓	农	弄	奴	努	怒	女	暖	虐	疟	挪	懦
20/39	糯	诺	欧	鸥	殴	藕	呕	偶	沤	啪	趴	爬	帕	怕	琶	拍	排	牌	徘徊
40/59	湃	派	攀	盘	磐	盼	畔	判	叛	乓	庞	旁	榜	胖	抛	咆	刨	炮	袍
60/79	跑	泡	吓	胚	培	赔	陪	配	佩	沛	喷	盆	砰	抨	烹	澎	彭	蓬	棚
80/94	硼	篷	膨	朋	鹏	捧	碰	坯	砒	霹	披	劈	琵	毗					
	4534	4535	4536	4537	4538	4539	4540	4541	4542	4543	4544	4545	4546	4547	4548	4549	4550	4551	4552
	4548	4549	454A	454B	454C	454D	454E	454F	4550	4551	4552	4553	4554	4555	4556	4557	4558	4559	4560
	455C	455D	455E	455F	4560	4561	4562	4563	4564	4565	4566	4567	4568	4569	456A	456B	456C	456E	456F
	4570	4571	4572	4573	4574	4575	4576	4577	4578	4579	457A	457B	457C	457D	457E				
38区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	啤	脾	疲	皮	匹	痞	僻	屁	譬	篇	偏	片	骗	飘	漂	瓢	票	撇	瞥
20/39	频	贫	品	聘	乒	坪	苹	萍	平	凭	瓶	评	屏	坡	泼	颇	婆	破	魄
40/59	迫	粕	剖	铺	仆	莆	葡	菩	蒲	埔	朴	圃	普	浦	谱	曝	瀑	期	欺
60/79	栖	戚	妻	凄	漆	柒	沏	其	棋	奇	歧	畦	崎	脐	齐	旗	祈	祁	骑
80/94	起	岂	乞	企	启	砌	器	气	迄	弃	汽	泣	迄	掐					
	4634	4635	4636	4637	4638	4639	4640	4641	4642	4643	4644	4645	4646	4647	4648	4649	4650	4651	4652
	4648	4649	464A	464B	464C	464D	464E	464F	4650	4651	4652	4653	4654	4655	4656	4657	4658	4659	4660
	465C	465D	465E	465F	4660	4661	4662	4663	4664	4665	4666	4667	4668	4669	466A	466B	466C	466E	466F
	4670	4671	4672	4673	4674	4675	4676	4677	4678	4679	467A	467B	467C	467D	467E				
39区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	恰	洽	牵	扞	钎	铅	千	迁	签	仟	谦	乾	黔	钱	钡	前	潜	遣	浅
20/39	塹	壑	欠	歉	枪	呛	腔	羌	墙	蔷	强	抢	橇	钦	敲	悄	桥	瞧	乔
40/59	侨	巧	鞘	翘	峭	俏	穹	切	茄	且	怯	窃	钦	侵	亲	秦	琴	勤	芹
60/79	擒	禽	寝	沁	青	轻	倾	卿	清	擎	晴	氩	情	顷	请	庆	琼	穷	秋
80/94	丘	邱	球	求	囚	酋	迥	趋	蛆	曲	驱	屈	驱	渠					
	4734	4735	4736	4737	4738	4739	4740	4741	4742	4743	4744	4745	4746	4747	4748	4749	4750	4751	4752
	4748	4749	474A	474B	474C	474D	474E	474F	4750	4751	4752	4753	4754	4755	4756	4757	4758	4759	4760
	475C	475D	475E	475F	4760	4761	4762	4763	4764	4765	4766	4767	4768	4769	476A	476B	476C	476E	476F
	4770	4771	4772	4773	4774	4775	4776	4777	4778	4779	477A	477B	477C	477D	477E				
40区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	取	娶	龉	趣	去	圈	颧	杈	醛	泉	全	痊	拳	犬	券	劝	缺	快	瘸
20/39	却	鹊	确	雀	裙	群	然	冉	冉	染	瓢	壤	攘	嚷	让	饶	扰	绕	惹
40/59	热	壬	仁	忍	韧	任	认	刃	妊	纫	扔	仍	日	戎	茸	蓉	荣	融	熔
60/79	溶	容	绒	冗	揉	肉	茹	蠕	儒	孺	如	辱	乳	汝	入	褥	阮	阮	蕊
80/94	瑞	锐	闰	润	若	弱	洒	萨	腮	鳃	塞	赛	叁						
	4834	4835	4836	4837	4838	4839	4840	4841	4842	4843	4844	4845	4846	4847	4848	4849	4850	4851	4852
	4848	4849	484A	484B	484C	484D	484E	484F	4850	4851	4852	4853	4854	4855	4856	4857	4858	4859	4860
	485C	485D	485E	485F	4860	4861	4862	4863	4864	4865	4866	4867	4868	4869	486A	486B	486C	486E	486F
	4870	4871	4872	4873	4874	4875	4876	4877	4878	4879	487A	487B	487C	487D	487E				

T0814520-93/d113

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 11

41区 01/19		01 伞 4921	02 散 4922	03 桑 4923	04 噪 4924	05 丧 4925	06 搔 4926	07 骚 4927	08 扫 4928	09 嫂 4929	10 瑟 492A	11 色 492B	12 涩 492C	13 森 492D	14 僧 492E	15 莎 492F	16 砂 4930	17 杀 4931	18 刹 4932	19 沙 4933
20/39	纱 4934	傻 4935	啥 4936	煞 4937	筛 4938	晒 4939	珊 493A	苦 493B	杉 493C	山 493D	删 493E	煽 493F	衫 4940	闪 4941	陕 4942	擅 4943	赡 4944	膳 4945	善 4946	汕 4947
40/59	扇 4948	缮 4949	墒 494A	伤 494B	商 494C	赏 494D	响 494E	上 494F	尚 4950	裳 4951	梢 4952	捎 4953	稍 4954	烧 4955	芍 4956	勺 4957	韶 4958	少 4959	哨 495A	邵 495B
60/79	绍 495C	奢 495D	赊 495E	蛇 495F	舌 4960	舍 4961	赦 4962	摄 4963	射 4964	悞 4965	涉 4966	社 4967	设 4968	伸 4969	申 496A	呻 496B	伸 496C	身 496D	深 496E	娠 496F
80/94	绅 4970	神 4971	沈 4972	审 4973	婶 4974	甚 4975	肾 4976	慎 4977	渗 4978	声 4979	生 497A	甥 497B	牲 497C	升 497D	绳 497E					
42区 01/19		01 省 4A21	02 盛 4A22	03 剩 4A23	04 胜 4A24	05 圣 4A25	06 师 4A26	07 失 4A27	08 狮 4A28	09 施 4A29	10 湿 4A2A	11 诗 4A2B	12 尸 4A2C	13 虱 4A2D	14 十 4A2E	15 石 4A2F	16 拾 4A30	17 时 4A31	18 什 4A32	19 食 4A33
20/39	蚀 4A34	实 4A35	识 4A36	史 4A37	矢 4A38	使 4A39	屎 4A3A	驶 4A3B	始 4A3C	式 4A3D	示 4A3E	士 4A3F	世 4A40	柿 4A41	事 4A42	拭 4A43	誓 4A44	逝 4A45	势 4A46	是 4A47
40/59	嗜 4A48	噬 4A49	适 4A4A	仕 4A4B	侍 4A4C	释 4A4D	饰 4A4E	氏 4A4F	市 4A50	恃 4A51	室 4A52	视 4A53	试 4A54	收 4A55	手 4A56	首 4A57	守 4A58	寿 4A59	授 4A5A	售 4A5B
60/79	受 4A5C	瘦 4A5D	兽 4A5E	蔬 4A5F	枢 4A60	梳 4A61	殊 4A62	抒 4A63	输 4A64	叔 4A65	舒 4A66	淑 4A67	疏 4A68	书 4A69	赎 4A6A	孰 4A6B	熟 4A6C	薯 4A6D	暑 4A6E	曙 4A6F
80/94	署 4A70	蜀 4A71	黍 4A72	鼠 4A73	属 4A74	术 4A75	述 4A76	树 4A77	束 4A78	戍 4A79	竖 4A7A	墅 4A7B	庶 4A7C	数 4A7D	漱 4A7E					
43区 01/19		01 恕 4B21	02 刷 4B22	03 耍 4B23	04 摔 4B24	05 衰 4B25	06 甩 4B26	07 帅 4B27	08 栓 4B28	09 拴 4B29	10 霜 4B2A	11 双 4B2B	12 爽 4B2C	13 谁 4B2D	14 水 4B2E	15 睡 4B2F	16 税 4B30	17 吮 4B31	18 瞬 4B32	19 顺 4B33
20/39	舜 4B34	说 4B35	硕 4B36	烁 4B37	烁 4B38	斯 4B39	撕 4B3A	嘶 4B3B	思 4B3C	私 4B3D	司 4B3E	丝 4B3F	死 4B40	肆 4B41	寺 4B42	嗣 4B43	四 4B44	伺 4B45	似 4B46	饲 4B47
40/59	巳 4B48	松 4B49	耸 4B4A	忪 4B4B	颂 4B4C	送 4B4D	宋 4B4E	讼 4B4F	诵 4B50	搜 4B51	艘 4B52	撇 4B53	嗽 4B54	苏 4B55	酥 4B56	俗 4B57	素 4B58	速 4B59	粟 4B5A	僂 4B5B
60/79	塑 4B5C	溯 4B5D	宿 4B5E	诉 4B5F	肃 4B60	酸 4B61	蒜 4B62	算 4B63	虽 4B64	隋 4B65	艘 4B66	绥 4B67	髓 4B68	碎 4B69	岁 4B6A	穗 4B6B	遂 4B6C	隧 4B6D	祟 4B6E	孙 4B6F
80/94	损 4B70	笋 4B71	蓑 4B72	梭 4B73	唆 4B74	缩 4B75	琐 4B76	索 4B77	锁 4B78	所 4B79	塌 4B7A	他 4B7B	它 4B7C	她 4B7D	塔 4B7E					
44区 01/19		01 獭 4C21	02 挞 4C22	03 蹋 4C23	04 踏 4C24	05 胎 4C25	06 苔 4C26	07 抬 4C27	08 台 4C28	09 泰 4C29	10 酖 4C2A	11 太 4C2B	12 态 4C2C	13 汰 4C2D	14 坍 4C2E	15 摊 4C2F	16 贪 4C30	17 瘫 4C31	18 滩 4C32	19 坛 4C33
20/39	檀 4C34	痰 4C35	潭 4C36	谈 4C37	坦 4C38	袒 4C39	毯 4C3A	袒 4C3B	碳 4C3C	探 4C3D	叹 4C3E	炭 4C3F	汤 4C40	塘 4C41	搪 4C42	堂 4C43	棠 4C44	膛 4C45	唐 4C46	糖 4C47
40/59	倘 4C48	鞘 4C49	淌 4C4A	趟 4C4B	烫 4C4C	掏 4C4D	涛 4C4E	滔 4C4F	绦 4C50	萄 4C51	桃 4C52	逃 4C53	淘 4C54	陶 4C55	讨 4C56	套 4C57	特 4C58	藤 4C59	腾 4C5A	疼 4C5B
60/79	誊 4C5C	梯 4C5D	剔 4C5E	踢 4C5F	绀 4C60	提 4C61	题 4C62	蹄 4C63	啼 4C64	体 4C65	替 4C66	嚏 4C67	惕 4C68	涕 4C69	剃 4C6A	屉 4C6B	天 4C6C	添 4C6D	填 4C6E	田 4C6F
80/94	甜 4C70	恬 4C71	舔 4C72	腆 4C73	挑 4C74	条 4C75	迢 4C76	眺 4C77	跳 4C78	贴 4C79	铁 4C7A	帖 4C7B	厅 4C7C	听 4C7D	烺 4C7E					

T0814530-93/d1114

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 12

45区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	汀	廷	停	亭	庭	挺	艇	通	桐	酏	瞳	同	铜	彤	童	桶	捅	筒	统
20/39	痛	偷	投	透	凸	秃	突	图	徒	途	涂	屠	土	吐	兔	湍	团	推	颓
40/59	腿	蜕	褪	吞	屯	臀	拖	托	脱	鸵	陀	驮	驼	惫	妥	拓	唾	挖	哇
60/79	蛙	洼	娃	瓦	袜	歪	外	弯	湾	玩	顽	丸	烷	完	碗	挽	晚	皖	惋
80/94	宛	婉	万	腕	汪	王	亡	枉	网	往	旺	望	妄	威					
	4D34	4D35	4D36	4D37	4D38	4D39	4D3A	4D3B	4D3C	4D3D	4D3E	4D3F	4D40	4D41	4D42	4D43	4D44	4D45	4D46
	4D48	4D49	4D4A	4D4B	4D4C	4D4D	4D4E	4D4F	4D50	4D51	4D52	4D53	4D54	4D55	4D56	4D57	4D58	4D59	4D5A
	4D5C	4D5D	4D5E	4D5F	4D60	4D61	4D62	4D63	4D64	4D65	4D66	4D67	4D68	4D69	4D6A	4D6B	4D6C	4D6D	4D6E
	4D70	4D71	4D72	4D73	4D74	4D75	4D76	4D77	4D78	4D79	4D7A	4D7B	4D7C	4D7D	4D7E				
46区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	巍	微	危	韦	违	桅	围	唯	惟	为	滩	维	苇	萎	委	伟	伪	尾	纬
20/39	未	蔚	畏	胃	喂	巍	位	渭	谓	尉	慰	卫	瘟	温	蚊	文	闻	纹	吻
40/59	稳	紊	问	翁	瓮	挝	蜗	渭	窝	我	斡	卧	握	沃	巫	鸣	钨	乌	污
60/79	诬	屋	无	梧	吾	吴	毋	武	五	捂	午	舞	伍	侮	坞	戊	雾	晤	物
80/94	勿	务	悟	昔	熙	析	西	硒	矽	晰	嘻	吸	锡	牺					
	4E34	4E35	4E36	4E37	4E38	4E39	4E3A	4E3B	4E3C	4E3D	4E3E	4E3F	4E40	4E41	4E42	4E43	4E44	4E45	4E46
	4E48	4E49	4E4A	4E4B	4E4C	4E4D	4E4E	4E4F	4E50	4E51	4E52	4E53	4E54	4E55	4E56	4E57	4E58	4E59	4E5A
	4E5C	4E5D	4E5E	4E5F	4E60	4E61	4E62	4E63	4E64	4E65	4E66	4E67	4E68	4E69	4E6A	4E6B	4E6C	4E6D	4E6E
	4E70	4E71	4E72	4E73	4E74	4E75	4E76	4E77	4E78	4E79	4E7A	4E7B	4E7C	4E7D	4E7E				
47区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	稀	息	希	悉	膝	夕	惜	熄	烯	溪	汐	犀	檄	袭	席	习	媳	喜	铤
20/39	洗	系	戏	细	膳	虾	匣	霞	辖	暇	峡	侠	狭	下	厦	夏	吓	掀	铍
40/59	先	仙	鲜	咸	贤	衍	舷	闲	涎	弦	嫌	显	险	现	献	县	腺	馅	羨
60/79	宪	陷	限	相	厢	镶	香	箱	襄	湘	乡	翔	祥	详	想	响	享	项	巷
80/94	橡	像	象	萧	硝	霄	削	哮	嚣	销	消	宵	淆	晓					
	4F34	4F35	4F36	4F37	4F38	4F39	4F3A	4F3B	4F3C	4F3D	4F3E	4F3F	4F40	4F41	4F42	4F43	4F44	4F45	4F46
	4F48	4F49	4F4A	4F4B	4F4C	4F4D	4F4E	4F4F	4F50	4F51	4F52	4F53	4F54	4F55	4F56	4F57	4F58	4F59	4F5A
	4F5C	4F5D	4F5E	4F5F	4F60	4F61	4F62	4F63	4F64	4F65	4F66	4F67	4F68	4F69	4F6A	4F6B	4F6C	4F6D	4F6E
	4F70	4F71	4F72	4F73	4F74	4F75	4F76	4F77	4F78	4F79	4F7A	4F7B	4F7C	4F7D	4F7E				
48区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	小	孝	校	肖	啸	笑	效	楔	些	歇	蝎	鞋	协	挟	携	邪	斜	胁	谐
20/39	写	卸	蟹	懈	泄	泻	谢	屑	薪	芯	锌	欣	辛	新	忻	心	信	衅	星
40/59	腥	猩	兴	刑	型	形	邢	行	醒	幸	杏	性	姓	兄	凶	匈	甸	汹	雄
60/79	熊	休	羞	朽	嗅	锈	秀	袖	绣	墟	戊	需	虚	嘘	须	徐	许	蓄	酗
80/94	叙	旭	畜	恤	絮	婿	绪	续	轩	喧	宣	悬	旋	玄					
	5034	5035	5036	5037	5038	5039	503A	503B	503C	503D	503E	503F	5040	5041	5042	5043	5044	5045	5046
	5048	5049	504A	504B	504C	504D	504E	504F	5050	5051	5052	5053	5054	5055	5056	5057	5058	5059	505A
	505C	505D	505E	505F	5060	5061	5062	5063	5064	5065	5066	5067	5068	5069	506A	506B	506C	506D	506E
	5070	5071	5072	5073	5074	5075	5076	5077	5078	5079	507A	507B	507C	507D	507E				

T0814540-93/d1115

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 13

49区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	选	癣	眩	绚	靴	薛	学	穴	雪	血	勋	熏	循	旬	询	寻	驯	巡	殉
20/39	训	讯	逊	迅	压	押	鸦	鸭	呀	丫	芽	牙	蚜	崖	衙	涯	雅	哑	亚
40/59	訝	焉	咽	烟	淹	盐	严	研	蜒	岩	延	言	颜	阎	炎	奄	掩	眼	
60/79	衍	演	艳	燕	厌	砚	雁	唁	彦	焰	宴	谚	验	殃	央	鸯	秧	杨	扬
80/94	佯	场	羊	阳	氧	仰	痒	养	样	漾	邀	腰	妖	瑶					
	5121	5122	5123	5124	5125	5126	5127	5128	5129	512A	512B	512C	512D	512E	512F	5130	5131	5132	5133
20/39	汛	训	逊	迅	压	押	鸦	鸭	呀	丫	芽	牙	蚜	崖	衙	涯	雅	哑	亚
40/59	訝	焉	咽	烟	淹	盐	严	研	蜒	岩	延	言	颜	阎	炎	奄	掩	眼	
60/79	衍	演	艳	燕	厌	砚	雁	唁	彦	焰	宴	谚	验	殃	央	鸯	秧	杨	扬
80/94	佯	场	羊	阳	氧	仰	痒	养	样	漾	邀	腰	妖	瑶					
	5134	5135	5136	5137	5138	513A	513B	513C	513D	513E	513F	5140	5141	5142	5143	5144	5145	5146	5147
50区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	播	尧	遙	窑	遥	姚	咬	舀	药	要	耀	椰	噎	耶	爷	野	冶	也	页
20/39	掖	业	曳	腋	夜	液	一	壹	医	揖	铤	依	伊	衣	颐	夷	遗	移	仪
40/59	胰	疑	沂	姨	彝	椅	蚁	倚	己	乙	矣	以	艺	抑	易	邑	屹	亿	役
60/79	臆	逸	肄	亦	裔	意	毅	忆	义	益	溢	诣	议	谊	译	异	翼	翌	绎
80/94	茵	荫	因	音	阴	姻	吟	银	淫	寅	饮	尹	引	隐					
	5221	5222	5223	5224	5225	5226	5227	5228	5229	522A	522B	522C	522D	522E	522F	5230	5231	5232	5233
20/39	掖	业	曳	腋	夜	液	一	壹	医	揖	铤	依	伊	衣	颐	夷	遗	移	仪
40/59	胰	疑	沂	姨	彝	椅	蚁	倚	己	乙	矣	以	艺	抑	易	邑	屹	亿	役
60/79	臆	逸	肄	亦	裔	意	毅	忆	义	益	溢	诣	议	谊	译	异	翼	翌	绎
80/94	茵	荫	因	音	阴	姻	吟	银	淫	寅	饮	尹	引	隐					
	5234	5235	5236	5237	5238	523A	523B	523C	523D	523E	523F	5240	5241	5242	5243	5244	5245	5246	5247
51区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	印	英	櫻	嬰	鷹	應	纓	莹	螢	營	荧	蝇	迎	贏	盈	影	穎	硬	映
20/39	哟	拥	佣	痈	庸	雍	踊	蛹	咏	泳	涌	永	愿	勇	用	幽	优	悠	忧
40/59	尤	由	邮	犹	油	游	酉	有	友	右	佑	釉	诱	又	幼	迂	淤	于	孟
60/79	榆	虞	愚	余	俞	逾	鱼	愉	渝	渔	隅	予	娱	雨	与	屿	禹	宇	语
80/94	羽	玉	域	郁	吁	遇	喻	峪	御	愈	欲	狱	育	誉					
	5321	5322	5323	5324	5325	5326	5327	5328	5329	532A	532B	532C	532D	532E	532F	5330	5331	5332	5333
20/39	哟	拥	佣	痈	庸	雍	踊	蛹	咏	泳	涌	永	愿	勇	用	幽	优	悠	忧
40/59	尤	由	邮	犹	油	游	酉	有	友	右	佑	釉	诱	又	幼	迂	淤	于	孟
60/79	榆	虞	愚	余	俞	逾	鱼	愉	渝	渔	隅	予	娱	雨	与	屿	禹	宇	语
80/94	羽	玉	域	郁	吁	遇	喻	峪	御	愈	欲	狱	育	誉					
	5334	5335	5336	5337	5338	533A	533B	533C	533D	533E	533F	5340	5341	5342	5343	5344	5345	5346	5347
52区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	浴	寓	裕	预	豫	驭	驾	渊	冤	元	垣	袁	原	援	猿	园	员	圆	猿
20/39	源	缘	苑	愿	怨	院	曰	约	越	跃	钥	岳	粤	月	悦	阅	耘	云	郢
40/59	匀	陨	允	蕴	酝	晕	韵	孕	匝	砸	杂	栽	哉	灾	宰	载	再	在	咱
60/79	攢	暂	赞	脏	葬	遭	槽	茁	藻	枣	早	澡	蚤	躁	噪	造	皂	灶	爆
80/94	责	择	则	泽	贼	怎	憎	曾	赠	扎	喳	渣	札	扎					
	5421	5422	5423	5424	5425	5426	5427	5428	5429	542A	542B	542C	542D	542E	542F	5430	5431	5432	5433
20/39	源	缘	苑	愿	怨	院	曰	约	越	跃	钥	岳	粤	月	悦	阅	耘	云	郢
40/59	匀	陨	允	蕴	酝	晕	韵	孕	匝	砸	杂	栽	哉	灾	宰	载	再	在	咱
60/79	攢	暂	赞	脏	葬	遭	槽	茁	藻	枣	早	澡	蚤	躁	噪	造	皂	灶	爆
80/94	责	择	则	泽	贼	怎	憎	曾	赠	扎	喳	渣	札	扎					
	5434	5435	5436	5437	5438	543A	543B	543C	543D	543E	543F	5440	5441	5442	5443	5444	5445	5446	5447

T0814550-93/d116

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 14

53区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	区	侧	闸	眨	栅	榨	乍	炸	诈	摘	斋	宅	窄	债	寨	瞻	毡	詹	粘
20/39	沾	盏	斩	辗	崭	蘸	栈	占	战	站	湛	绽	樟	章	彰	漳	张	掌	涨
40/59	杖	丈	帐	仗	胀	瘴	障	招	昭	找	沼	赵	照	罩	兆	肇	召	遮	折
60/79	哲	塾	辙	者	储	蔗	浙	珍	斟	真	甄	砧	臻	贞	针	侦	枕	疹	诊
80/94	震	振	镇	阵	蒸	挣	征	狰	争	怔	整	拯	正	政					
	5570	5571	5572	5573	5574	5575	5576	5577	5578	5579	557A	557B	557C	557D	557E				
54区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	区	幀	症	郑	证	芝	枝	支	吱	蚰	知	肢	脂	汁	织	职	直	植	殖
20/39	执	值	侄	址	指	止	趾	只	旨	纸	志	摯	掷	至	置	帜	峙	制	智
40/59	秩	稚	质	炙	痔	滞	治	室	中	盅	忠	钟	衷	终	肿	重	仲	众	舟
60/79	周	州	洲	迨	粥	轴	肘	帚	咒	皱	宙	昼	骤	珠	株	朱	猪	诸	诛
80/94	逐	竹	烛	煮	拄	嘱	嘱	主	著	柱	助	蛀	贮	铸	筑				
	5670	5671	5672	5673	5674	5675	5676	5677	5678	5679	567A	567B	567C	567D	567E				
55区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	区	住	注	祝	驻	抓	爪	拽	专	砖	转	撰	赚	篆	桩	庄	妆	撞	壮
20/39	状	椎	锥	追	赘	坠	缀	淳	准	捉	拙	卓	桌	琢	茁	酌	着	灼	浊
40/59	兹	咨	资	姿	滋	淄	孜	紫	仔	籽	滓	子	自	渍	字	鬃	踪	宗	综
60/79	总	纵	邹	走	奏	揍	租	足	卒	族	祖	诅	阻	组	钻	纂	醉	最	罪
80/94	尊	遵	昨	左	佐	柞	倣	作	坐	座									
	5770	5771	5772	5773	5774	5775	5776	5777	5778	5779	577A	577B	577C	577D	577E				
56区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	区	亍	丌	兀	丐	甘	卅	丕	亘	丞	鬲	舛	噩	乚	乚	乚	乚	乚	乚
20/39	厄	氏	凶	胤	廴	毓	峯	叢	亘	丞	鬲	舛	噩	乚	乚	乚	乚	乚	乚
40/59	厝	庀	厥	廛	廛	廛	廛	廛	廛	廛	廛	廛	廛	廛	廛	廛	廛	廛	廛
60/79	割	刺	剡	剡	剡	剡	剡	剡	剡	剡	剡	剡	剡	剡	剡	剡	剡	剡	剡
80/94	仞	伋	仇	仇	仇	仇	仇	仇	仇	仇	仇	仇	仇	仇	仇	仇	仇	仇	仇
	5870	5871	5872	5873	5874	5875	5876	5877	5878	5879	587A	587B	587C	587D	587E				

T0814560-93/d117

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 15

57区 01/19	01 佟 5921	02 佗 5922	03 佢 5923	04 伽 5924	05 使 5925	06 佢 5926	07 侑 5927	08 侑 5928	09 侃 5929	10 侏 592A	11 侑 592B	12 佻 592C	13 侑 592D	14 佻 592E	15 侏 592F	16 侑 5930	17 侑 5931	18 侑 5932	19 侑 5933
20/39	侏 5934	侏 5935	侏 5936	侏 5937	侏 5938	侏 5939	侏 593A	侏 593B	侏 593C	侏 593D	侏 593E	侏 593F	侏 5940	侏 5941	侏 5942	侏 5943	侏 5944	侏 5945	侏 5946
40/59	侏 5947	侏 5948	侏 5949	侏 594A	侏 594B	侏 594C	侏 594D	侏 594E	侏 594F	侏 5950	侏 5951	侏 5952	侏 5953	侏 5954	侏 5955	侏 5956	侏 5957	侏 5958	侏 5959
60/79	侏 595A	侏 595B	侏 595C	侏 595D	侏 595E	侏 595F	侏 5960	侏 5961	侏 5962	侏 5963	侏 5964	侏 5965	侏 5966	侏 5967	侏 5968	侏 5969	侏 596A	侏 596B	侏 596C
80/94	侏 596D	侏 596E	侏 596F	侏 596G	侏 596H	侏 596I	侏 596J	侏 596K	侏 596L	侏 596M	侏 596N	侏 596O	侏 596P	侏 596Q	侏 596R	侏 596S	侏 596T	侏 596U	侏 596V
58区 01/19	01 侏 5A21	02 侏 5A22	03 侏 5A23	04 侏 5A24	05 侏 5A25	06 侏 5A26	07 侏 5A27	08 侏 5A28	09 侏 5A29	10 侏 5A2A	11 侏 5A2B	12 侏 5A2C	13 侏 5A2D	14 侏 5A2E	15 侏 5A2F	16 侏 5A30	17 侏 5A31	18 侏 5A32	19 侏 5A33
20/39	侏 5A34	侏 5A35	侏 5A36	侏 5A37	侏 5A38	侏 5A39	侏 5A3A	侏 5A3B	侏 5A3C	侏 5A3D	侏 5A3E	侏 5A3F	侏 5A40	侏 5A41	侏 5A42	侏 5A43	侏 5A44	侏 5A45	侏 5A46
40/59	侏 5A47	侏 5A48	侏 5A49	侏 5A4A	侏 5A4B	侏 5A4C	侏 5A4D	侏 5A4E	侏 5A4F	侏 5A50	侏 5A51	侏 5A52	侏 5A53	侏 5A54	侏 5A55	侏 5A56	侏 5A57	侏 5A58	侏 5A59
60/79	侏 5A5A	侏 5A5B	侏 5A5C	侏 5A5D	侏 5A5E	侏 5A5F	侏 5A60	侏 5A61	侏 5A62	侏 5A63	侏 5A64	侏 5A65	侏 5A66	侏 5A67	侏 5A68	侏 5A69	侏 5A6A	侏 5A6B	侏 5A6C
80/94	侏 5A6D	侏 5A6E	侏 5A6F	侏 5A6G	侏 5A6H	侏 5A6I	侏 5A6J	侏 5A6K	侏 5A6L	侏 5A6M	侏 5A6N	侏 5A6O	侏 5A6P	侏 5A6Q	侏 5A6R	侏 5A6S	侏 5A6T	侏 5A6U	侏 5A6V
59区 01/19	01 鄯 5B21	02 鄯 5B22	03 鄯 5B23	04 鄯 5B24	05 鄯 5B25	06 鄯 5B26	07 鄯 5B27	08 鄯 5B28	09 鄯 5B29	10 鄯 5B2A	11 鄯 5B2B	12 鄯 5B2C	13 鄯 5B2D	14 鄯 5B2E	15 鄯 5B2F	16 鄯 5B30	17 鄯 5B31	18 鄯 5B32	19 鄯 5B33
20/39	鄯 5B34	鄯 5B35	鄯 5B36	鄯 5B37	鄯 5B38	鄯 5B39	鄯 5B3A	鄯 5B3B	鄯 5B3C	鄯 5B3D	鄯 5B3E	鄯 5B3F	鄯 5B40	鄯 5B41	鄯 5B42	鄯 5B43	鄯 5B44	鄯 5B45	鄯 5B46
40/59	鄯 5B47	鄯 5B48	鄯 5B49	鄯 5B4A	鄯 5B4B	鄯 5B4C	鄯 5B4D	鄯 5B4E	鄯 5B4F	鄯 5B50	鄯 5B51	鄯 5B52	鄯 5B53	鄯 5B54	鄯 5B55	鄯 5B56	鄯 5B57	鄯 5B58	鄯 5B59
60/79	鄯 5B5A	鄯 5B5B	鄯 5B5C	鄯 5B5D	鄯 5B5E	鄯 5B5F	鄯 5B60	鄯 5B61	鄯 5B62	鄯 5B63	鄯 5B64	鄯 5B65	鄯 5B66	鄯 5B67	鄯 5B68	鄯 5B69	鄯 5B6A	鄯 5B6B	鄯 5B6C
80/94	鄯 5B6D	鄯 5B6E	鄯 5B6F	鄯 5B6G	鄯 5B6H	鄯 5B6I	鄯 5B6J	鄯 5B6K	鄯 5B6L	鄯 5B6M	鄯 5B6N	鄯 5B6O	鄯 5B6P	鄯 5B6Q	鄯 5B6R	鄯 5B6S	鄯 5B6T	鄯 5B6U	鄯 5B6V
60区 01/19	01 芫 5C21	02 芫 5C22	03 芫 5C23	04 芫 5C24	05 芫 5C25	06 芫 5C26	07 芫 5C27	08 芫 5C28	09 芫 5C29	10 芫 5C2A	11 芫 5C2B	12 芫 5C2C	13 芫 5C2D	14 芫 5C2E	15 芫 5C2F	16 芫 5C30	17 芫 5C31	18 芫 5C32	19 芫 5C33
20/39	芫 5C34	芫 5C35	芫 5C36	芫 5C37	芫 5C38	芫 5C39	芫 5C3A	芫 5C3B	芫 5C3C	芫 5C3D	芫 5C3E	芫 5C3F	芫 5C40	芫 5C41	芫 5C42	芫 5C43	芫 5C44	芫 5C45	芫 5C46
40/59	芫 5C47	芫 5C48	芫 5C49	芫 5C4A	芫 5C4B	芫 5C4C	芫 5C4D	芫 5C4E	芫 5C4F	芫 5C50	芫 5C51	芫 5C52	芫 5C53	芫 5C54	芫 5C55	芫 5C56	芫 5C57	芫 5C58	芫 5C59
60/79	芫 5C5A	芫 5C5B	芫 5C5C	芫 5C5D	芫 5C5E	芫 5C5F	芫 5C60	芫 5C61	芫 5C62	芫 5C63	芫 5C64	芫 5C65	芫 5C66	芫 5C67	芫 5C68	芫 5C69	芫 5C6A	芫 5C6B	芫 5C6C
80/94	芫 5C6D	芫 5C6E	芫 5C6F	芫 5C6G	芫 5C6H	芫 5C6I	芫 5C6J	芫 5C6K	芫 5C6L	芫 5C6M	芫 5C6N	芫 5C6O	芫 5C6P	芫 5C6Q	芫 5C6R	芫 5C6S	芫 5C6T	芫 5C6U	芫 5C6V

T0814570-93/d118

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 16

61区 01/19	01 5D21	02 5D22	03 5D23	04 5D24	05 5D25	06 5D26	07 5D27	08 5D28	09 5D29	10 5D2A	11 5D2B	12 5D2C	13 5D2D	14 5D2E	15 5D2F	16 5D30	17 5D31	18 5D32	19 5D33
20/39	萆 5D34	荻 5D35	荻 5D36	荻 5D37	荻 5D38	荻 5D39	荻 5D3A	荻 5D3B	荻 5D3C	荻 5D3D	荻 5D3E	荻 5D3F	荻 5D40	荻 5D41	荻 5D42	荻 5D43	荻 5D44	荻 5D45	荻 5D46
40/59	荻 5D47	荻 5D48	荻 5D49	荻 5D4A	荻 5D4B	荻 5D4C	荻 5D4D	荻 5D4E	荻 5D4F	荻 5D50	荻 5D51	荻 5D52	荻 5D53	荻 5D54	荻 5D55	荻 5D56	荻 5D57	荻 5D58	荻 5D59
60/79	荻 5D5A	荻 5D5B	荻 5D5C	荻 5D5D	荻 5D5E	荻 5D5F	荻 5D60	荻 5D61	荻 5D62	荻 5D63	荻 5D64	荻 5D65	荻 5D66	荻 5D67	荻 5D68	荻 5D69	荻 5D6A	荻 5D6B	荻 5D6C
80/94	荻 5D6D	荻 5D6E	荻 5D6F	荻 5D70	荻 5D71	荻 5D72	荻 5D73	荻 5D74	荻 5D75	荻 5D76	荻 5D77	荻 5D78	荻 5D79	荻 5D7A	荻 5D7B	荻 5D7C	荻 5D7D	荻 5D7E	荻 5D7F
62区 01/19	01 5E21	02 5E22	03 5E23	04 5E24	05 5E25	06 5E26	07 5E27	08 5E28	09 5E29	10 5E2A	11 5E2B	12 5E2C	13 5E2D	14 5E2E	15 5E2F	16 5E30	17 5E31	18 5E32	19 5E33
20/39	藁 5E34	藁 5E35	藁 5E36	藁 5E37	藁 5E38	藁 5E39	藁 5E3A	藁 5E3B	藁 5E3C	藁 5E3D	藁 5E3E	藁 5E3F	藁 5E40	藁 5E41	藁 5E42	藁 5E43	藁 5E44	藁 5E45	藁 5E46
40/59	藁 5E47	藁 5E48	藁 5E49	藁 5E4A	藁 5E4B	藁 5E4C	藁 5E4D	藁 5E4E	藁 5E4F	藁 5E50	藁 5E51	藁 5E52	藁 5E53	藁 5E54	藁 5E55	藁 5E56	藁 5E57	藁 5E58	藁 5E59
60/79	藁 5E5A	藁 5E5B	藁 5E5C	藁 5E5D	藁 5E5E	藁 5E5F	藁 5E60	藁 5E61	藁 5E62	藁 5E63	藁 5E64	藁 5E65	藁 5E66	藁 5E67	藁 5E68	藁 5E69	藁 5E6A	藁 5E6B	藁 5E6C
80/94	藁 5E6D	藁 5E6E	藁 5E6F	藁 5E70	藁 5E71	藁 5E72	藁 5E73	藁 5E74	藁 5E75	藁 5E76	藁 5E77	藁 5E78	藁 5E79	藁 5E7A	藁 5E7B	藁 5E7C	藁 5E7D	藁 5E7E	藁 5E7F
63区 01/19	01 5F21	02 5F22	03 5F23	04 5F24	05 5F25	06 5F26	07 5F27	08 5F28	09 5F29	10 5F2A	11 5F2B	12 5F2C	13 5F2D	14 5F2E	15 5F2F	16 5F30	17 5F31	18 5F32	19 5F33
20/39	叭 5F34	叭 5F35	叭 5F36	叭 5F37	叭 5F38	叭 5F39	叭 5F3A	叭 5F3B	叭 5F3C	叭 5F3D	叭 5F3E	叭 5F3F	叭 5F40	叭 5F41	叭 5F42	叭 5F43	叭 5F44	叭 5F45	叭 5F46
40/59	叭 5F47	叭 5F48	叭 5F49	叭 5F4A	叭 5F4B	叭 5F4C	叭 5F4D	叭 5F4E	叭 5F4F	叭 5F50	叭 5F51	叭 5F52	叭 5F53	叭 5F54	叭 5F55	叭 5F56	叭 5F57	叭 5F58	叭 5F59
60/79	叭 5F5A	叭 5F5B	叭 5F5C	叭 5F5D	叭 5F5E	叭 5F5F	叭 5F60	叭 5F61	叭 5F62	叭 5F63	叭 5F64	叭 5F65	叭 5F66	叭 5F67	叭 5F68	叭 5F69	叭 5F6A	叭 5F6B	叭 5F6C
80/94	叭 5F6D	叭 5F6E	叭 5F6F	叭 5F70	叭 5F71	叭 5F72	叭 5F73	叭 5F74	叭 5F75	叭 5F76	叭 5F77	叭 5F78	叭 5F79	叭 5F7A	叭 5F7B	叭 5F7C	叭 5F7D	叭 5F7E	叭 5F7F
64区 01/19	01 6021	02 6022	03 6023	04 6024	05 6025	06 6026	07 6027	08 6028	09 6029	10 602A	11 602B	12 602C	13 602D	14 602E	15 602F	16 6030	17 6031	18 6032	19 6033
20/39	啞 6034	啞 6035	啞 6036	啞 6037	啞 6038	啞 6039	啞 603A	啞 603B	啞 603C	啞 603D	啞 603E	啞 603F	啞 6040	啞 6041	啞 6042	啞 6043	啞 6044	啞 6045	啞 6046
40/59	啞 6047	啞 6048	啞 6049	啞 604A	啞 604B	啞 604C	啞 604D	啞 604E	啞 604F	啞 6050	啞 6051	啞 6052	啞 6053	啞 6054	啞 6055	啞 6056	啞 6057	啞 6058	啞 6059
60/79	啞 605A	啞 605B	啞 605C	啞 605D	啞 605E	啞 605F	啞 6060	啞 6061	啞 6062	啞 6063	啞 6064	啞 6065	啞 6066	啞 6067	啞 6068	啞 6069	啞 606A	啞 606B	啞 606C
80/94	啞 606D	啞 606E	啞 606F	啞 6070	啞 6071	啞 6072	啞 6073	啞 6074	啞 6075	啞 6076	啞 6077	啞 6078	啞 6079	啞 607A	啞 607B	啞 607C	啞 607D	啞 607E	啞 607F

T0814580-93/d119

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 17

65 区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	峁	峁	峁	嶂	幞	幡	岌	岌	岌	岐	岨	岨	岨	岨	岨	岨	岨	岨	岨
20/39	崇	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬
40/59	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬
60/79	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬
80/94	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬	岬
66 区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯
20/39	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯
40/59	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯
60/79	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯
80/94	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯	猯
67 区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃
20/39	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃
40/59	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃
60/79	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃
80/94	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃	悃
68 区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄
20/39	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄
40/59	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄
60/79	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄
80/94	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄	洄

T0814590-93/d120

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 18

69区	01 11	宥 6534	20 38	19 充 6531
	40 58	迦 6548	40 58	18 一 6531
	60 78	遴 6550	60 78	17 灏 6531
	80 94	屨 6570	80 94	16 一 6531
70区	01 19		01 19	15 充 6531
	20 39	媧 6634	20 39	14 灏 6531
	40 59	媛 6648	40 59	13 灏 6531
	60 79	孽 6650	60 79	12 灏 6531
	80 94	鸞 6670	80 94	11 灏 6531
81区	01 15		01 15	10 灏 6531
	20 35	竊 6734	20 35	09 灏 6531
	40 55	竊 6748	40 55	08 灏 6531
	60 75	讖 6750	60 75	07 灏 6531
	80 94	耶 6770	80 94	06 灏 6531
92区	01 19		01 19	05 灏 6531
	20 39	路 6834	20 39	04 灏 6531
	40 59	枳 6848	40 59	03 灏 6531
	60 79	抵 6850	60 79	02 灏 6531
	80 94	素 6870	80 94	01 灏 6531

T0814600-93/d121

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 19

73区 01/19	01 椶 6921	02 槿 6922	03 棕 6923	04 椶 6924	05 槿 6925	06 棣 6926	07 椶 6927	08 棣 6928	09 椶 6929	10 椶 692A	11 椶 692B	12 椶 692C	13 椶 692D	14 椶 692E	15 椶 692F	16 椶 6930	17 椶 6931	18 椶 6932	19 椶 6933
20/39	椶 6934	椶 6935	椶 6936	椶 6937	椶 6938	椶 6939	椶 693A	椶 693B	椶 693C	椶 693D	椶 693E	椶 693F	椶 6940	椶 6941	椶 6942	椶 6943	椶 6944	椶 6945	椶 6946
40/59	椶 6947	椶 6948	椶 6949	椶 694A	椶 694B	椶 694C	椶 694D	椶 694E	椶 694F	椶 6950	椶 6951	椶 6952	椶 6953	椶 6954	椶 6955	椶 6956	椶 6957	椶 6958	椶 6959
60/79	椶 695C	椶 695D	椶 695E	椶 695F	椶 6960	椶 6961	椶 6962	椶 6963	椶 6964	椶 6965	椶 6966	椶 6967	椶 6968	椶 6969	椶 696A	椶 696B	椶 696C	椶 696D	椶 696E
80/94	椶 6970	椶 6971	椶 6972	椶 6973	椶 6974	椶 6975	椶 6976	椶 6977	椶 6978	椶 6979	椶 697A	椶 697B	椶 697C	椶 697D	椶 697E				
74区 01/19	01 椶 6A21	02 椶 6A22	03 椶 6A23	04 椶 6A24	05 椶 6A25	06 椶 6A26	07 椶 6A27	08 椶 6A28	09 椶 6A29	10 椶 6A2A	11 椶 6A2B	12 椶 6A2C	13 椶 6A2D	14 椶 6A2E	15 椶 6A2F	16 椶 6A30	17 椶 6A31	18 椶 6A32	19 椶 6A33
20/39	椶 6A34	椶 6A35	椶 6A36	椶 6A37	椶 6A38	椶 6A39	椶 6A3A	椶 6A3B	椶 6A3C	椶 6A3D	椶 6A3E	椶 6A3F	椶 6A40	椶 6A41	椶 6A42	椶 6A43	椶 6A44	椶 6A45	椶 6A46
40/59	椶 6A47	椶 6A48	椶 6A49	椶 6A4A	椶 6A4B	椶 6A4C	椶 6A4D	椶 6A4E	椶 6A4F	椶 6A50	椶 6A51	椶 6A52	椶 6A53	椶 6A54	椶 6A55	椶 6A56	椶 6A57	椶 6A58	椶 6A59
60/79	椶 6A5C	椶 6A5D	椶 6A5E	椶 6A5F	椶 6A60	椶 6A61	椶 6A62	椶 6A63	椶 6A64	椶 6A65	椶 6A66	椶 6A67	椶 6A68	椶 6A69	椶 6A6A	椶 6A6B	椶 6A6C	椶 6A6D	椶 6A6E
80/94	椶 6A70	椶 6A71	椶 6A72	椶 6A73	椶 6A74	椶 6A75	椶 6A76	椶 6A77	椶 6A78	椶 6A79	椶 6A7A	椶 6A7B	椶 6A7C	椶 6A7D	椶 6A7E				
75区 01/19	01 椶 6B21	02 椶 6B22	03 椶 6B23	04 椶 6B24	05 椶 6B25	06 椶 6B26	07 椶 6B27	08 椶 6B28	09 椶 6B29	10 椶 6B2A	11 椶 6B2B	12 椶 6B2C	13 椶 6B2D	14 椶 6B2E	15 椶 6B2F	16 椶 6B30	17 椶 6B31	18 椶 6B32	19 椶 6B33
20/39	椶 6B34	椶 6B35	椶 6B36	椶 6B37	椶 6B38	椶 6B39	椶 6B3A	椶 6B3B	椶 6B3C	椶 6B3D	椶 6B3E	椶 6B3F	椶 6B40	椶 6B41	椶 6B42	椶 6B43	椶 6B44	椶 6B45	椶 6B46
40/59	椶 6B47	椶 6B48	椶 6B49	椶 6B4A	椶 6B4B	椶 6B4C	椶 6B4D	椶 6B4E	椶 6B4F	椶 6B50	椶 6B51	椶 6B52	椶 6B53	椶 6B54	椶 6B55	椶 6B56	椶 6B57	椶 6B58	椶 6B59
60/79	椶 6B5C	椶 6B5D	椶 6B5E	椶 6B5F	椶 6B60	椶 6B61	椶 6B62	椶 6B63	椶 6B64	椶 6B65	椶 6B66	椶 6B67	椶 6B68	椶 6B69	椶 6B6A	椶 6B6B	椶 6B6C	椶 6B6D	椶 6B6E
80/94	椶 6B70	椶 6B71	椶 6B72	椶 6B73	椶 6B74	椶 6B75	椶 6B76	椶 6B77	椶 6B78	椶 6B79	椶 6B7A	椶 6B7B	椶 6B7C	椶 6B7D	椶 6B7E				
76区 01/19	01 椶 6C21	02 椶 6C22	03 椶 6C23	04 椶 6C24	05 椶 6C25	06 椶 6C26	07 椶 6C27	08 椶 6C28	09 椶 6C29	10 椶 6C2A	11 椶 6C2B	12 椶 6C2C	13 椶 6C2D	14 椶 6C2E	15 椶 6C2F	16 椶 6C30	17 椶 6C31	18 椶 6C32	19 椶 6C33
20/39	椶 6C34	椶 6C35	椶 6C36	椶 6C37	椶 6C38	椶 6C39	椶 6C3A	椶 6C3B	椶 6C3C	椶 6C3D	椶 6C3E	椶 6C3F	椶 6C40	椶 6C41	椶 6C42	椶 6C43	椶 6C44	椶 6C45	椶 6C46
40/59	椶 6C47	椶 6C48	椶 6C49	椶 6C4A	椶 6C4B	椶 6C4C	椶 6C4D	椶 6C4E	椶 6C4F	椶 6C50	椶 6C51	椶 6C52	椶 6C53	椶 6C54	椶 6C55	椶 6C56	椶 6C57	椶 6C58	椶 6C59
60/79	椶 6C5C	椶 6C5D	椶 6C5E	椶 6C5F	椶 6C60	椶 6C61	椶 6C62	椶 6C63	椶 6C64	椶 6C65	椶 6C66	椶 6C67	椶 6C68	椶 6C69	椶 6C6A	椶 6C6B	椶 6C6C	椶 6C6D	椶 6C6E
80/94	椶 6C70	椶 6C71	椶 6C72	椶 6C73	椶 6C74	椶 6C75	椶 6C76	椶 6C77	椶 6C78	椶 6C79	椶 6C7A	椶 6C7B	椶 6C7C	椶 6C7D	椶 6C7E				

T0814610-93/d122

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 20

77区 01/19		01 𡗗 6D21	02 𡗘 6D22	03 𡗙 6D23	04 𡗚 6D24	05 𡗛 6D25	06 𡗜 6D26	07 𡗝 6D27	08 𡗞 6D28	09 𡗟 6D29	10 𡗠 6D2A	11 𡗡 6D2B	12 𡗢 6D2C	13 𡗣 6D2D	14 𡗤 6D2E	15 𡗥 6D2F	16 𡗦 6D30	17 𡗧 6D31	18 𡗨 6D32	19 𡗩 6D33
20/39	𡗪 6D34	𡗫 6D35	𡗬 6D36	𡗭 6D37	𡗮 6D38	𡗯 6D39	𡗰 6D3A	𡗱 6D3B	𡗲 6D3C	𡗳 6D3D	𡗴 6D3E	𡗵 6D3F	𡗶 6D40	𡗷 6D41	𡗸 6D42	𡗹 6D43	𡗺 6D44	𡗻 6D45	𡗼 6D46	𡗽 6D47
40/59	𡗾 6D48	𡗿 6D49	𡘀 6D4A	𡘁 6D4B	𡘂 6D4C	𡘃 6D4D	𡘄 6D4E	𡘅 6D4F	𡘆 6D50	𡘇 6D51	𡘈 6D52	𡘉 6D53	𡘊 6D54	𡘋 6D55	𡘌 6D56	𡘍 6D57	𡘎 6D58	𡘏 6D59	𡘐 6D5A	𡘑 6D5B
60/79	𡘒 6D5C	𡘓 6D5D	𡘔 6D5E	𡘕 6D5F	𡘖 6D60	𡘗 6D61	𡘘 6D62	𡘙 6D63	𡘚 6D64	𡘛 6D65	𡘜 6D66	𡘝 6D67	𡘞 6D68	𡘟 6D69	𡘠 6D6A	𡘡 6D6B	𡘢 6D6C	𡘣 6D6D	𡘤 6D6E	𡘥 6D6F
80/94	𡘦 6D70	𡘧 6D71	𡘨 6D72	𡘩 6D73	𡘪 6D74	𡘫 6D75	𡘬 6D76	𡘭 6D77	𡘮 6D78	𡘯 6D79	𡘰 6D7A	𡘱 6D7B	𡘲 6D7C	𡘳 6D7D	𡘴 6D7E					
78区 01/19		01 𡗗 6E21	02 𡗘 6E22	03 𡗙 6E23	04 𡗚 6E24	05 𡗛 6E25	06 𡗜 6E26	07 𡗝 6E27	08 𡗞 6E28	09 𡗟 6E29	10 𡗠 6E2A	11 𡗡 6E2B	12 𡗢 6E2C	13 𡗣 6E2D	14 𡗤 6E2E	15 𡗥 6E2F	16 𡗦 6E30	17 𡗧 6E31	18 𡗨 6E32	19 𡗩 6E33
20/39	𡗪 6E34	𡗫 6E35	𡗬 6E36	𡗭 6E37	𡗮 6E38	𡗯 6E39	𡗰 6E3A	𡗱 6E3B	𡗲 6E3C	𡗳 6E3D	𡗴 6E3E	𡗵 6E3F	𡗶 6E40	𡗷 6E41	𡗸 6E42	𡗹 6E43	𡗺 6E44	𡗻 6E45	𡗼 6E46	𡗽 6E47
40/59	𡗾 6E48	𡗿 6E49	𡘀 6E4A	𡘁 6E4B	𡘂 6E4C	𡘃 6E4D	𡘄 6E4E	𡘅 6E4F	𡘆 6E50	𡘇 6E51	𡘈 6E52	𡘉 6E53	𡘊 6E54	𡘋 6E55	𡘌 6E56	𡘍 6E57	𡘎 6E58	𡘏 6E59	𡘐 6E5A	𡘑 6E5B
60/79	𡘒 6E5C	𡘓 6E5D	𡘔 6E5E	𡘕 6E5F	𡘖 6E60	𡘗 6E61	𡘘 6E62	𡘙 6E63	𡘚 6E64	𡘛 6E65	𡘜 6E66	𡘝 6E67	𡘞 6E68	𡘟 6E69	𡘠 6E6A	𡘡 6E6B	𡘢 6E6C	𡘣 6E6D	𡘤 6E6E	𡘥 6E6F
80/94	𡘦 6E70	𡘧 6E71	𡘨 6E72	𡘩 6E73	𡘪 6E74	𡘫 6E75	𡘬 6E76	𡘭 6E77	𡘮 6E78	𡘯 6E79	𡘰 6E7A	𡘱 6E7B	𡘲 6E7C	𡘳 6E7D	𡘴 6E7E					
79区 01/19		01 𡗗 6F21	02 𡗘 6F22	03 𡗙 6F23	04 𡗚 6F24	05 𡗛 6F25	06 𡗜 6F26	07 𡗝 6F27	08 𡗞 6F28	09 𡗟 6F29	10 𡗠 6F2A	11 𡗡 6F2B	12 𡗢 6F2C	13 𡗣 6F2D	14 𡗤 6F2E	15 𡗥 6F2F	16 𡗦 6F30	17 𡗧 6F31	18 𡗨 6F32	19 𡗩 6F33
20/39	𡗪 6F34	𡗫 6F35	𡗬 6F36	𡗭 6F37	𡗮 6F38	𡗯 6F39	𡗰 6F3A	𡗱 6F3B	𡗲 6F3C	𡗳 6F3D	𡗴 6F3E	𡗵 6F3F	𡗶 6F40	𡗷 6F41	𡗸 6F42	𡗹 6F43	𡗺 6F44	𡗻 6F45	𡗼 6F46	𡗽 6F47
40/59	𡗾 6F48	𡗿 6F49	𡘀 6F4A	𡘁 6F4B	𡘂 6F4C	𡘃 6F4D	𡘄 6F4E	𡘅 6F4F	𡘆 6F50	𡘇 6F51	𡘈 6F52	𡘉 6F53	𡘊 6F54	𡘋 6F55	𡘌 6F56	𡘍 6F57	𡘎 6F58	𡘏 6F59	𡘐 6F5A	𡘑 6F5B
60/79	𡘒 6F5C	𡘓 6F5D	𡘔 6F5E	𡘕 6F5F	𡘖 6F60	𡘗 6F61	𡘘 6F62	𡘙 6F63	𡘚 6F64	𡘛 6F65	𡘜 6F66	𡘝 6F67	𡘞 6F68	𡘟 6F69	𡘠 6F6A	𡘡 6F6B	𡘢 6F6C	𡘣 6F6D	𡘤 6F6E	𡘥 6F6F
80/94	𡘦 6F70	𡘧 6F71	𡘨 6F72	𡘩 6F73	𡘪 6F74	𡘫 6F75	𡘬 6F76	𡘭 6F77	𡘮 6F78	𡘯 6F79	𡘰 6F7A	𡘱 6F7B	𡘲 6F7C	𡘳 6F7D	𡘴 6F7E					
80区 01/19		01 𡗗 7021	02 𡗘 7022	03 𡗙 7023	04 𡗚 7024	05 𡗛 7025	06 𡗜 7026	07 𡗝 7027	08 𡗞 7028	09 𡗟 7029	10 𡗠 702A	11 𡗡 702B	12 𡗢 702C	13 𡗣 702D	14 𡗤 702E	15 𡗥 702F	16 𡗦 7030	17 𡗧 7031	18 𡗨 7032	19 𡗩 7033
20/39	𡗪 7034	𡗫 7035	𡗬 7036	𡗭 7037	𡗮 7038	𡗯 7039	𡗰 703A	𡗱 703B	𡗲 703C	𡗳 703D	𡗴 703E	𡗵 703F	𡗶 7040	𡗷 7041	𡗸 7042	𡗹 7043	𡗺 7044	𡗻 7045	𡗼 7046	𡗽 7047
40/59	𡗾 7048	𡗿 7049	𡘀 704A	𡘁 704B	𡘂 704C	𡘃 704D	𡘄 704E	𡘅 704F	𡘆 7050	𡘇 7051	𡘈 7052	𡘉 7053	𡘊 7054	𡘋 7055	𡘌 7056	𡘍 7057	𡘎 7058	𡘏 7059	𡘐 705A	𡘑 705B
60/79	𡘒 705C	𡘓 705D	𡘔 705E	𡘕 705F	𡘖 7060	𡘗 7061	𡘘 7062	𡘙 7063	𡘚 7064	𡘛 7065	𡘜 7066	𡘝 7067	𡘞 7068	𡘟 7069	𡘠 706A	𡘡 706B	𡘢 706C	𡘣 706D	𡘤 706E	𡘥 706F
80/94	𡘦 7070	𡘧 7071	𡘨 7072	𡘩 7073	𡘪 7074	𡘫 7075	𡘬 7076	𡘭 7077	𡘮 7078	𡘯 7079	𡘰 707A	𡘱 707B	𡘲 707C	𡘳 707D	𡘴 707E					

T0814620-93/d123

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 21

81区	翊	痼	瘰	癰	瘡	癩	癬	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰
01-19	7134	7121	7122	7123	7124	7125	7126	7127	7128	7129	712A	712B	712C	712D	712E	712F	7130	7131	7132	7133
20-39	翊	痼	瘰	癰	瘡	癩	癬	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰
40-59	翊	痼	瘰	癰	瘡	癩	癬	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰
60-79	翊	痼	瘰	癰	瘡	癩	癬	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰
80-94	翊	痼	瘰	癰	瘡	癩	癬	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰	癰
82区	虵	頤	頤	頤	頤	頤	頤	頤	頤	頤	頤	頤	頤	虵	虵	虵	虵	虵	虵	虵
01-19	7234	7221	7222	7223	7224	7225	7226	7227	7228	7229	722A	722B	722C	722D	722E	722F	7230	7231	7232	7233
20-39	虵	頤	頤	頤	頤	頤	頤	頤	頤	頤	頤	頤	頤	虵	虵	虵	虵	虵	虵	虵
40-59	虵	頤	頤	頤	頤	頤	頤	頤	頤	頤	頤	頤	頤	虵	虵	虵	虵	虵	虵	虵
60-79	虵	頤	頤	頤	頤	頤	頤	頤	頤	頤	頤	頤	頤	虵	虵	虵	虵	虵	虵	虵
80-94	虵	頤	頤	頤	頤	頤	頤	頤	頤	頤	頤	頤	頤	虵	虵	虵	虵	虵	虵	虵
83区	蟠	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪
01-19	7334	7321	7322	7323	7324	7325	7326	7327	7328	7329	732A	732B	732C	732D	732E	732F	7330	7331	7332	7333
20-39	蟠	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪
40-59	蟠	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪
60-79	蟠	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪
80-94	蟠	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪	蟪
84区	舩	簞	簞	簞	簞	簞	簞	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩
01-19	7434	7421	7422	7423	7424	7425	7426	7427	7428	7429	742A	742B	742C	742D	742E	742F	7430	7431	7432	7433
20-39	舩	簞	簞	簞	簞	簞	簞	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩
40-59	舩	簞	簞	簞	簞	簞	簞	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩
60-79	舩	簞	簞	簞	簞	簞	簞	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩
80-94	舩	簞	簞	簞	簞	簞	簞	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩

T0814630-93/d124

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 22

85 区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
20/39	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
40/59	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
60/79	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
80/94	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
86 区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
20/39	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
40/59	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
60/79	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
80/94	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
87 区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
20/39	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
40/59	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
60/79	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
80/94	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
88 区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01/19	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
20/39	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
40/59	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
60/79	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜
80/94	醅	醑	醒	醔	醖	醗	醘	醙	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜	醜

T0814640-93/d125

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 23

89区 01-19	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	
20-39		7921	7922	7923	7924	7925	7926	7927	7928	7929	792A	792B	792C	792D	792E	792F	7930	7931	7932	7933
40-59	7934	7935	7936	7937	7938	7939	793A	793B	793C	793D	793E	793F	7940	7941	7942	7943	7944	7945	7946	7947
60-79	7948	7949	794A	794B	794C	794D	794E	794F	7950	7951	7952	7953	7954	7955	7956	7957	7958	7959	795A	795B
80-94	795C	795D	795E	795F	7960	7961	7962	7963	7964	7965	7966	7967	7968	7969	796A	796B	796C	796D	796E	796F
	7970	7971	7972	7973	7974	7975	7976	7977	7978	7979	797A	797B	797C	797D	797E					
90区 01-19	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	
20-39	倭 7A34	倭 7A35	倭 7A36	倭 7A37	倭 7A38	倭 7A39	倭 7A3A	倭 7A3B	倭 7A3C	倭 7A3D	倭 7A3E	倭 7A3F	倭 7A40	倭 7A41	倭 7A42	倭 7A43	倭 7A44	倭 7A45	倭 7A46	倭 7A47
40-59	倭 7A48	倭 7A49	倭 7A4A	倭 7A4B	倭 7A4C	倭 7A4D	倭 7A4E	倭 7A4F	倭 7A50	倭 7A51	倭 7A52	倭 7A53	倭 7A54	倭 7A55	倭 7A56	倭 7A57	倭 7A58	倭 7A59	倭 7A5A	倭 7A5B
60-79	倭 7A5C	倭 7A5D	倭 7A5E	倭 7A5F	倭 7A60	倭 7A61	倭 7A62	倭 7A63	倭 7A64	倭 7A65	倭 7A66	倭 7A67	倭 7A68	倭 7A69	倭 7A6A	倭 7A6B	倭 7A6C	倭 7A6D	倭 7A6E	倭 7A6F
80-94	倭 7A70	倭 7A71	倭 7A72	倭 7A73	倭 7A74	倭 7A75	倭 7A76	倭 7A77	倭 7A78	倭 7A79	倭 7A7A	倭 7A7B	倭 7A7C	倭 7A7D	倭 7A7E					
91区 01-19	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	
20-39	屏 7B34	屏 7B35	屏 7B36	屏 7B37	屏 7B38	屏 7B39	屏 7B3A	屏 7B3B	屏 7B3C	屏 7B3D	屏 7B3E	屏 7B3F	屏 7B40	屏 7B41	屏 7B42	屏 7B43	屏 7B44	屏 7B45	屏 7B46	屏 7B47
40-59	珣 7B48	珣 7B49	珣 7B4A	珣 7B4B	珣 7B4C	珣 7B4D	珣 7B4E	珣 7B4F	珣 7B50	珣 7B51	珣 7B52	珣 7B53	珣 7B54	珣 7B55	珣 7B56	珣 7B57	珣 7B58	珣 7B59	珣 7B5A	珣 7B5B
60-79	珣 7B5C	珣 7B5D	珣 7B5E	珣 7B5F	珣 7B60	珣 7B61	珣 7B62	珣 7B63	珣 7B64	珣 7B65	珣 7B66	珣 7B67	珣 7B68	珣 7B69	珣 7B6A	珣 7B6B	珣 7B6C	珣 7B6D	珣 7B6E	珣 7B6F
80-94	珣 7B70	珣 7B71	珣 7B72	珣 7B73	珣 7B74	珣 7B75	珣 7B76	珣 7B77	珣 7B78	珣 7B79	珣 7B7A	珣 7B7B	珣 7B7C	珣 7B7D	珣 7B7E					
92区 01-19	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	
20-39	赅 7C34	赅 7C35	赅 7C36	赅 7C37	赅 7C38	赅 7C39	赅 7C3A	赅 7C3B	赅 7C3C	赅 7C3D	赅 7C3E	赅 7C3F	赅 7C40	赅 7C41	赅 7C42	赅 7C43	赅 7C44	赅 7C45	赅 7C46	赅 7C47
40-59	赅 7C48	赅 7C49	赅 7C4A	赅 7C4B	赅 7C4C	赅 7C4D	赅 7C4E	赅 7C4F	赅 7C50	赅 7C51	赅 7C52	赅 7C53	赅 7C54	赅 7C55	赅 7C56	赅 7C57	赅 7C58	赅 7C59	赅 7C5A	赅 7C5B
60-79	赅 7C5C	赅 7C5D	赅 7C5E	赅 7C5F	赅 7C60	赅 7C61	赅 7C62	赅 7C63	赅 7C64	赅 7C65	赅 7C66	赅 7C67	赅 7C68	赅 7C69	赅 7C6A	赅 7C6B	赅 7C6C	赅 7C6D	赅 7C6E	赅 7C6F
80-94	赅 7C70	赅 7C71	赅 7C72	赅 7C73	赅 7C74	赅 7C75	赅 7C76	赅 7C77	赅 7C78	赅 7C79	赅 7C7A	赅 7C7B	赅 7C7C	赅 7C7D	赅 7C7E					

T0814650-93/d126

Table C.16/T.101 – The Chinese primary set of graphic characters

Part 24

93 区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01-19	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑
20-39	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑
40-59	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑
60-79	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑
80-94	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑
94 区	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01-19	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑
20-39	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑
40-59	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑
60-79	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑
80-94	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑	涑

T0814660-93/d127

C.3.4.2 The coding of characters with diacritical marks

This applies only to Latin or Greek characters.

Each of these characters is represented by a sequence of two bit-combinations. The first part of this sequence consists of a bit-combination in the range 4/0 to 4/15 from the supplementary set representing a diacritical mark. The second part consists of a bit-combination in the range 4/1 to 5/10 or 6/1 to 7/10 from the primary set representing a basic Latin letter or a bit combination in the range 4/1 to 5/9 or 6/1 to 7/9 from the primary set representing a basic Greek letter or space. The diacritical marks for both Latin and Greek are shown in column 4 of Table C.9 and the basic Latin letters are shown in Table C.8. The Greek basic letters are shown in Table C.13.

NOTE – If a diacritical mark is used in combination with a basic character such that the resulting character is not within the repertoire the terminal will display the basic character.

C.3.4.3 Designation of graphic sets

ESC	2/8	4/0	Latin Primary set of graphic characters (G0)	to G0
ESC	2/9	4/0	Latin Primary set of graphic characters (G0)	to G1
ESC	2/10	4/0	Latin Primary set of graphic characters (G0)	to G2
ESC	2/11	4/0	Latin Primary set of graphic characters (G0)	to G3
ESC	2/8	6/3	Secondary supplementary set of mosaic characters (G1)	to G0
ESC	2/9	6/3	Secondary supplementary set of mosaic characters (G1)	to G1
ESC	2/10	6/3	Secondary supplementary set of mosaic characters (G1)	to G2
ESC	2/11	6/3	Secondary supplementary set of mosaic characters (G1)	to G3
ESC	2/8	6/2	Supplementary set of graphic characters (G2)	to G0
ESC	2/9	6/2	Supplementary set of graphic characters (G2)	to G1
ESC	2/10	6/2	Supplementary set of graphic characters (G2)	to G2
ESC	2/11	6/2	Supplementary set of graphic characters (G2)	to G3

ESC	2/8	6/4	Third supplementary set of mosaic characters (G3)	to G0
ESC	2/9	6/4	Third supplementary set of mosaic characters (G3)	to G1
ESC	2/10	6/4	Third supplementary set of mosaic characters (G3)	to G2
ESC	2/11	6/4	Third supplementary set of mosaic characters (G3)	to G3

NOTE – The default position is shown in brackets.

C.3.4.4 Designation of the greek primary set

ESC	2/8	2/1	4/0	Greek Primary set of graphic characters	to G0
ESC	2/9	2/1	4/0	Greek Primary set of graphic characters	to G1
ESC	2/10	2/1	4/0	Greek Primary set of graphic characters	to G2
ESC	2/11	2/1	4/0	Greek Primary set of graphic characters	to G3

C.3.4.5 Designation of the arabic primary set

ESC	2/8	6/11	Arabic Primary set of graphic characters	to G0
ESC	2/9	6/11	Arabic Primary set of graphic characters	to G1
ESC	2/10	6/11	Arabic Primary set of graphic characters	to G2
ESC	2/11	6/11	Arabic Primary set of graphic characters	to G3

C.3.4.6 Designation of the hebrew primary set

ESC	2/13	4/8	Hebrew Supplementary set of graphic characters	to G1
ESC	2/14	4/8	Hebrew Supplementary set of graphic characters	to G2
ESC	2/15	4/8	Hebrew Supplementary set of graphic characters	to G3

C.3.4.7 Designation of the chinese Primary Set

C.3.4.7.1 Designation of the chinese coded ideogram set for communication

ESC	2/4	2/8	4/5	Chinese coded ideogram set for communication	to G0
ESC	2/4	2/9	4/5	Chinese coded ideogram set for communication	to G1
ESC	2/4	2/10	4/5	Chinese coded ideogram set for communication	to G2
ESC	2/4	2/11	4/5	Chinese coded ideogram set for communication	to G3

C.3.4.7.2 Designation of the chinese graphic DRCS

ESC	2/8	2/0	4/1	Chinese graphic DRCS	to G0
ESC	2/9	2/0	4/1	Chinese graphic DRCS	to G1
ESC	2/10	2/0	4/1	Chinese graphic DRCS	to G2
ESC	2/11	2/0	4/1	Chinese graphic DRCS	to G3

C.3.4.7.3 Designation of the chinese character DRCS

ESC	2/8	2/0	4/0	Chinese character DRCS	to G0
ESC	2/9	2/0	4/0	Chinese character DRCS	to G1
ESC	2/10	2/0	4/0	Chinese character DRCS	to G2
ESC	2/11	2/0	4/0	Chinese character DRCS	to G3

C.3.5 Supplementary attribute and qualified area controls

C.3.5.1 Serial control STOP CONCEAL

<i>Abbreviation</i>	<i>Name and Coding</i>
STC	STOP CONCEAL CSI 4/2

C.3.5.2 Full screen and full row attributes

The attributes:

FOREGROUND COLOUR
BACKGROUND COLOUR
LINED

SIZE
FLASH
CONCEAL
INVERT
WINDOW/BOX

are coded as four-character escape sequences of the form:

ESC 2/3 2/0 (Fe) for full screen attributes;

ESC 2/3 2/1 (Fe) for full row attributes;

where Fe is the attribute control character from the parallel C1 set in the 7-bit environment

ie Fe is 4/1 for Red foreground

C.3.5.3 Marking controls

<i>Abbreviation</i>	<i>Name and Coding</i>	
MMS	MARKED MODE START	
	Full screen control	CSI 3/0 5/3
	Full row control	CSI 3/1 5/3
	Serial or parallel control	CSI 3/2 5/3
MMT	MARKED MODE STOP	
	Full screen control	CSI 3/0 5/4
	Full row control	CSI 3/1 5/4
	Serial or parallel control	CSI 3/2 5/4

C.3.5.4 Protecting controls

<i>Abbreviation</i>	<i>Name and Coding</i>	
PMS	PROTECTED MODE START	
	Full screen control	CSI 3/0 5/0
	Full row control	CSI 3/1 5/0
	Serial or parallel control	CSI 3/2 5/0
PMC	PROTECTED MODE CANCEL	
	Full screen control	CSI 3/0 5/1
	Full row control	CSI 3/1 5/1
	Serial or parallel control	CSI 3/2 5/1
PMI	PROTECTED MODE IDLE	
	Serial or parallel control	CSI 3/2 5/2

The currently invoked C1 set indicates whether the above controls for MARKED and PROTECTED should be interpreted as serial or parallel controls.

C.3.5.5 Definition of a scrolling area

Similar CSI sequences are used for CREATE SCROLLING AREA and DELETE SCROLLING AREA; only the final characters are different.

CSI <URH> <URT> <URU> 3/11 <LRH> <LRT> <LRU> <F>

URH Hundreds value of the upper row
URT Tens value of the upper row
URU Units value of the upper row
LRH Hundreds value of the lower row
LRT Tens value of the lower row
LRU Units value of the lower row

These values are coded from column 3 of the code table. Leading zeros may be omitted.

F: 5/5 for CREATE SCROLLING AREA
5/6 for DELETE SCROLLING AREA

The action of scrolling is initiated as described in C.2.3.12 and C.3.6 of Part 1.

C.3.5.6 Colour table controls

The coding of the colour table invocation controls is as follows:

<i>Abbreviation</i>	<i>Name and Coding</i>	
CT1	COLOUR TABLE 1	CSI 3/0 4/0
CT2	COLOUR TABLE 2	CSI 3/1 4/0
CT3	COLOUR TABLE 3	CSI 3/2 4/0
CT4	COLOUR TABLE 4	CSI 3/3 4/0

C.3.5.7 Additional FLASH Controls

The additional FLASH controls are coded as follows:

<i>Abbreviation</i>	<i>Name and Coding</i>	
IVF	INVERTED FLASH	CSI 3/0 4/1
RIF	REDUCED INTENSITY FLASH	CSI 3/1 4/1
FF1	FAST FLASH 1	CSI 3/2 4/1
FF2	FAST FLASH 2	CSI 3/3 4/1
FF3	FAST FLASH 3	CSI 3/4 4/1
ICF	INCREMENT FLASH	CSI 3/5 4/1
DCF	DECREMENT FLASH	CSI 3/6 4/1

C.3.6 Device controls

C.3.6.1 Cursor controls

See primary control function set, Part 1, C.3.2.

C.3.6.2 Supplementary device controls

<i>Abbreviation</i>	<i>Name and Coding</i>	
RDW	RECORDING DEVICE WAIT	ESC 3/5
RDS	RECORDING DEVICE START	ESC 3/6
RDT	RECORDING DEVICE STOP	ESC 3/7
HCW	HARD COPY WAIT	ESC 3/8
HCS	HARD COPY START	ESC 3/9
HCT	HARD COPY STOP	ESC 3/10
DDO	DISPLAY DEVICE ON	ESC 3/12
DDF	DISPLAY DEVICE OFF	ESC 3/13
ADO	AUXILIARY DEVICE ON	ESC 3/14
ADF	AUXILIARY DEVICE OFF	ESC 3/15
SCU	SCROLL UP	CSI 3/0 6/0
SCD	SCROLL DOWN	CSI 3/1 6/0
AIS	ACTIVATE IMPLICIT SCROLLING	CSI 3/2 6/0
DIS	DEACTIVATE IMPLICIT SCROLLING	CSI 3/3 6/0
EBU	EMPTY BUFFER	ESC 3/11

C.3.7 Designation and invocation in the 7-bit environment (see Figure C.3)

C.3.7.1 General

For the 7-bit environment the bases of the coding structure for the Videotex service are the CCITT recommendation V3 (ISO 646), and International Standards ISO 2022 (Rev 79) and ISO 6937.

C.3.7.2 Coding of code extension control functions

<i>Abbreviation</i>	<i>Name and Coding</i>	
SI	SHIFT IN	0/15
SO	SHIFT OUT	0/14
LS2	LOCKING SHIFT 2	ESC 6/14
LS3	LOCKING SHIFT 3	ESC 6/15
SS2	SINGLE SHIFT 2	1/9
SS3	SINGLE SHIFT 3	1/13

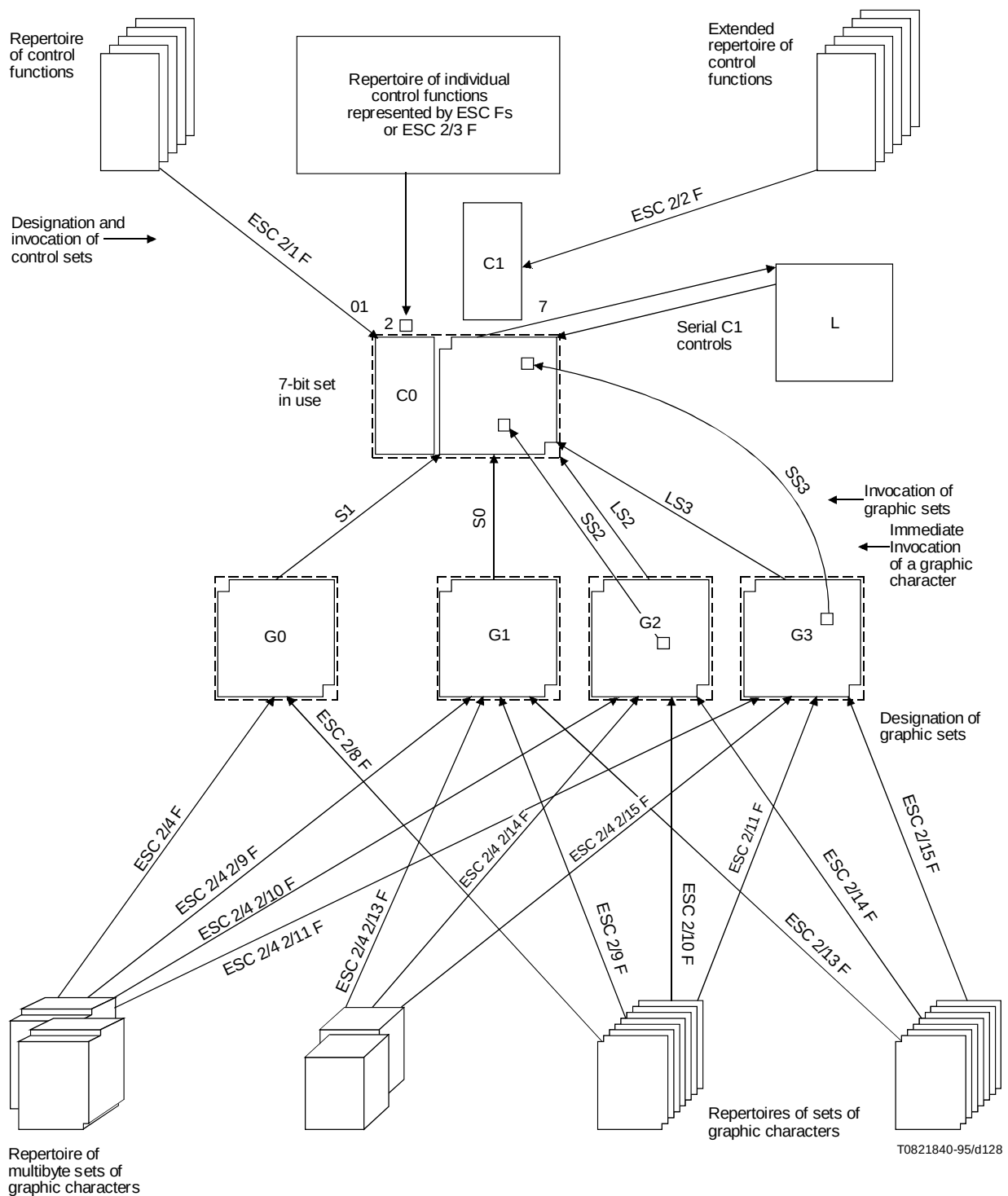
C.3.8 Designation and invocation in the 8-bit environment (see Figure C.4)

C.3.8.1 General

The 8-bit code environment preserves the code extension structure of ISO 2022, i.e the G0 set is invoked into the left-hand part (positions 2/1 to 7/14) and the G2 set into the right-hand part (positions 10/1 to 7/14) of the code table.

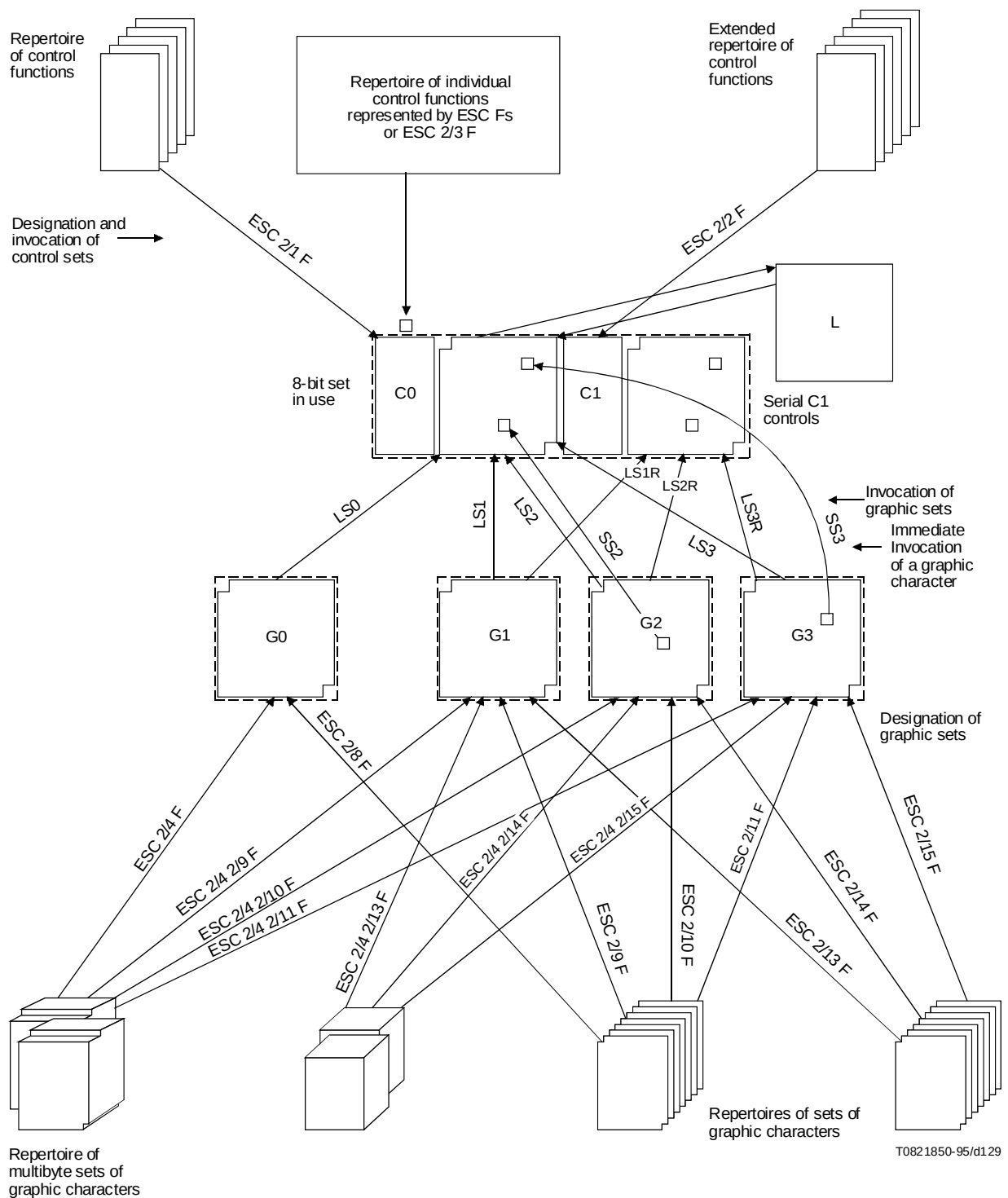
C.3.8.2 Coding of code extension control functions

<i>Abbreviation</i>	<i>Name and Coding</i>	
LS0	LOCKING SHIFT 0	0/15
LS1	LOCKING SHIFT 1	0/14
LS1R	LOCKING SHIFT 1 RIGHT	ESC 7/14
LS2	LOCKING SHIFT 2	ESC 6/14
LS2R	LOCKING SHIFT 2 RIGHT	ESC 7/13
LS3	LOCKING SHIFT 3	ESC 6/15
LS3R	LOCKING SHIFT 3 RIGHT	ESC 7/12
SS2	SINGLE SHIFT	1/9
SS3	SINGLE SHIFT 3	1/13



NOTE – See C.3.1.2.

Figure C.3/T.101 – Code extension in a 7-bit environment



NOTE – See C.3.1.3.

Figure C.4/T.101 – Code extension in an 8-bit environment

Appendix I

(to Part 1 of Annex C to Recommendation T.101)

Identification system

- 1) For the purpose of this Recommendation, a system has been developed that allows for the identification and description of each graphic character or control function. The system is shown in Figure C.5.
- 2) Each identifier consists of two letters and two digits.
- 3) The first letter indicates the alphabet, the language, etc.
- 4) The second letter indicates the letter of an alphabet or, in the case of a non-alphabetic graphic character or a control function, the group of characters or control functions.
- 5) The first digit indicates whether the letter in the second position is modified with a diacritical mark, the position of the diacritical mark, etc. It has no special meaning in the case of the first letter being a C, N or S.
- 6) The second digit indicates whether the letter is a capital or a small one (even or odd respectively). If the first letter is a C, N or S, this digit being even or odd has no significance.
- 7) The numbering is used in a consistent manner so that each diacritical mark is always given the same number.
- 8) The numbering principle is shown in Table C.17.

Table C.17/T.101 – Numbering principle for alphabetic characters

Item	Small	Capital
No diacritical mark	01	02
Acute accent	11	12
Grave accent	13	14
Circumflex	15	16
Diaeresis or umlaut	17	18
Tilde	19	20
Caron	21	22
Breve	23	24
Double acute accent	25	26
Ring	27	28
Dot	29	30
Macron	31	32
Diaeresis with acute acent	33	–
Cedilla	41	42
Ogonek	43	44
Diphthong or ligature	51	52
Special form	61, 63, etc.	62, 64, etc.

L	A	0	1
			_ For alphabetic characters:
			odd digit = small letter;
			even digit = capital letter.
			_ If C, N or S in first position:
			no special meaning.
			_ For alphabetic characters:
			0 = letter without diacritical mark;
			1, 2 or 3 = letter with diacritical mark above it;
			4 = letter with diacritical mark below it;
			5 = diphthong or ligature;
			6 = special form.
			_ If C, N or S in first position:
			no special meaning.
			_ For alphabetic characters:
			A to Z = the respective letter of the Latin alphabet, or the
			Latin equivalent in the case of a non-Latin letter.
			_ If C in first position:
			E = code extension control function;
			F = format effector;
			P = presentation control function;
			M = other control function.
			_ If N in first position:
			D = decimal digit;
			F = fraction;
			S = subscript or superscript
			_ If S in first position:
			A = arithmetic sign;
			C = currency sign;
			D = diacritical mark;
			P = punctuation mark;
			M = other symbol.
			_ L = Latin alphabetic character;
			_ C = control function;
			_ N = non-alphabetic graphic character;
			_ S = special graphic character;
			_ A = Arabic alphabetic character
			_ G = Greek alphabetic character.
			_ H = Hebrew alphabetic character

FIGURE C.5/T.101 – Identification system

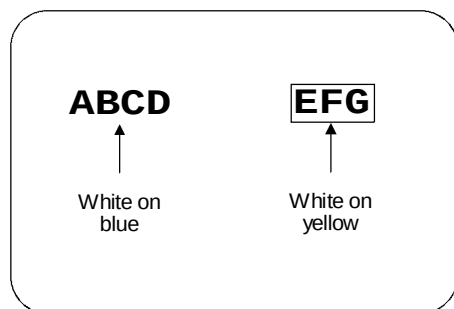
Appendix II
(to Part 1 of Annex C to Recommendation T.101)

Example of time dependency in the unified alphamosaic model

EXAMPLE 1

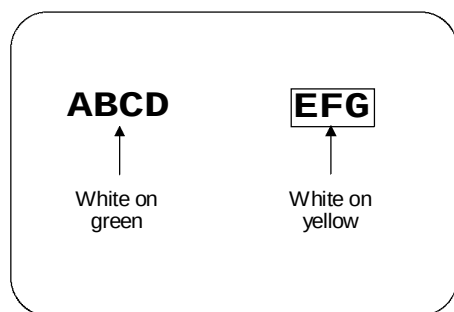
Full screen blue

Codes:



CS, full screen blue background,
(transparent background), A, B, C,
D, yellow background E, F, G

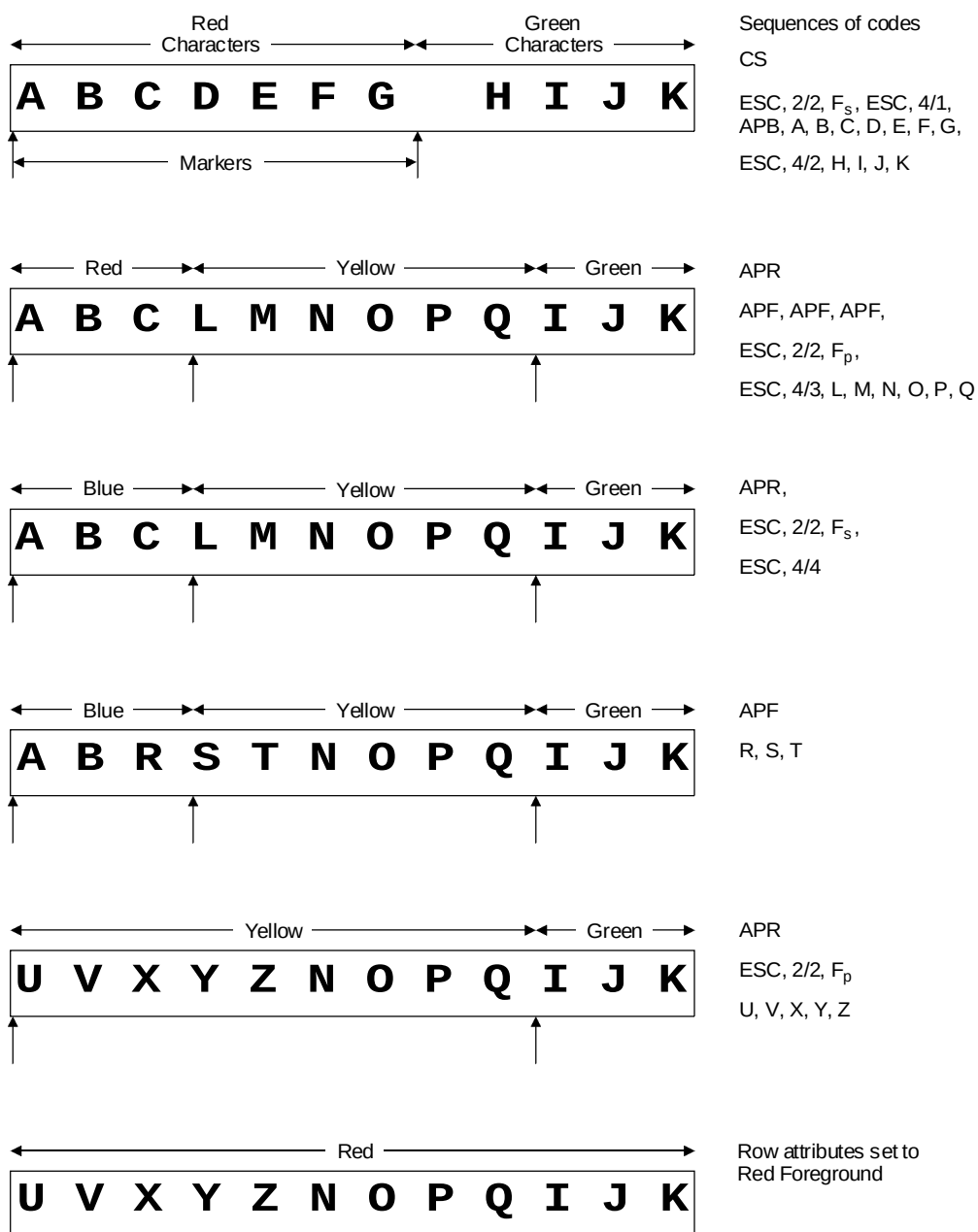
Full screen green



Full screen green background

T0821860-95/d130

EXAMPLE 2



T0821870-95/d1.31

Part 2 – Geometric displays

CONTENTS

	<i>Page</i>
C.1 Introduction	290
C.2 Scope and field of application	290
C.3 References.....	290
C.4 Definitions and Abbreviations	291
C.4.1 Definitions.....	292
C.4.2 Abbreviations.....	295
C.5 General description.....	296
C.5.1 Graphical output.....	296
C.5.2 Workstations	312
C.5.3 Coordinate systems and transformations	314
C.5.4 Segments	318
C.5.5 Deferring picture changes	321
C.5.6 Graphical input.....	323
C.5.7 Inquiry	324
C.5.8 Error detection	324
C.5.9 Error handling	324
C.5.10 Levels	324
C.6 Description of the primitives	328
C.6.1 Introduction.....	328
C.6.2 Geometric primitives.....	328
C.7 Encoding principles	361
C.7.1 Encoding principles of the opcode.....	363
C.7.2 Encoding principles of the operands	364
C.8 Encoding of the primitives.....	383
C.8.1 Primitive encoding	383
C.8.2 Coding of the primitives	384
C.9 Defaults.....	397
C.10 Conformance.....	398
Appendix I – Primitives, Workstation Categories, Levels and Options	399
Appendix II – B-spline curves and ellipses	401
II.1 Short note on B-spline curves.....	401
II.2 Ellipse primitives	403
Appendix III – Cross references	405
III.1 Functional order	405
III.2 Alphabetic order	406
Appendix IV – Selection of the geometric display.....	408

C.1 Introduction

ECMA-96 specifies a Graphics Data Syntax for a multiple workstation interface (GDS).

It is based on ISO 7942 Information Processing – Graphical Kernel System (GKS) – Functional description, therefore taking advantage of the work already done in the international computer graphics community.

GDS' functionalities are based on the concept of a workstation as defined in GKS. Although the full GKS workstation concept can only be realized by GKS itself, this standard provides the capability to communicate groups of GKS functions to the graphics configuration. Following the GKS definition this allows advantage to be taken of the different capabilities of the various devices of which the graphics configuration is comprised.

In order to have one syntax for the functions used in the computer graphics community, the encoding structure of GDS is based on the encoding structure as defined in this Recommendation (Data Syntax II).

This part (Part 2 of T/CD 6.1, Geometric display) is a proper subset of GDS. This subset is defined in terms of GDS. All references to inquire functions and input functions are not applicable to the Videotex geometric display and should be ignored.

NOTE – Wherever GDS is mentioned in this annex it also reflects to this Part 2 of T/CD 6.1.

C.2 Scope and field of application

This annex specifies the set of functions to be used in a graphics configuration and their encoding in a 7-bit or 8-bit environment. In addition, the code tables are structured in accordance with ISO 646.

The intention of this annex is to facilitate data interchange, not to standardize equipment. The specification of the concepts is included only to delimit the field of application. The definitions of the primitives may not be applicable to a graphics configuration, which does not conform to the specified concepts.

The graphics primitives contained in this annex are derived from GKS. The set of primitives necessary in a graphics configuration depends on the required GKS level.

Figure C.1 shows the model describing the GKS environment and its interfaces. The Graphics application in the field of Computer Aided Engineering (CAE), Computer Aided Design (CAD), Business Graphics, Telematic Services, etc., can be written in high level languages for which specific bindings with GKS are in the process of standardization.

A graphics application program, using GKS functions, communicates with the graphics configuration through the multiple workstation interface. Above the multiple workstation interface are the GKS normalization transformations, which convert world coordinates to normalized device coordinates. Below the multiple workstation interface are the GKS workstations, which are connected to and driven by GKS. The workstations are mapped to the graphics configuration and the normalized device coordinates are transformed to device coordinates by the workstation transformation. Non-existent capabilities of the graphics configuration must be simulated by using emulation software.

The data syntax and encoding for the GKS multiple workstation interface is provided in this annex.

C.3 References

- ISO 7942, *Information processing – Graphical Kernel System (GKS) functional description*.
- ISO 646, *Information processing – ISO 7-bit coded character set for information interchange*.
- ISO 2022, *Information processing – ISO 7-bit and 8-bit coded character sets – Coded extension techniques*.

- ISO DP 8632, *Information processing – Computer graphics – Metafile for the storage and transfer of picture description information*.
- CCITT Rec. T.101, *International interworking for Videotex services*.
- ECMA-96, *Syntax of graphical data for a multiple-workstation interface (GDS)*.

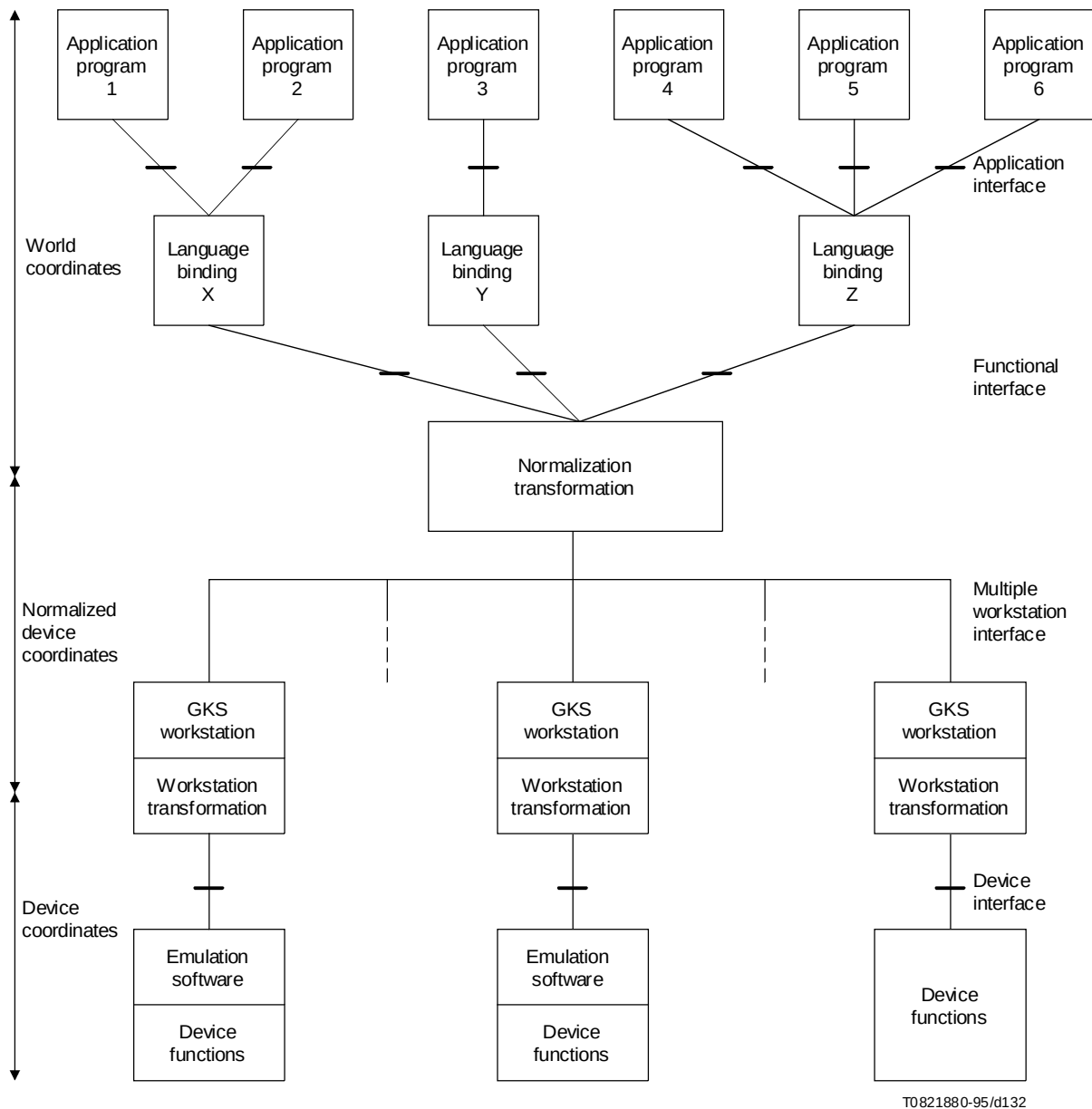


Figure C.1/T.101 – Model describing the GKS environment and its interfaces

C.4 Definitions and Abbreviations

This clause contains the definition and abbreviations of terms used in this annex.

C.4.1 Definitions

aspect ratio: The ratio of the width to the height of a rectangular area, such as workstation window or workstation viewport.

Example: an aspect ratio of 2:1 indicates an area twice as wide as it is high.

aspect source flag (ASF): An indicator (flag) controlling whether the value of the associated attribute is obtained from a bundle table (BUNDLED) or from an individual specification (INDIVIDUAL).

attribute: A particular property that applies to a display element (output primitive) or a segment.

Examples: highlighting, character height.

basic grid unit (BGU): A binary fraction that identifies the accuracy of coordinates.

bundle: A set of attributes associated with one of the output primitives.

bundle index: An index into a bundle table for a particular output primitive. It defines the workstation dependent aspects of the primitive.

bundle table: A workstation dependent table associated with a particular output primitive. Entries in the table specify all the workstation-dependent aspects of an output primitive. Bundle tables exist for the following output primitives: polyline, polymarker, text and fill area.

cell array: A display element consisting of a parallelogram subdivided into parallelograms of equal size, each having a single colour. These cells do not necessarily map one-to-one to pixels.

choice device: A logical input device providing a non-negative integer defining one of a set of alternatives.

clipping: Removing parts of display elements that lie outside a given boundary, usually a workstation window, workstation viewport or clipping rectangle.

clipping rectangle: A rectangle defined in NDC space used as a clipping boundary when the display elements have to be clipped.

colour table: A workstation dependent table, in which the entries specify the values of the red, green and blue intensities of a particular colour.

detectability: A segment attribute which makes the selection of a segment possible or not by a pick input device.

device coordinate (DC): A coordinate expressed in a coordinate system that is device dependent.

device space: The space defined by the addressable points of a display device.

differential chain code (DCC): A coding method used in Incremental code, identifying differences between steps (increments).

display device: A device on which display images can be represented.

display element: A basic graphic element that can be used to construct a display image.

display element attribute: Display element attribute values (for output display elements) are selected by the application in a workstation independent manner, but can have workstation dependent effects.

display image; picture: A collection of display elements that are represented together on a display surface.

display space: The portion of the device space that corresponds to the area available for displaying images.

display surface; view surface: The medium in a display device on which the display images may appear. (For example: the screen of a display, the paper in a plotter.)

domain ring: A mechanism for defining the ring used in encoding incremental mode coordinate data.

echo: The immediate notification of the current value provided by an input device to the operator at the display surface.

fill area: A display element consisting of a polygon (closed boundary) which may be hollow or may be filled with a uniform colour, a pattern or a hatch style.

fill area bundle table: A table associating specific values of FILL AREA attributes with a fill area bundle index. This table contains entries consisting of fill area colour, fill area interior style and fill area style index.

font: A family or assortment of characters of a given size and style.

GDS escape: A mechanism used to access implementation or device dependent features, other than those for the generation of graphical output, which are otherwise not addressable by any primitive.

generalized drawing primitive (GDP): A display element (graphics primitive) used to address special geometric workstation capabilities such as curve drawing.

GKS instance: A combination of GKS and one or more graphics configurations.

graphical kernel system (GKS): The application programmer's interface to graphics defined in ISO 7942.

graphics device: An output device (for example: refresh display, storage tube display or plotter) on which display images can be represented.

highlighting: A device dependent way of emphasizing a segment by modifying its visual attributes (a generalization of blinking).

input class: A set of input devices that are logically equivalent with respect to their function. The input classes are: LOCATOR, STROKE, VALUATOR, CHOICE, PICK and STRING.

locator device: A logical input device providing a position in normalized device coordinates.

logical input device: A logical input device is an abstraction of one or more physical devices delivering a logical input value. Logical input devices can be of class LOCATOR, STROKE, VALUATOR, CHOICE, PICK and STRING.

logical input value: A value delivered by a logical input device.

marker: A glyph with a specified appearance which is used to identify a particular location.

measure: A value (associated with a logical input device), which is determined by one or more physical input devices and a mapping from the values delivered by the physical devices. The logical input value delivered by a logical input device is the current value of the measure.

normalized device coordinate (NDC): A coordinate specified in a device independent intermediate coordinate system, normalized to some range.

operator: A person manipulating physical input devices in order to validate the measures of logical input devices.

output primitive: A display element (primitive) that actually generates (parts of) a display image. These are: POLYLINE, FILL AREA, POLYMARKER, CELL ARRAY, TEXT and GDPS.

pick device: A logical input device providing the pick identifier attached to an output primitive and the associated segment name.

pick identifier: A name, attached to individual output primitives within a segment, and returned by the pick device. The same pick identifier can be assigned to different output primitives.

picture: (See display image.)

picture element: (See pixel.)

pixel; picture element: The smallest element of a display surface that can be independently assigned a colour or intensity.

polyline: A display element consisting of a set of connected lines.

polyline bundle table: A table associating specific values for all workstation dependent aspects of a polyline display element with a polyline bundle index. This table contains entries consisting of line type, line width scale factor and colour index.

polymarker: A display element consisting of a set of locations, each to be indicated by a marker.

polymarker bundle table: A table associating specific values for all workstation dependent aspects of a polymarker display element with a polymarker bundle index. This table contains entries consisting of marker type, marker size scale factor and colour index.

primitive: A basic graphic element that can be used to construct a display image.

primitive attribute: Primitive attribute values (for output primitives) are selected in a workstation independent manner, but can have workstation dependent effects.

prompt: Output to the operator indicating that a specific logical input device is available.

ring: A square defined by its radius and angular resolution factor, used for encoding increments in the Incremental mode.

rotation: Turning all or part of a display image about an axis. Rotation is restricted to segments.

scaling: Enlarging or reducing all or part of a display image by multiplying the coordinates of the display elements by a constant value. Scaling is restricted to segments.

segment: A collection of output primitives that can be manipulated as a unit.

segment attributes: Attributes that apply only to segments. They are: visibility, highlighting, detectability, segment priority and segment transformation.

segment priority: A segment attribute used to determine which of several overlapping segments take precedence for graphics output and input.

segment transformation: A transformation which causes the display elements defined by a segment to appear with varying position (translation), size (scaling), and or orientation (rotation) on the display surface.

string device: A logical input device providing a character string.

stroke device: A logical input device providing a sequence of points in normalized device coordinates.

text: A display element consisting of a character string.

text bundle table: A table associating specific values for all workstation dependent aspects of a text display element with a text bundle index. This table contains entries consisting of text font and precision, character expansion factor, character spacing and colour index.

text font and precision: An aspect of text having two components, font and precision, which together determine the shape of the characters being output on a particular workstation. In addition, the precision describes the fidelity with which the other text aspects match those requested by an application program. In order of increasing fidelity, the precisions are: STRING, CHARACTER and STROKE.

translation: The application of a constant displacement to the position of all or part of a display image. Translation is restricted to segments.

trigger: A physical device or set of devices which an operator can use to indicate significant moments in time.

valuator device: A logical input device providing a real number.

view surface: (See display surface.)

visibility: A segment attribute which determines whether a segment is displayed or not.

workstation: The concept of an abstract graphics device which provides the logical interface through which the physical devices are controlled.

workstation transformation: A transformation that maps the boundary and interior of a workstation window to the boundary and interior of a workstation viewpoint, preserving the aspect ratio. It maps positions in Normalized Device Coordinates to Device Coordinates.

workstation viewpoint: A portion of device coordinate space currently selected for both input and output operations.

workstation window: A rectangular region within the normalized device coordinate system which is represented on a display space.

C.4.2 Abbreviations

ASAP	As Soon As Possible
ASF	Aspect Source Flag
ASTI	At Some TIme
BGU	Basic Grid Unit
BNIG	Before the Next Interaction Globally
BNIL	Before the Next Interaction Locally
DC	Device Coordinates
DCC	Differential Chain Code
GDS	Graphics Data Syntax for a multiple workstation interface
GDP	Generalized Drawing Primitive
GKS	Graphical Kernal System
IMM	IMMediately
INPUT	INPUT only workstation
IRG	Implicit ReGeneration
NDC	Normalized Device Coordinates
OUTIN	OUTput and INput workstation
OUTPUT	OUTPUT only workstation
WDSS	Workstation Dependent Segment Storage
WISS	Workstation Independent Segment Storage

C.5 General description

This clause provides a description of all the concepts involved in graphic operations.

For this purpose two groups of basic elements are introduced: PRIMITIVES and ATTRIBUTES.

The PRIMITIVES are abstractions of basic actions a graphics device can perform, such as drawing lines. The ATTRIBUTES specify the characteristics of the OUTPUT PRIMITIVES on a display device, such as colour and line thickness.

Another main concept is that of GRAPHICS WORKSTATION which can be regarded as the abstraction of the collection of graphics input and output devices, i.e. graphics configuration.

Two coordinate systems are provided:

- Normalized Device Coordinates (NDC) used to define a uniform coordinate system for all graphics workstations;
- Device Coordinates (DC), the actual coordinate system of the physical device representing its addressable space.

OUTPUT PRIMITIVES and their ATTRIBUTES may be grouped together in SEGMENTS. SEGMENTS are collections of display elements that can be manipulated and changed as a unit.

In a GDS implementation some primitives are mandatory whereas others are not. Appendix I lists all the primitives defined in this annex and specifies which are mandatory and which are optional.

C.5.1 Graphical output

The basic display elements from which a picture is built up are defined by output primitives. The display elements are specified by their geometry and by their appearance on the display surface of a workstation. These aspects are controlled by a set of attributes that belong to the display element. Certain attributes may vary from one workstation to another. Therefore, they are called workstation dependent attributes.

There are primitives for the creation of display elements and for the setting of attributes. Examples of different display elements are shown in Figure C.2.

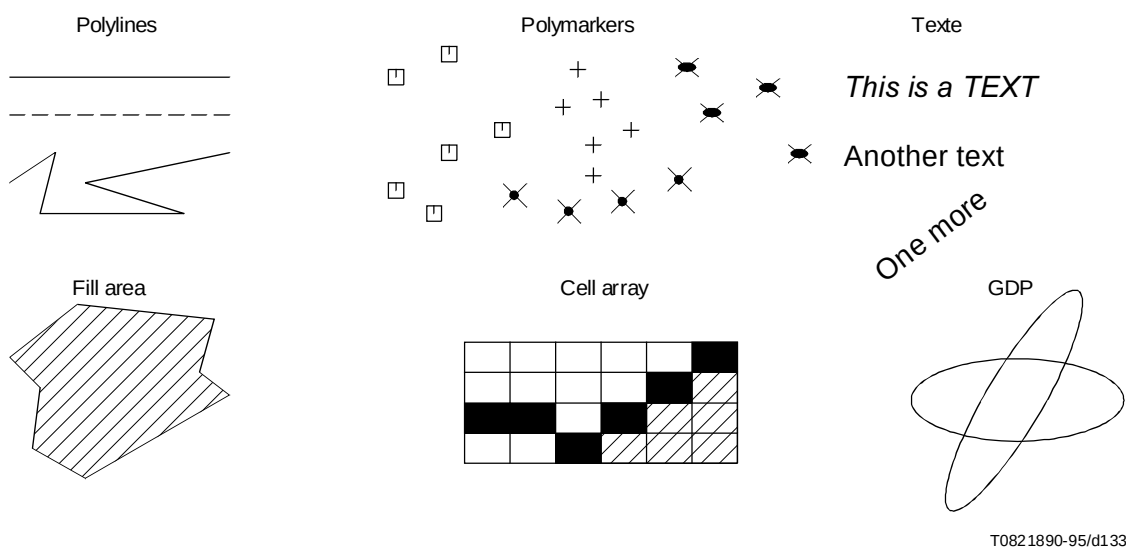


Figure C.2/T.101 – Examples of display elements

C.5.1.1 Output primitives

The following primitives for the creation of display elements are provided:

- POLYLINE

The display element to be created is a set of connected straight lines defined by a sequence of points.

- POLYMARKER

The display element consists of symbols centred at given positions. The symbols, called markers, are glyphs with specified appearances which are used to identify a set of locations.

- TEXT

The display element is a character string placed at a given position.

- FILL AREA

The display element is an area closed by a set of connected straight lines that may be hollow or filled with a uniform colour, a pattern or a hatch style.

- CELL ARRAY

The display element defines a parallelogram of equal sized parallelogram cells with individual colours.

- Generalized drawing primitives (GDPs)

A general display element addressing special geometrical drawing capabilities of a graphics terminal. All transformations are applied to the points of a GDP but the interpretation is workstation dependent. The standardized GDPs are:

- RECTANGLE

The display element is an area with a rectangular boundary that may be hollow or filled.

- CIRCLE

The display element is an area with a circle as boundary that may be hollow or filled.

- CIRCULAR ARC (CENTRE/3 POINT)

The display element is a circular arc.

- CIRCULAR ARC CHORD (CENTRE/3 POINT)

The display element is an area closed by a circular arc and a chord that may be hollow or filled.

- CIRCULAR ARC PIE (CENTRE/3 POINT)

The display element is an area closed by a circular arc and the two radii that may be hollow or filled.

- ELLIPSE

The display element is an area with an ellipse as a boundary that may be hollow or filled.

- ELLIPTIC ARC

The display element is an elliptic arc.

- ELLIPTIC ARC CHORD

The display element is an area closed by an elliptic arc and a chord that may be hollow or filled.

- ELLIPTIC ARC PIE

The display element is an area closed by an elliptic arc and the two radii that may be hollow or filled.

- **SPLINE**

The display element is a smooth curve drawn through a series of control points (uniform quadratic B-SPLINE).

- **NON-STANDARDIZED GDP**

Non-standardized GDPs may be defined for private use. Each GDP is specified by a negative valued identifier, a set of points and additional data.

C.5.1.2 Output primitive attributes

Three types of attributes (geometric attributes, non-geometric attributes and identification) can potentially be specified for each display element. The first two attributes determine the appearance of the display elements while the third is used in connection with graphical input. The values of the attributes can be set modally. During creation of a display element these values are bound to the display element and cannot be changed afterwards.

Geometric attributes control the geometric aspect of display elements which affect shape or size. Hence they are workstation independent. Non-geometric attributes merely affect the appearance (for example line type for POLYLINE) of the display elements. The non-geometric attributes for each primitive may be specified by means of a bundle or individually. For specification of aspects by means of a bundle, there is one attribute per display element which is an index into the bundle table.

For each display element (except for Generalized Drawing Primitives and CELL ARRAY) there is a bundle table. An entry of such a table contains all the non-geometric aspects of a display element.

In this specification mode, the non-geometric attributes are workstation dependent and each workstation has its own set of bundle tables with different values in a particular bundle for different workstations. For individual specification of aspects, there is a separate attribute for each non-geometric aspect. With this specification mode these attributes are workstation independent.

The values that can be assigned to a non-geometric attribute are the same in both specification modes, but in bundled mode the values are restricted to the valid value of each particular workstation. In individual mode if an invalid value of an attribute is set, default actions for the display element are defined to occur.

Generalizing Drawing Primitives and CELL ARRAY do not have associated bundle tables or corresponding individually specified attributes. For each Generalized Drawing Primitive the bundle tables and sets of individually specified attributes are specified to be used when creating the display element. CELL ARRAY contains colour index information, but no other non-geometric aspects.

The method of specification of the non-geometric aspects of a display element may be chosen separately for each aspect. For each non-geometric aspect of each display element exists the attribute ASPECT SOURCE FLAG that takes the values INDIVIDUAL or BUNDLED to specify the choice. The initial values of flags are defined in clause C.9 (Defaults). The values of the flags may be changed by the primitive SET ASPECT SOURCE FLAGS. This enables some non-geometric aspects of a primitive to be specified individually and others bundled.

When a display element is created, the values of the non-geometric attributes with which it is displayed are determined as follows:

- If the ASF of an aspect is INDIVIDUAL, the value used on all workstations is the value of the corresponding individually specified attribute of that display element.
- If the ASF of an aspect is BUNDLED, the value used on a workstation is obtained via the bundle table for that display element on the workstation. The corresponding component of the bundle table entry, pointed to by the bundle index, is used.

If colour is a non-geometric aspect of a display element, it is specified as an index into an unique table (colour table) on each workstation. Similarly, other attribute values may be indices into specific workstation tables or fixed lists.

There is one attribute of the third type per display element, called PICK IDENTIFIER, which is used for identifying a display element or a group of display elements in a segment, when that display element or group is picked. The PICK IDENTIFIER is only used when workstations support input facilities.

The attributes which apply to each display element are:

- | | |
|-------------------------------------|---|
| a) POLYLINE | POLYLINE INDEX
LINE TYPE
LINE WIDTH SCALE FACTOR
POLYLINE COLOUR INDEX
LINE TYPE ASF
LINE WIDTH SCALE FACTOR ASF
POLYLINE COLOUR INDEX ASF
PICK IDENTIFIER |
| b) POLYMARKER | POLYMARKER INDEX
MARKER TYPE
MARKER SIZE SCALE FACTOR
POLYMARKER COLOUR INDEX
MARKER TYPE ASF
MARKER SIZE SCALE FACTOR ASF
POLYMARKER COLOUR INDEX ASF
PICK IDENTIFIER |
| c) TEXT | TEXT INDEX
TEXT FONT AND PRECISION
CHARACTER EXPANSION FACTOR
CHARACTER SPACING
TEXT COLOUR INDEX
TEXT FONT AND PRECISION ASF
CHARACTER EXPANSION FACTOR ASF
CHARACTER SPACING ASF
TEXT COLOUR INDEX ASF
CHARACTER VECTORS
TEXT PATH
TEXT ALIGNMENT
PICK IDENTIFIER |
| d) FILL AREA | FILL AREA INDEX
FILL AREA INTERIOR STYLE
FILL AREA STYLE INDEX
FILL AREA COLOUR INDEX
FILL AREA INTERIOR STYLE ASF
FILL AREA STYLE INDEX ASF
FILL AREA COLOUR INDEX ASF
PATTERN VECTORS
PATTERN REFERENCE POINT
PICK IDENTIFIER |
| e) CELL ARRAY | PICK IDENTIFIER |
| f) GENERALIZED DRAWING
PRIMITIVE | Zero or more of the sets a) to e) except that PICK IDENTIFIER is always an attribute. |

RECTANGLE	set d)
CIRCLE	set d)
CIRCULAR ARC 3 POINT	set a)
CIRCULAR ARC 3 POINT CHORD	set d)
CIRCULAR ARC 3 POINT PIE	set d)
CIRCULAR ARC CENTRE	set a)
CIRCULAR ARC CENTRE CHORD	set d)
CIRCULAR ARC CENTRE PIE	set d)
ELLIPSE	set d)
ELLIPTIC ARC	set a)
ELLIPTIC ARC CHORD	set d)
ELLIPTIC ARC PIE	set d)
SPLINE	set a)

The entries in the bundle, pattern and colour tables may be set separately for each workstation. The tables, which are on every workstation with output facilities, are:

- Polyline bundle table;
- Polymarker bundle table;
- Text bundle table;
- Fill area bundle table;
- Pattern bundle table;
- Colour table.

The values in these tables may be changed. The criterion “dynamic modification accepted” associated with each aspect in a workstation indicates which changes:

- lead to an implicit regeneration (may be deferred);
- can be performed immediately.

Some standard definitions for table entries are contained in a workstation and are used as initial values. Only the most commonly used combinations of values need to be predefined for each output type workstation. The predefined entries with indices up to the minimum number of predefined entries at a given level (see C.5.10) must be distinguishable from each other.

C.5.1.2.1 POLYLINE attributes

POLYLINE has no geometric attributes. The following attributes control the representation of a polyline:

- POLYLINE COLOUR INDEX

Determine the POLYLINE COLOUR INDEX with which the lines will be drawn. It is controlled with SET POLYLINE COLOUR INDEX and is used if the “polyline colour” ASF is INDIVIDUAL.

- LINE WIDTH SCALE FACTOR

Determines the width of the line to be used. The line width is calculated as a nominal line width multiplied by the line width scale factor. This value is mapped by the workstation to the nearest available line width. It is controlled with SET LINE WIDTH SCALE FACTOR and used if the “line width scale factor” ASF is INDIVIDUAL.

- LINE TYPE

Determines the type of the line: solid, dashed, dotted or dash-dotted, etc. The LINE TYPE is selected with SET LINE TYPE and is used if the “line type” ASF is INDIVIDUAL.

- POLYLINE INDEX

Determines the entry of the polyline bundle table to be used in drawing lines. The POLYLINE INDEX is specified with SET POLYLINE INDEX and used for BUNDLED ASFs.

- POLYLINE REPRESENTATION

Determines the attribute values to be loaded in the specified entry of the polyline bundle table. The POLYLINE REPRESENTATION is specified with SET POLYLINE REPRESENTATION and contains the attributes: POLYLINE INDEX, LINE TYPE, POLYLINE COLOUR INDEX and LINE WIDTH SCALE FACTOR.

The polyline bundle table contains three attributes per entry: POLYLINE COLOUR INDEX, LINE TYPE and LINE WIDTH SCALE FACTOR.

C.5.1.2.2 POLYMARKER attributes

POLYMARKER has no geometric attributes. The following attributes control the representation of a polymarker:

- POLYMARKER COLOUR INDEX

Determines the POLYMARKER COLOUR INDEX to be used in drawing the centred markers. It is controlled with SET POLYMARKER COLOUR INDEX and used if the “polymarker colour” ASF is INDIVIDUAL.

- MARKER TYPE

Determines the type of the marker: a dot, a plus, a star, a circle or a diagonal cross, etc. The MARKER TYPE is selected with SET MARKER TYPE and is used if the “marker type” ASF is INDIVIDUAL.

- MARKER SIZE

Determines the size of the marker, e.g. height and width. The marker size is calculated as a nominal size multiplied by the marker size scale factor. This size is mapped by the workstation to the nearest available size. The size is defined with SET MARKER SIZE SCALE FACTOR and is used when the “marker size scale factor” ASF is INDIVIDUAL.

- POLYMARKER INDEX

Determines the entry of the polymarker bundle table to be used in drawing markers. the POLYMARKER INDEX is specified with SET POLYMARKER INDEX and used for BUNDLED ASFs.

- POLYMARKER REPRESENTATION

Determines the attribute values to be loaded in the specified entry of the polymarker bundle table. The POLYMARKER REPRESENTATION is specified with SET POLYMARKER REPRESENTATION and contains the attributes: POLYMARKER INDEX, MARKER TYPE, POLYMARKER COLOUR INDEX and MARKER SIZE SCALE FACTOR.

The polymarker bundle table contains three attributes per entry: POLYMARKER COLOUR INDEX, MARKER TYPE and MARKER SIZE SCALE FACTOR.

C.5.1.2.3 TEXT attributes

Text has the geometric attributes: CHARACTER VECTORS, TEXT PATH AND TEXT ALIGNMENT.

- CHARACTER VECTORS

It represents the character height vector and width vector (defined by direction and length) determining the orientation, skew and distortion of the characters.

The direction of the character height vector fixes the up direction of the character. The length of the character height vector specifies the distance from baseline to capline along the up direction of the character. The direction of the width vector fixes the baseline direction of the character. The length of the character width vector specifies the nominal width of the character. The actual width is the product of the length of the character width vector times the character expansion factor times the width to height ratio of the character.

As a result of a segment transformation, the positive angle from the height vector to the width vector can be greater than 180 degrees. In this case the characters are mirror imaged and the notions of right and left used for TEXT PATH and TEXT ALIGNMENT are reversed.

The CHARACTER VECTORS are specified with SET CHARACTER VECTORS.

- TEXT PATH

Determines the writing direction of a text string. Up (respectively down) means in the direction (respectively opposite direction) of the height vector. Right (respectively left) means in the direction (respectively opposite direction) of the width vector. The TEXT PATH is specified with SET TEXT PATH.

- TEXT ALIGNMENT

Has two components, which are horizontal and vertical alignments.

- HORIZONTAL ALIGNMENT

Determines the horizontal positioning of the text string in relation to the text position: Normal, Left, Centre or Right. The HORIZONTAL ALIGNMENT is specified with SET TEXT ALIGNMENT.

- VERTICAL ALIGNMENT

Determines the vertical positioning of the text string in relation to the text position: Normal, Top, Cap, Half, Base or Bottom. The VERTICAL ALIGNMENT is specified with SET TEXT ALIGNMENT.

The representation of text at a workstation is controlled by:

- TEXT COLOUR INDEX

Determines the TEXT COLOUR INDEX of the generated text string. SET TEXT COLOUR INDEX specifies the TEXT COLOUR INDEX and is used if the “text colour” ASF is INDIVIDUAL.

- TEXT INDEX

Determines the entry of the text bundle table to be used in drawing strings. The TEXT INDEX is specified with SET TEXT INDEX and used for BUNDLED ASFs.

– CHARACTER SPACING

Determines how much additional space is to be inserted between characters. If the value of CHARACTER SPACING is zero, the characters are arranged one after each other along the TEXT PATH. The CHARACTER SPACING may be negative or positive. The CHARACTER SPACING is specified as a fraction of the length of the character height vector. The CHARACTER SPACING is specified with SET CHARACTER SPACING and is used if the “character spacing” ASF is INDIVIDUAL.

– TEXT FONT AND PRECISION

Has two components which are TEXT FONT and TEXT PRECISION:

- TEXT FONT

Determines the TEXT FONT to be used in generating text strings. Each display device should support at least one TEXT FONT which is text font number 0. The TEXT FONT is selected with SET TEXT FONT AND PRECISION and is used if the “text font and precision” ASF is INDIVIDUAL.

- TEXT PRECISION

Determines the accuracy with which a text string is generated: String, Character or Stroke. The TEXT PRECISION is selected with SET TEXT FONT AND PRECISION and is used if the “text font and precision” ASF is INDIVIDUAL.

– CHARACTER EXPANSION FACTOR

Determines the deviation of the width to height ratio of the characters from the width to height ratio indicated by the font designer. The CHARACTER EXPANSION FACTOR is defined by SET CHARACTER EXPANSION FACTOR and is used if the “character expansion factor” ASF is INDIVIDUAL.

– TEXT REPRESENTATION

Determines the attribute values to be loaded in the specified entry of the text bundle table. The TEXT REPRESENTATION is specified with SET TEXT REPRESENTATION and contains the attributes: TEXT INDEX, TEXT COLOUR INDEX, CHARACTER EXPANSION FACTOR, CHARACTER SPACING and TEXT FONT and PRECISION.

HORIZONTAL and VERTICAL ALIGNMENTS both can have the value Normal. For each value of TEXT PATH, the effect of a particular component being Normal is equivalent to one of the other values. The following list applies:

TEXT PATH	Normal HORIZONTAL and VERTICAL ALIGNMENT
Right	(Left, Base)
Left	(Right, Base)
Up	(Centre, Base)
Down	(Centre, Top)

The characters defined in a particular text font are display device dependent. Fonts are defined in a local 2D cartesian coordinate system. Fonts are either monospaced or proportionally spaced. Each character has an associated character body, a font base line, a font half line, a capline and a centre line (see Figure C.3).

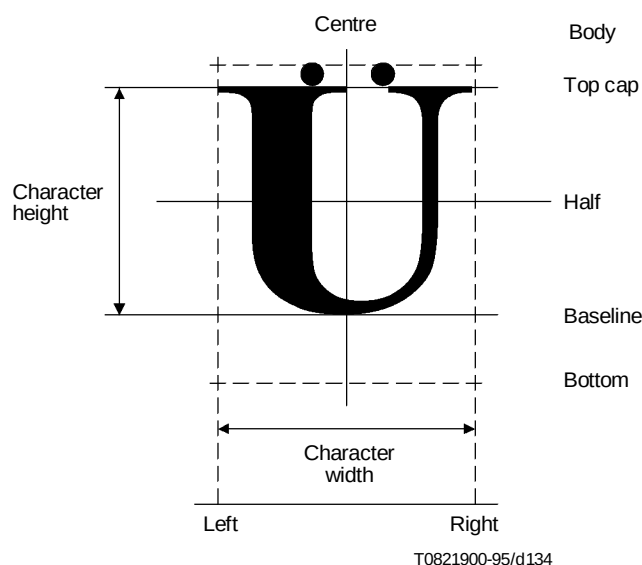


Figure C.3/T.101 – Font description coordinate system

For monospaced fonts the character bodies of all characters have the same size. For proportionally spaced fonts, the width of the bodies may differ from character to character. The character body edges must be parallel to the character vectors. The font baseline and the capline must be parallel to the width vector and within the vertical extent of the body. The centre line is parallel to the height vector and bisects the body.

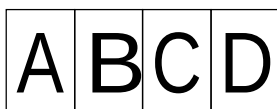
The height of a character in the font coordinate system is given by the height from the font base line to the capline. The width may include space on either side of the character. It is given by the width of the character body. It is assumed that the characters lie within their body, except that kerned characters may exceed the side limits of the character body.

In general, the top limits of the bodies for a font will be identical with or very close to the typographical capline or ascender line and the bottom limit to the descender line. However, these and other details are purely for the use of the font designer. The intention is only that characters placed with their bodies touching in the horizontal direction should give an appearance of good normal spacing and characters touching in the vertical direction will avoid ascender/descender clashes.

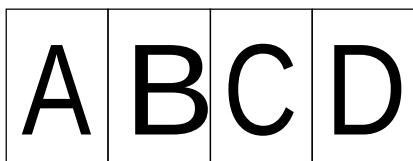
Figures C.4 to C.8 are only inserted for clarification purposes. They show the effects of the different attributes on the display of the text “ABCD”.

Text strings are delimited by the OPEN CHARACTER STRING (OCS) and STRING TERMINATOR (ST) (see C.7.2.4). These delimiters are not considered a part of the text string.

The characters in the text string can be defined in a 7-bit or 8-bit environment defined by ISO 2022. The characters may be taken from the invoked G-set (i.e. from columns 2 to 7) in a 7-bit environment, or from both invoked G-sets (i.e. from columns 02-07 or 10-15) in an 8-bit environment.



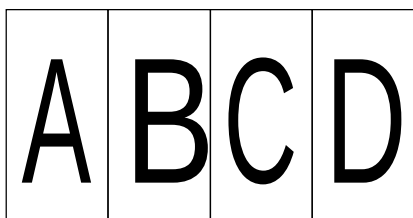
CHARACTER VECTORS = (0.0, 0.025) and (0.025, 0.0)
 CHARACTER EXPANSION FACTOR = 1.0



CHARACTER VECTORS = (0.0, 0.0375) and (0.0375, 0.0)
 CHARACTER EXPANSION FACTOR = 1.0



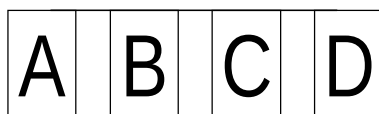
CHARACTER VECTORS = (0.0, 0.025) and (0.025, 0.0)
 CHARACTER EXPANSION FACTOR = 1.5



CHARACTER VECTORS = (0.0, 0.050) and (0.050, 0.0)
 CHARACTER EXPANSION FACTOR = 0.75

T0821910-95/d135

Figure C.4/T.101 – Character vectors and character expansion factor



CHARACTER VECTORS = (0.0, 0.025) and (0.025, 0.0)

CHARACTER SPACING = 0.67

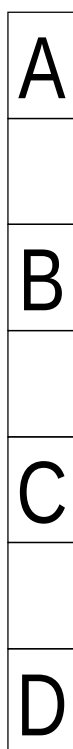
TEXT PATH = right



CHARACTER VECTORS = (0.0, 0.025) and (0.025, 0.0)

CHARACTER SPACING = -0.67

TEXT PATH = right



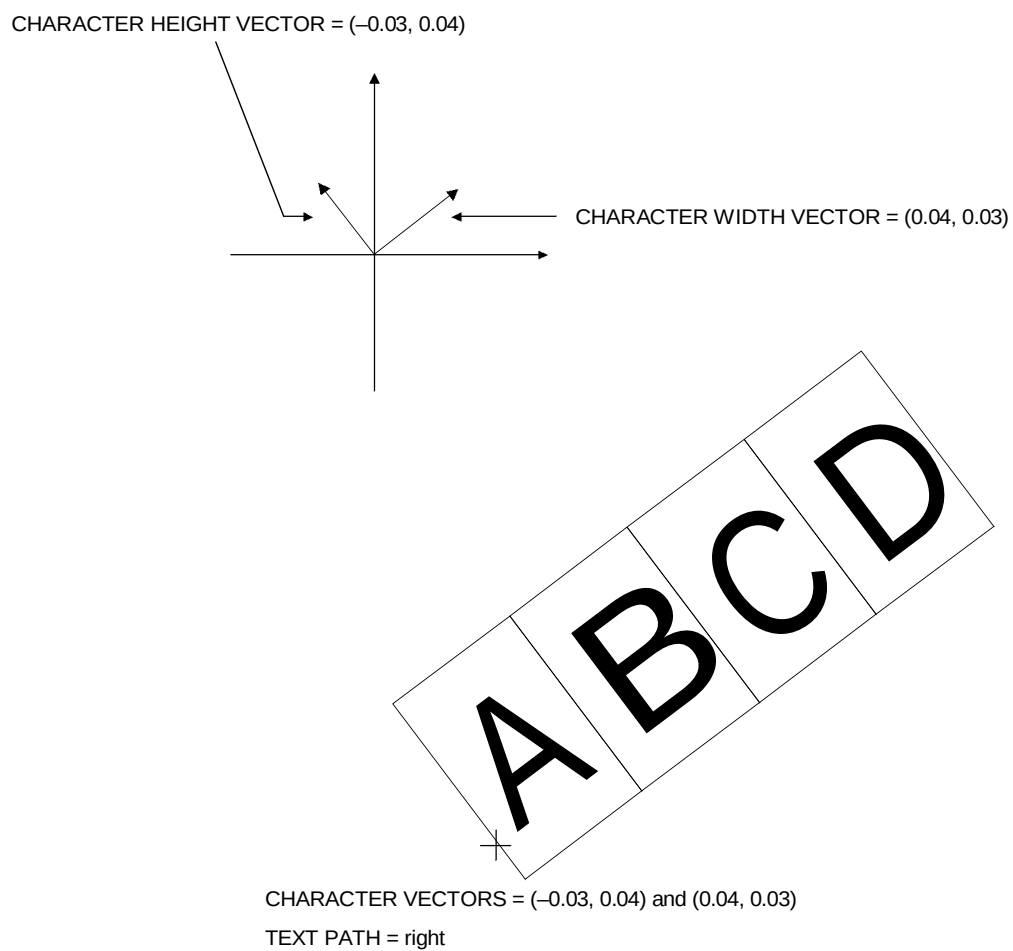
CHARACTER VECTORS = (0.0, 0.025) and (0.025, 0.0)

CHARACTER SPACING = 2.0

TEXT PATH = down

T0821920-95/d136

Figure C.5/T.101 – Character spacing and text path



T0821930-95/d137

Figure C.6/T.101 – Character vectors

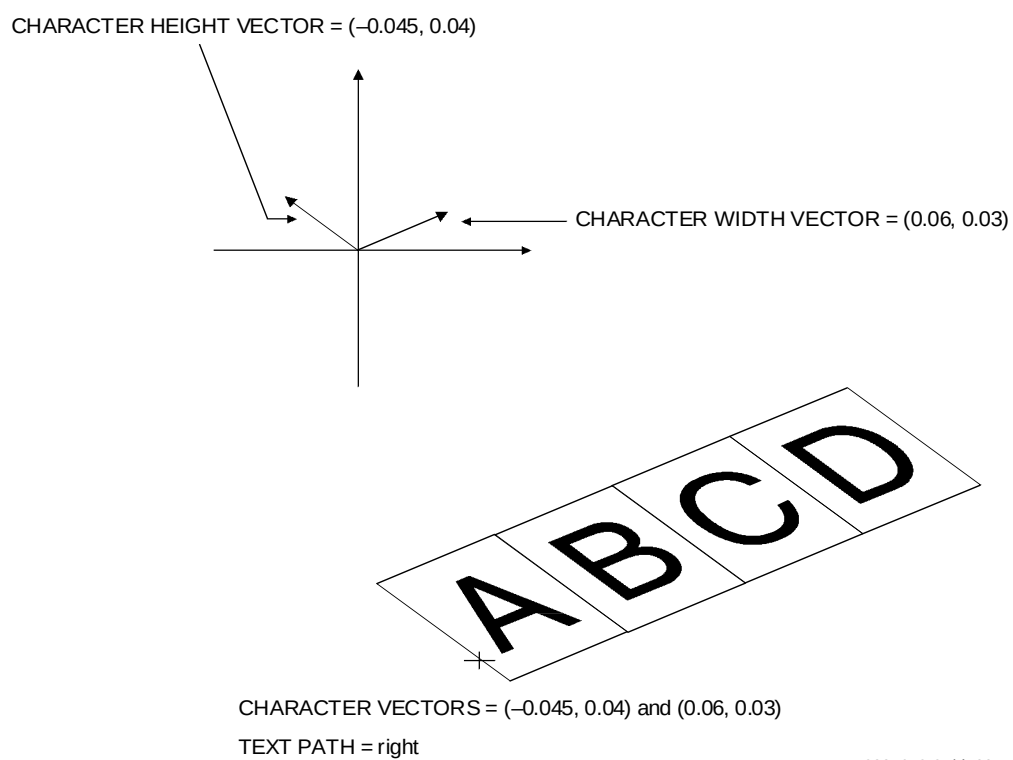
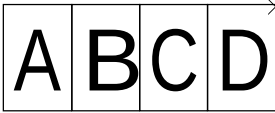


Figure C.7/T.101 – Character vectors after anisotropic transformation


 TEXT ALIGNMENT = (left, base)
 TEXT PATH = right


 TEXT ALIGNMENT = (right, top)
 TEXT PATH = right


 TEXT ALIGNMENT = (centre, bottom)
 TEXT PATH = down


 TEXT ALIGNMENT = (left, half)
 TEXT PATH = down

T0821950-95/d139

Figure C.8/T.101 – Text alignment and text path

Besides the characters from the in-use G-set, in a 7-bit environment, the shift functions contained in Table C.1 may be used for invocation purposes. The coded representation of these shift functions is defined in Part 1 of this annex.

Table C.1/T.101 – Permitted invocation sequences in a 7-bit environment

Abbreviation	Name
SI	SHIFT-IN
SO	SHIFT-OUT
LS2	LOCKING SHIFT 2
LS3	LOCKING SHIFT 3
SS2	SINGLE SHIFT 2
SS3	SINGLE SHIFT 3

In an 8-bit environment the shift functions contained in Table C.2 may be used for invocation purposes. The coded representation of these shift functions is defined in Part 1 of this annex.

Characters in the text string not from the invoked G-set or not from Table C.1, in a 7-bit environment, will be ignored. In an 8-bit environment characters not from the invoked G-sets or not from Table C.2 will be ignored.

Table C.2/T.101 – Permitted invocation sequences in an 8-bit environment

Abbreviation	Name
LS0	LOCKING SHIFT 0
LS1	LOCKING SHIFT 1
LS1R	LOCKING SHIFT 1 RIGHT
LS2	LOCKING SHIFT 2
LS2R	LOCKING SHIFT 2 RIGHT
LS3	LOCKING SHIFT 3
LS3R	LOCKING SHIFT 3 RIGHT
SS2	SINGLE SHIFT 2
SS3	SINGLE SHIFT 3

C.5.1.2.4 FILL AREA attributes

FILL AREA has the geometric attributes PATTERN REFERENCE POINT and PATTERN VECTORS:

- PATTERN VECTORS

Determines the size of the parallelogram in which the pattern cells are defined. The PATTERN VECTORS are defined with SET PATTERN VECTORS and used when the selected (either BUNDLED or INDIVIDUAL) FILL AREA INTERIOR STYLE is Pattern.

- PATTERN REFERENCE POINT

Determines the position of the start of the pattern. The PATTERN REFERENCE POINT is defined with SET PATTERN REFERENCE POINT and used when the selected (either BUNDLED or INDIVIDUALLY) FILL AREA is Pattern.

The representation of FILL AREA at a workstation is controlled by:

- FILL AREA COLOUR INDEX

Determines the colour which is used to fill the closed boundary. It is controlled with SET FILL AREA COLOUR and used if the “fill area colour” ASF is INDIVIDUAL.

- FILL AREA INTERIOR STYLE

Determines how the closed boundary is filled: Hollow, Solid, Pattern or Hatch. The FILL AREA INTERIOR STYLE is selected with SET FILL AREA INTERIOR STYLE and used if the “fill area interior style” ASF is INDIVIDUAL.

- FILL AREA STYLE INDEX

Determines, for FILL AREA INTERIOR STYLE = Hatch, the hatch style to be used and for FILL AREA INTERIOR STYLE = Pattern, the entry from the pattern table to be used. The FILL AREA STYLE INDEX is selected with SET FILL AREA STYLE INDEX and used when the “fill area style index” ASF is INDIVIDUAL.

- **FILL AREA INDEX**

Determines the entry of the fill area bundle table to be used in filling area. The FILL AREA INDEX is specified with SET FILL AREA INDEX and used for BUNDLED ASFs.

- **FILL AREA REPRESENTATION**

Determines the attribute values to be loaded in the specified entry of the fill area bundle table. The FILL AREA REPRESENTATION is specified with SET FILL AREA REPRESENTATION and contains the attributes: FILL AREA INDEX, FILL AREA COLOUR INDEX, FILL AREA INTERIOR STYLE and FILL AREA STYLE INDEX.

The FILL AREA bundle contains three attributes per entry: FILL AREA COLOUR INDEX, FILL AREA INTERIOR STYLE and FILL AREA STYLE INDEX.

The pattern table contains the following attributes per entry:

- **PATTERN DIMENSIONS (DX, DY)**, which define the number of cells in horizontal and vertical directions,
- **PATTERN CELL COLOUR INDEX LIST**, which determines a colour index value for each of the defined cells.

An entry in the pattern table is defined by SET PATTERN REPRESENTATION.

For interior style Pattern, the pattern is defined by the pattern representation, which specifies a pattern cell colour index list, which is conceptually an array (DX * DY) of colour indices, that are pointers into the colour table. The size and position of the start of the pattern are defined by a pattern box. The pattern box, which is a parallelogram, is defined by the PATTERN VECTORS located relative to the PATTERN REFERENCE POINT. The pattern box is conceptually divided into a grid of DX * DY equally sized cells. The colour index array is associated with the cells as follows: the element (1, DY) is associated with the cell having the PATTERN REFERENCE POINT at one corner. Elements with increasing first dimension are associated with successive cells in the direction of the PATTERN WIDTH VECTOR. Elements with decreasing second dimension are associated with successive cells in the direction of the PATTERN HEIGHT VECTOR. The attributes defining the pattern box are subject to all the transformations producing a transformed pattern box. The pattern is mapped onto the closed boundary by conceptually replicating the transformed pattern box in directions parallel to its sides until the interior of the complete closed boundary is covered.

C.5.1.2.5 CELL ARRAY attributes

CELL ARRAY has no attributes other than PICK IDENTIFIER. However, an array of colour indices, which are pointers into the colour table, is part of the definition of a cell array.

C.5.1.2.6 GDP attributes

The GDP primitives that generate closed boundaries:

RECTANGLE
CIRCLE
CIRCULAR ARC 3 POINT CHORD
CIRCULAR ARC 3 POINT PIE
CIRCULAR ARC CENTRE CHORD
CIRCULAR ARC CENTRE PIE
ELLIPSE
ELLIPTIC ARC CHORD
ELLIPSE ARC PIE

use the FILL AREA attributes. These are described in C.5.1.2.4. The GDP primitives that do not generate closed boundaries:

CIRCULAR ARC 3 POINT
CIRCULAR ARC CENTRE
ELLIPSE ARC
SPLINE

use the polyline attributes. These are described in C.5.1.2.1.

C.5.1.2.7 Colour

The colour is specified as an index into a colour table in the workstation. Each workstation has one colour table into which all the colour indices point.

The size of the colour table is workstation dependent but entries 0 and 1 always exist. Entry 0 corresponds to the colour of the display surface after it has been cleared. Entry 1 is the default colour to display pictures and entries higher than 1 correspond to different colours. All entries may be redefined. Entries in the table are set by SET COLOUR REPRESENTATION which specifies the colour as combination of red, green and blue intensities. The specified colour is mapped to the nearest available by the workstation. The accuracy of the intensity of each colour component is set by SET COLOUR HEADER.

On monochrome workstations (workstations only capable of displaying colours with equal intensities of red, green and blue or displaying colours which are different intensities of the same colour), the intensity is computed from the colour values as follows:

$$\text{intensity} = 0.3 * \text{red} + 0.59 * \text{green} + 0.11 * \text{blue}$$

and this intensity is mapped to the nearest intensity available.

C.5.2 Workstations

C.5.2.1 Graphics workstations

This annex is based on the concept of graphics workstations of GKS, which are abstractions of collections of physical devices. The concept of workstation allows to specify device independent applications that can, at the same time, take full advantage of the physical device capabilities. An abstract graphical workstation with maximum capabilities:

- has one addressable display surface of fixed resolution;
- allows only rectangular display spaces, that cannot consist of a number of separate parts;
- permits the specification and use of smaller display spaces than the maximum, while guaranteeing that no display image is generated outside the specified display space;
- supports several line types, text fonts, character sizes, etc., in order to allow output primitives to be drawn with different attributes;
- has one or more logical input devices for each input class and permits different input modes;
- allows storage for short term of output primitives in segments and provides facilities for manipulating them.

C.5.2.2 Workstation characteristics

Each workstation falls into one of the following categories:

- OUTPUT
Output workstation, having a display surface for displaying output primitives (e.g. a plotter).
- INPUT
Input workstation, having at least one input device (e.g. a digitizer, a keyboard).

- OUTIN

Output/input workstation, having a display surface and at least one input device, also called on interactive graphical workstation.

- WISS

Workstation independent segment storage.

A graphics configuration is comprised of one or more workstations, each of which pertains to a given category. As an example, a graphics configuration made up of:

- a CRT and its associated keyboard,
- a printer,
- a diskette.

can be logically interfaced through the following workstations:

- an OUTIN workstation (CRT and keyboard);
- an OUTPUT workstation (printer);
- a workstation independent segment storage (diskette).

A combination of GKS and graphics configuration might be regarded as a GKS instance. Within such an instance one and only one WISS is allowed to exist.

C.5.2.3 Selecting a workstation

The workstations are identified by a workstation identifier. Connection to a particular workstation is established by the primitive OPEN WORKSTATION.

The current state of each open workstation is kept in a workstation state list. Segment manipulations and input can be performed on all open workstations. Output primitives are sent to, and segments are stored on, all active workstations and no others. An open workstation is made active by the primitive ACTIVATE WORKSTATION.

An active workstation is made inactive by the primitive DEACTIVATE WORKSTATION. An open workstation is closed by the primitive CLOSE WORKSTATION.

The following sequence of primitives illustrates workstation selection:

```

OPEN WORKSTATION (N1);
OPEN WORKSTATION (N2);
ACTIVATE WORKSTATION (N1);

    Output primitives:           {generated only on N1}
    Attribute setting;           {possible on N1, N2}

ACTIVATE WORKSTATION (N2);

    Output primitives;           {generated on n1, n2}

DEACTIVATE WORKSTATION (N1);

    Output primitives:           {generated only on N2}
    Attribute setting;           {possible on N1, N2}

CLOSE WORKSTATION (N1);
DEACTIVATE WORKSTATION (N2);
CLOSE WORKSTATION (N2);

```

C.5.2.4 State variable

The set of the state variables is organized into four tables that encode the specific characteristics of a configuration and their evolution. They are called GDS state list, workstation description table, workstation state list and segment state list.

The GDS state list gathers both static and dynamically updated information regarding:

- a) global configuration information (e.g. maximum number of simultaneously open workstations, etc.);
- b) current global values (e.g. set of open workstations, current line type);
- c) last error condition (e.g. error in parameter, etc.).

The GDS state list can be inquired by a primitive that is provided for basic debugging purposes.

The workstation description table gathers only static information concerning the workstation initial state for every single workstation that can be configured onto the graphics configuration. The workstation description table can be inquired by a non-mandatory set of primitives.

The workstation state list is allowed when a configured workstation is effectively opened. It gathers information that changes accordingly to any graphical transaction that will be performed onto the workstation. The workstation state list can be inquired by a non-mandatory set of primitives.

The segment state list gathers global information concerning the segments stored on the workstations and can be inquired by a non-mandatory set of primitives.

C.5.3 Coordinate systems and transformations

C.5.3.1 Coordinate systems

Output devices that are used for representing the visual image of the graphical elements normally require the use of a specific coordinate system. In order to maintain device independency, two coordinate systems have been defined:

- NORMALIZED DEVICE COORDINATE (NDC)

A coordinate specified in a device independent intermediate coordinate system, normalized to some range, including -7 to $+7$, as GKS requires. All output primitives are defined in the NDC space. The coordinates used in constructing display images are expressed in this coordinate system. The actual coordinates are expressed in fractional units based on the Basic Grid Unit (BGU) which is determined by the accuracy of the coordinate encoding.

- DEVICE COORDINATE (DC)

A coordinate specified in the actual coordinate system of the workstation display space. The mapping from NDC to DC is done by the workstation itself. Every workstation may have a different device coordinate space, resulting in a different mapping. The device coordinate system maps onto the display space in the following way:

- 1) the DC origin is at the bottom left corner of the display space;
- 2) the device coordinate units are related to the display space in such a way that a square in device coordinates appears as a square on the display surface;
- 3) x and y increase to the right and upwards respectively.

C.5.3.2 Workstation transformation

The normalized device coordinate space can be regarded as a workstation independent abstract viewing surface. Each workstation can select independently some part of the NDC space in the range $[0.0, 1.0] \times [0.0, 1.0]$ to be displayed somewhere on the workstation display surface. The workstation transformation is a uniform mapping from NDC onto DC and thus performs translation and equal scaling with a positive scale factor for the two axes.

A workstation transformation is specified by defining the limits of an area in the normalized device coordinate system within the range $[0.0, 1.0] \times [0.0, 1.0]$ (WORKSTATION WINDOW) which is to be mapped onto a specified area of the device space (WORKSTATION VIEWPORT) defined in the device coordinate system (see Figure C.9).

Window and viewport limits specify rectangles parallel to the coordinate axes in NDC and DC. The rectangle includes their boundaries. To ensure that no output outside the workstation window is displayed, the picture is clipped at the workstation window boundaries, and this clipping cannot be disabled.

If the workstation window and the workstation viewport have different aspect ratios, the specified scaling would be different on each axis, if the window was mapped onto the viewport in its entirety. To ensure equal scaling on each axis, the transformation maps the window onto the largest rectangle that can fit within the viewport such that (see Figure C.10):

- the aspect ratio is preserved
- the lower left hand corner of the workstation window is mapped to the lower left hand corner of the workstation viewport.

The largest square which fits into the display area and having its bottom and left sides coincident with the bottom and left sides of the rectangular display surface is seen as the default workstation viewport.

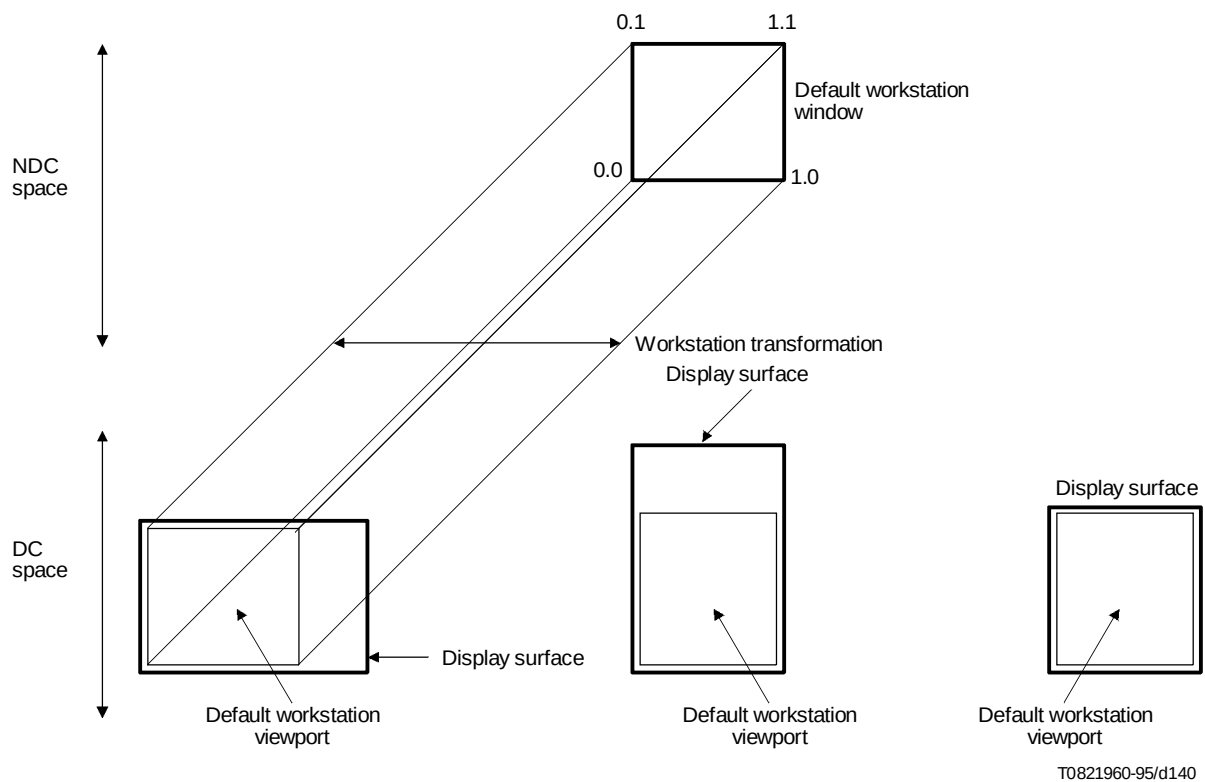


Figure C.9/T.101 – Default workstation transformation

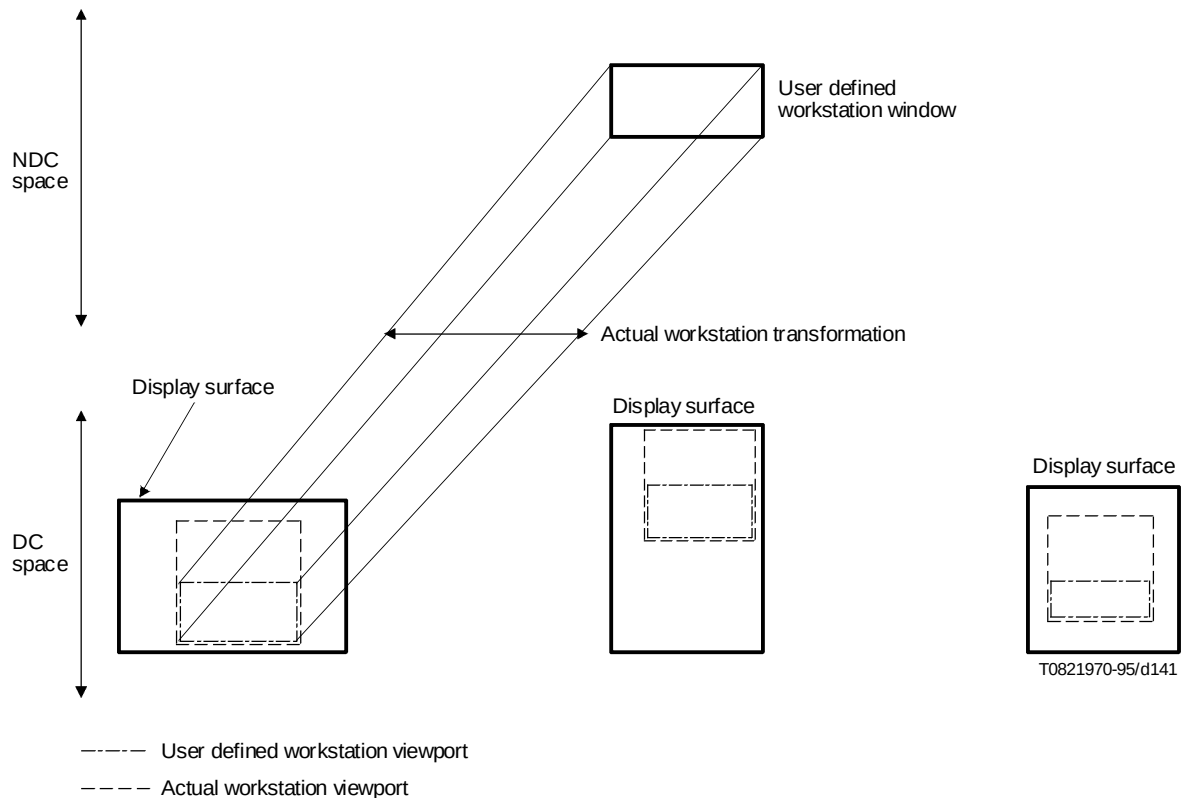


Figure C.10/T.101 – Workstation transformation with anisotropic workstation window and workstation viewport

C.5.3.3 Clipping

Only those parts of the NDC space that lie within user definable rectangles can be shown. Cutting away parts outside such rectangles (CLIPPING RECTANGLE) is called clipping. Clipping takes place when the output primitives are displayed on the display surface of a workstation. Output primitives stored in segments will have the associated clipping rectangle stored with the primitives. An example of clipping and workstation transformation is given in Figure C.11.

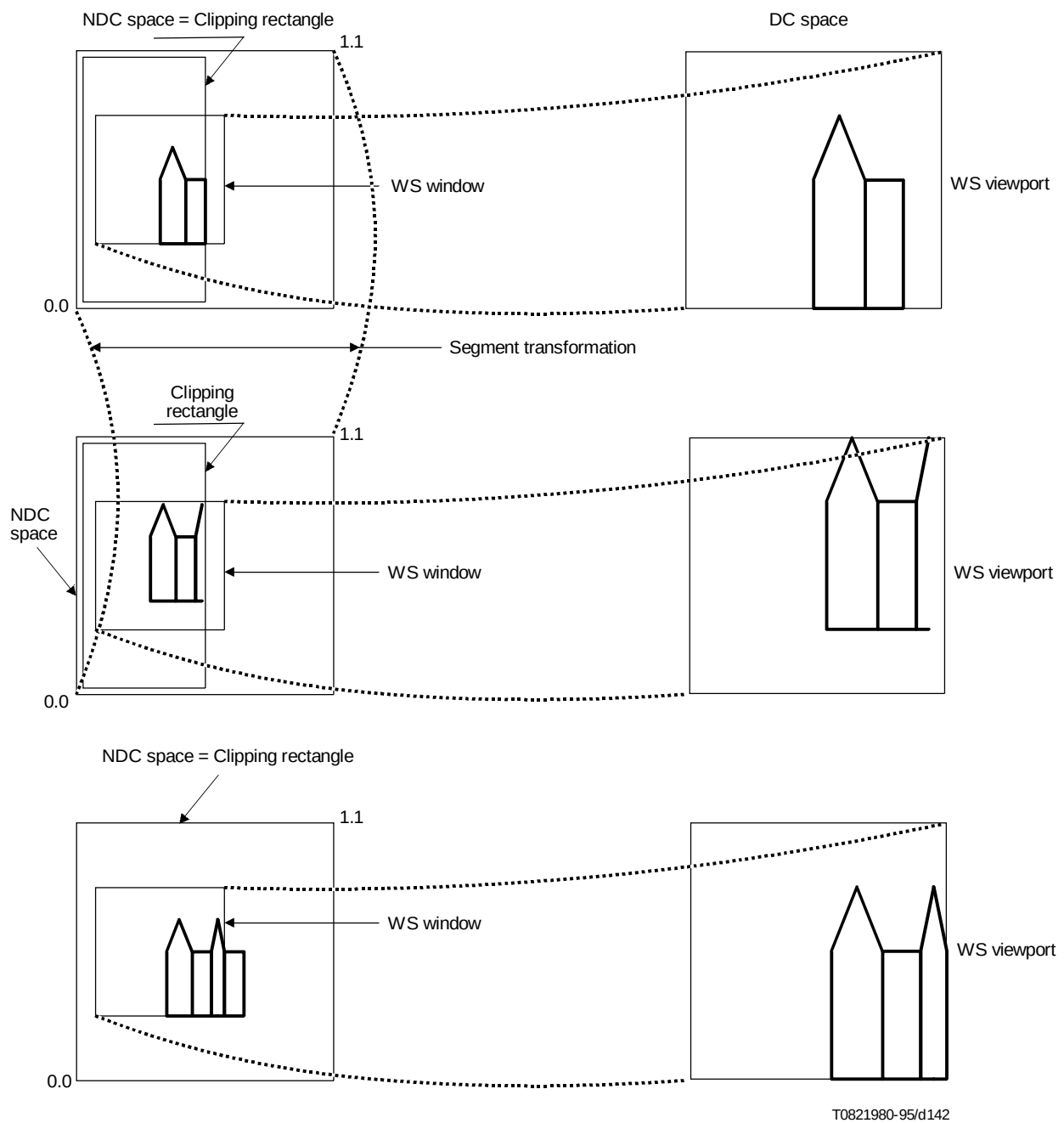
C.5.3.4 Coordinate specification

The coordinates may be specified in two possible modes:

- *Displacement mode*, defining a displacement from the preceding point. For the first point of each point list, the displacement is a displacement from the origin;
- *Incremental mode*, defining steps (increments) from one coordinate position to another.

The GDS does not contain the concept of current position. Therefore, the primitives are logically independent.

The precision of the displacement mode coordinates can be specified by SET COORDINATE PRECISION.



T0821980-95/d142

Figure C.11/T.101 – Clipping and workstation transformation

Incremental mode coordinates are based on a variable ring size and a variable number of points on the ring. The primitive SET DOMAIN RING has two parameters related to the Incremental mode:

- BASIC RADIUS
Determines the size of the ring.
- ANGULAR RESOLUTION FACTOR
Determines the number of points on a ring.

A detailed description of the coordinate data encoding is given in clause C.7.

C.5.4 Segments

C.5.4.1 Concept of segments

A picture is composed of output primitives. They may be grouped into parts that can be addressed and manipulated as a whole. These picture parts are called segments.

Segments are identified by a unique name called segment identifier.

A segment may be:

- transformed;
- made visible or invisible;
- highlighted or not;
- assigned different priorities;
- made detectable or undetectable;
- deleted;
- renamed;
- inserted into the open segment or into the stream of primitives outside segments.

Each segment is stored on all workstations active at the time the segment is created.

All output primitives are collected in a segment after it has been created and until it is closed. After a segment is closed, no primitives can be added to or deleted from the segment.

The output primitives within a segment can have an additional identification that needs not be unique, called pick identifier. It is part of the input value delivered by a pick input device when a segment is picked. The pick identifier has no meaning for workstations of category OUTPUT.

C.5.4.2 Segment attributes

Segment attributes affect all the primitives in a segment. The segment attributes are:

- SEGMENT TRANSFORMATION
A segment may be transformed by translation, rotation, scaling or a combination of them.
- VISIBILITY
A segment is either displayed or not.
- HIGHLIGHTING
A visible segment is either highlighted or not.
- SEGMENT PRIORITY
If parts of segments (for example, FILL AREA, CELL ARRAY) overlap. The segment with the highest priority will be preferred, both when the segments are displayed and when they are picked.
- DETECTABILITY
A segment can either be selected by a pick input device or it cannot.

The segment attributes are unique for each segment and do not vary on different workstations. The default segment attributes (identity transformation, visible, not highlighted, priority 0.0, undetectable) are assigned to a segment when it is created. The segment attributes of any segment in existence, including the open segment, may be changed.

Segment priority affects segments being displayed (i.e. performing segment and workstation transformations, including clipping, for each primitive of the segment). If parts of primitives overlap with others of a visible segment with higher priority, these parts may be invisible. Whether a workstation supports this feature is indicated in the workstation description table. This feature is intended to address appropriate hardware capabilities only. It is not intended to mandate shielding on non-raster displays. When primitives within a segment overlap, the implementation determines the appearance of the overlapped parts.

When primitives of segments overlapping each other are picked, the segment with the highest priority is selected. When primitives of the same segment or of segments with equal priority overlap, the results are implementation dependent.

C.5.4.3 Segment transformations

Segment transformations are a mapping from NDC onto NDC. They perform translation, scaling and rotation.

Segment transformations are characterized by:

- segment name;
- transformation matrix.

The transformation matrix is a 2 by 3 matrix consisting of a 2 by 2 scaling and rotation portion and a 2 by 1 translation portion.

The segment transformation takes place before any clipping.

A segment transformation, specified by the SET SEGMENT TRANSFORMATION primitive, is not actually performed in the segment storage but only saved in the segment state list. Every time the segment is redrawn this segment transformation is applied before clipping. Successive SET SEGMENT TRANSFORMATION primitives for the same segment are not accumulated. Each succeeding transformation matrix replaces its predecessor. By calling SET SEGMENT TRANSFORMATION with an identity transformation matrix, the original segment can be obtained without loss of information.

Note that locator input data is not affected by any segment transformation.

C.5.4.4 Clipping and WDSS

Clipping takes place after the segment transformation has been applied. Each primitive is clipped against the clipping rectangle associated with the primitive when it was put into the segment.

Note that clipping rectangles are not transformed by the segment transformation and thus, clipping is always performed against a rectangle whose edges are parallel to the NDC coordinate axes.

C.5.4.5 Workstation Independent Segment Storage

A Workstation Independent Segment Storage (WISS) is defined, where segments can be stored for use by the COPY SEGMENT TO WORKSTATION, ASSOCIATE SEGMENT WITH WORKSTATION and INSERT SEGMENT primitives. None of these primitives modify the contents of the segments to which they are applied. Only one WISS is permitted in a GKS instance.

The ability to manipulate segments requires the storage of all segments when they are created, so that they can be reused on whatever workstations are active. By contrast, primitives outside segments cannot be re-used.

The treatment of the primitives COPY SEGMENT TO WORKSTATION, ASSOCIATE SEGMENT WITH WORKSTATION and INSERT SEGMENT is explained in Figure C.12.

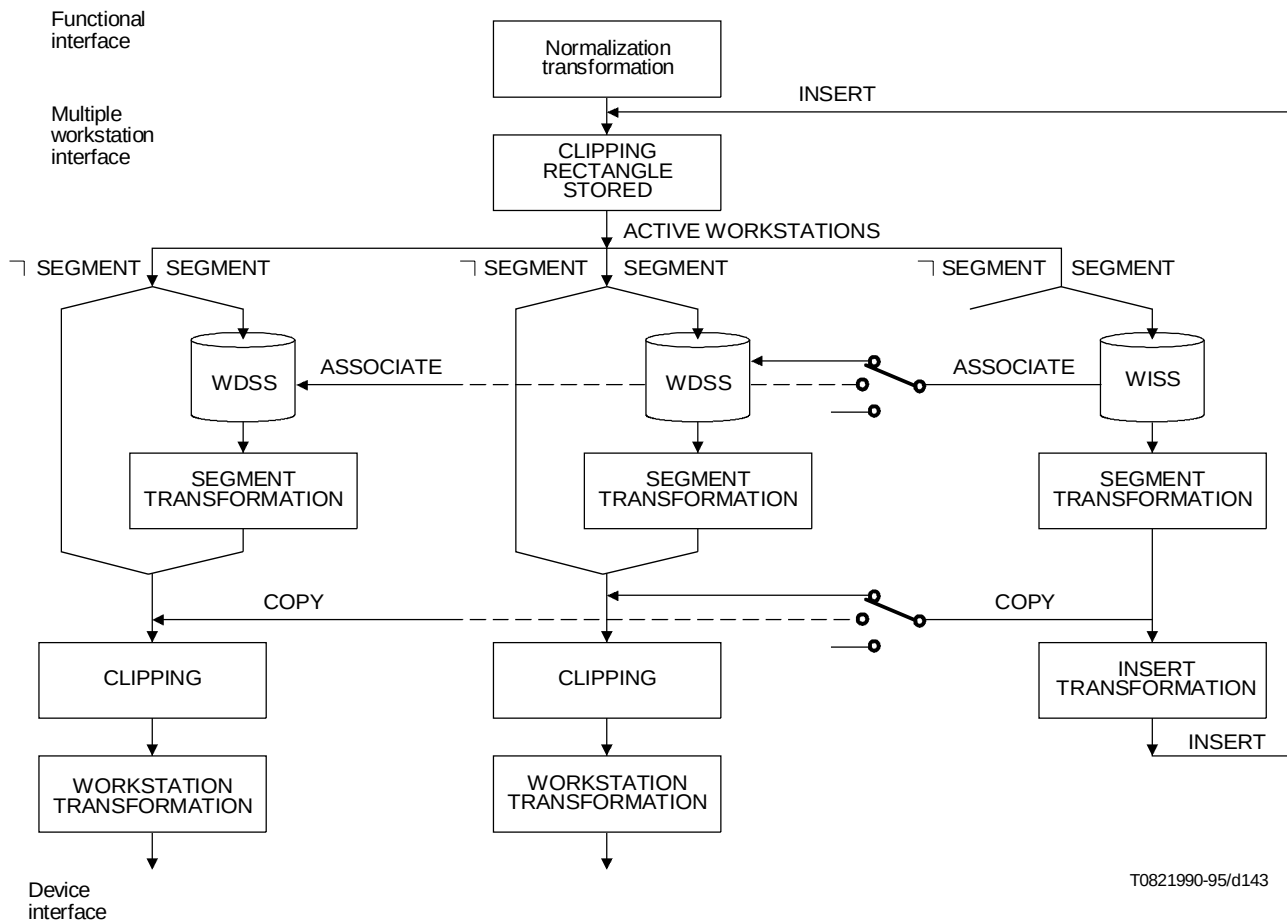


Figure C.12/T.101 – Data flow chart for graphical output

C.5.4.6 WISS functions and clipping

Just as in other workstations, a segment is stored in WISS if WISS is active when the segment is created and the current clipping rectangle is associated with each primitive.

COPY SEGMENT TO WORKSTATION copies primitives from a segment in WISS to be output on the specified workstation. The primitive takes a copy of each primitive and its associated clipping rectangle from a segment in WISS, transforms the primitives by the segment transformation and puts the clipping rectangles and the transformed primitives into the viewing pipeline at the place equivalent to the one where the information left (but it is sent only to the workstation specified in the invocation). This primitive cannot be invoked when a segment is open. By contrast with **ASSOCIATE SEGMENT WITH WORKSTATION**, this primitive does not cause a segment to exist on the specified workstation.

ASSOCIATE SEGMENT WITH WORKSTATION copies the segment to the WDSS of the specified workstation in the same way as if the workstation was active when the segment was created. Clipping rectangles are copied unchanged. This primitive cannot be invoked when a segment is open.

INSERT SEGMENT allows previously stored primitives (in segments in WISS) to be transformed and again placed into the stream of output primitives. INSERT SEGMENT reads the primitives from a segment in the WISS, applies the segment transformation followed by the insert transformation and then inserts them into the viewing pipeline at the point before the data is distributed to the workstations. All clipping rectangles in the inserted segment are ignored. Each primitive processed is assigned a new clipping rectangle which is the current clipping rectangle. Note that all primitives processed by a single invocation of INSERT SEGMENT receive the same clipping rectangle, and that inserted information may re-enter the WISS, if the WISS is active and a segment is open.

An invocation of INSERT SEGMENT has no effect on output primitives passing through the pipeline before or after the invocation. The INSERT SEGMENT primitive can be used when a segment is open but the open segment itself cannot be inserted.

C.5.5 Deferring picture changes

The display of a workstation should reflect as far as possible the actual state of the picture as defined by the sending entity. However, to use efficiently the capabilities of a workstation, the requested action may be delayed for a certain period of time. During this period, the state of the display may be undefined.

A workstation state variable controlling if and how long such a delay of pictures is allowed, is called the deferral state of the workstation. The SET DEFERRAL STATE primitive allows the sending entity to choose that deferral state which takes into account the capabilities of the workstation. Two attributes are defined for this purpose. Deferral model control the time at which output primitives have their visual effect. Implicit regeneration controls the time at which picture changes have their visual effects: picture changes in general imply an alteration, not just an addition to the picture.

The concept of deferral refers only to visible effects of the primitives. Effects on the segments or on the state of the workstation are not deferred.

Deferral mode controls the possible delaying of output primitives. The values of deferral mode (in increasing order of delay) are:

a) **ASAP**

The visual effect of each primitive will be achieved on the workstation As Soon As Possible (ASAP).

b) **BNIG**

The visual effect of each primitive will be achieved on the workstation Before the Next Interaction Globally (BNIG), i.e. before the next interaction with a logical input device gets underway on any workstation. If an interaction on any workstation is already underway, the visual effect will be achieved as soon as possible. This value is meaningful in situations where all the workstations handled by an application through a specific occurrence of GKS System are located in one graphics device.

c) **BNIL**

The visual effect of each primitive will be achieved on the workstation Before the Next Interaction Locally (BNIL), i.e. before the next interaction with a logical input device gets underway on that workstation. If an interaction on that workstation is already underway, the visual effect will be achieved as soon as possible.

d) **ASTI**

The visual effect on each primitive will be achieved on the workstation At Some Time (ASTI).

Deferral applies to the following primitives that generate output:

POLYLINE
POLYMARKER
TEXT
FILL AREA
CELL ARRAY

GENERALIZED DRAWING PRIMITIVES
INSERT SEGMENT
ASSOCIATE SEGMENT WITH WORKSTATION
COPY SEGMENT TO WORKSTATION

Certain primitives can be performed immediately on some workstations, but on other workstations they imply a regeneration of the whole picture to achieve their effect. For example, an implicit regeneration is necessary when picture changes require new paper to be put on a plotter. The entries 'dynamic modification accepted' in the workstation description table indicate which changes:

- a) lead to an Implicit ReGeneration (IRG);
- b) can be performed IMMEDIATELY (IMM).

If changes can be performed immediately, those changes may affect primitives outside segments in addition to those inside segments. If regeneration occurs, all primitives outside segments will be deleted from the display surface.

An implicit regeneration is equivalent to an invocation of the primitive REDRAW ALL SEGMENTS ON WORKSTATION. Its possible delay is controlled by the implicit regeneration mode. The mode can be specified as follows:

- a) SUPPRESSED

Implicit regeneration of the picture is suppressed, until it is explicitly requested: the entry 'new frame necessary at update' is set to YES;

- b) ALLOWED

Implicit regeneration of the picture is allowed.

An implicit regeneration is made necessary if the primitives listed below have a visible effect on the display image of the respective workstation:

- a) If the 'dynamic modification accepted' entry in the workstation description table is IRG (implicit regeneration necessary) for the specified representation:

SET POLYLINE REPRESENTATION
SET POLYMARKER REPRESENTATION
SET TEXT REPRESENTATION
SET FILL AREA REPRESENTATION
SET PATTERN REPRESENTATION
SET COLOUR REPRESENTATION

- b) If the 'dynamic modification accepted' entry in the workstation description table is IRG for the workstation transformation:

SET WORKSTATION WINDOW
SET WORKSTATION VIEWPORT

- c) If the 'dynamic modification accepted' entry in the workstation description table is IRG for segment priority and this workstation supports segment priority:

- 1) If primitives are added to the open segment overlapping a segment of higher priority:

POLYLINE
POLYMARKER
TEXT
FILL AREA
CELL ARRAY
GENERALIZED DRAWING PRIMITIVES
INSERT SEGMENT

(since only segments have priority, primitives outside segments do not make an implicit regeneration necessary).

- 2) If the complete execution of one of the following primitives would be affected by segment priority:

DELETE SEGMENT
DELETE SEGMENT FROM WORKSTATION
ASSOCIATE SEGMENT WITH WORKSTATION
SET SEGMENT TRANSFORMATION
SET VISIBILITY
SET SEGMENT PRIORITY

- d) If the 'dynamic modification accepted' entry in the workstation description table is IRG for segment transformation:

SET SEGMENT TRANSFORMATION

- e) If the 'dynamic modification accepted' entry in the workstation description table is IRG for 'visibility (visible --> invisible)':

SET VISIBILITY (INVISIBLE)

- f) If the 'dynamic modification accepted' entry in the workstation description table is IRG for 'visibility (invisible --> visible)':

SET VISIBILITY (VISIBLE)

- g) If the 'dynamic modification accepted' entry in the workstation description table is IRG for highlighting:

SET HIGHLIGHTING

- h) If the 'dynamic modification accepted' entry in the workstation description table is IRG for delete segment:

DELETE SEGMENT
DELETE SEGMENT FROM WORKSTATION

An implicit regeneration has to be done (including deletion of primitives outside segments) only if one of the primitives listed causes a visible effect on the display: for example, if an invisible segment is deleted, a regeneration does not need to be done. However, an implementation is allowed to perform an implicit regeneration in any of the cases listed above.

Deferred actions can be made visible at any time by the use of the UPDATE WORKSTATION primitive or by an appropriate change of the deferral state.

C.5.6 Graphical input

C.5.6.1 Logical input devices

A Videotex environment will contain no workstations of the type INPUT, so this subclause is not applicable for Videotex purposes.

C.5.6.2 Logical input device model

A Videotex environment will contain no workstations of the type INPUT, so this subclause is not applicable for Videotex purposes.

C.5.6.3 Measures of each input class

A Videotex environment will contain no workstations of the type INPUT, so this subclause is not applicable for Videotex purposes.

C.5.6.4 Input queue and current event report

A Videotex environment will contain no workstations of the type INPUT, so this subclause is not applicable for Videotex purposes.

C.5.6.5 Initialization of input devices

A Videotex environment will contain no workstations of the type INPUT, so this subclause is not applicable for Videotex purposes.

C.5.7 Inquiry

A Videotex environment will support no inquire primitives, so this subclause is not applicable for Videotex purposes.

C.5.8 Error detection

A Videotex environment will support no interactive synchronous error detection mechanism in the sense of what is defined within GKS, so this subclause is not applicable for Videotex purposes.

C.5.9 Error handling

Each primitive will appear in a specified format. The possible forms of each primitive are defined in this annex. The following rules will apply if a workstation detects an error in the received format:

- if a parameter value is not defined or not in the range of allowed values, the default value will be used (e.g. a marker type 6 will default to a marker type 2);
- if a primitive is received that is not supported, the primitive is ignored;
- a primitive, which is not defined will be ignored;
- a primitive with parameters not encoded according to this annex will be ignored;
- if an error in a point list occurs, the processing of the primitive will stop after the previously received correct points have been processed.

As a result of the encoding technique, the skipping of erroneous primitives is always possible.

These general error handling rules may be overruled by specific error handling as described in the discussion subclauses of clause C.6.

C.5.10 Levels

The functional capabilities are grouped into the major areas:

- a) output (minimal performance, full performance);
- b) input (no input, REQUEST input, full input);
- c) number of workstations (one workstation, multiple workstations);
- d) attributes (only predefined bundles and individual attribute specification possible, full bundle concept);
- e) segmentation [none, basic segmentation (without Workstation Independent Segment Storage), full segmentation].

Nine levels are defined in order to allow the implementation of several categories of graphics configurations.

The level structure has two independent axes: input and “all the other primitives”, summarized as output.

The output level axis has the three possibilities:

- 0: minimal output;
- 1: basic segmentation with full output;
- 2: Workstation Independent Segment Storage.

The input level axis has the three possibilities:

- a: no input;
- b: REQUEST input;
- c: full input.

Capabilities are expressed by primitives and by ranges of parameters.

There are three different types of capability at each level:

- An explicitly defined and required capability. Each graphics configuration in conformance with a specific level supports the capability at that level.
- An explicitly defined and non-required capability. A graphics configuration may support the capability and, if it does, it is implemented according to the explicit primitive definitions.
- A conceptually defined and non-required capability. A graphics configuration may provide the capability. Its implementation follows general rules given by the concepts (see C.5.2) and functional definitions.

The set of explicitly defined and required capabilities includes:

- a) predefined bundle entries up to the required minimum;
- b) line types: solid, dashed, dotted and dashed-dotted;
- c) marker types: dot, plus sign, asterisk, circle and diagonal cross;
- d) text precision STROKE (output levels 1 and 2);
- e) interior style HOLLOW;
- f) one input device for each input class (input levels b and c);
- g) prompt and echo type 1 (input levels b and c).

The set of explicitly defined and non-required capabilities includes:

- a) text precision STROKE (output level 0);
- b) interior style SOLID, PATTERN, HATCH;
- c) transformable patterns;
- d) segment priority (output levels 1 and 2);
- e) prompt and echo types (input levels b and c).

The set of conceptually defined and non-required capabilities includes:

- a) line types other than those explicitly defined;
- b) marker types other than those explicitly defined;
- c) specific generalized drawing primitives;
- d) prompt and echo types different from the defined set (input levels b and c);
- e) specific escape primitives.

Explicitly defined and non-required capabilities of a specific level can become explicitly defined and required capabilities in a higher level, through variations in the range of parameters, for example text precision STROKE. Each level contains precisely those primitives that are explicitly defined and required at that level. However, ranges of parameters may contain additional explicitly defined and non-required capabilities and conceptually defined and non-required capabilities.

The facilities making up each of the level components are as follows:

Output level 0: Minimal output

- a) basic control;
- b) all primitives available at least in minimal performance;
- c) use of predefined bundles only (no modification to bundles);
- d) colour representation modification possible;
- e) only one workstation with output capabilities available at a time;
- f) suitable basic inquiries;
- g) pixel readback provided (non-pixel devices may report non-processing).

Output level 1: Basic segmentation with full output

- a) all output level 0 capabilities;
- b) full workstation control;
- c) full output features;
- d) full bundle concept;
- e) multiple workstation concept;
- f) basic segmentation (no Workstation Independent Segment Storage);
- g) suitable inquiries.

Output level 2: Workstation Independent Segment Storage

- a) all output level 1 capabilities;
- b) Workstation Independent Segment Storage.

Input level a: No input

- no facilities.

Input level b: REQUEST input

- a) input device initialization and mode setting primitives;
- b) REQUEST primitives on all appropriate devices;
- c) appropriate logical input device include PICK if and only if combined with output level 1 capabilities.

Input level c: full input

- a) all input level b capabilities;
- b) SAMPLE and EVENT mode input.

Table C.3 gives a short overview of the functionality of each level. Each box contains only those functions added to the previous boxes of the same row and column.

Table C.3/T.101 – Level concept

Output level	Input level		
	a	b	c
0	No input, minimal control, only predefined bundles and all output primitives.	REQUEST input, mode setting and initialize primitives for logical input devices, no PICK.	SAMPLE and EVENT input, no PICK.
1	Full output including full bundle concept, multiple workstation concept basic segmentation, (everything except Workstation Independent Segment Storage).	REQUEST PICK, mode setting and initialize for PICK.	SAMPLE and EVENT input for PICK.
2	Workstation Independent Segment Storage.		

Embedded in the levels summarized above are variations in the number of possibilities required in the set of explicitly defined and required capabilities. Table C.4 exactly identifies the minimum support which is always provided at each level.

Table C.4/T.101 – Minimal support required at each level

	Level								
CAPABILITY	0a	0b	0c	1a	1b	1c	2a	2b	2c
Colours (intensity)	1	1	1	1	1	1	1	1	1
Line types	4	4	4	4	4	4	4	4	4
Line widths	1	1		1	1	1	1	1	1
Predefined polyline bundles	5	5	5	5	5	5	5	5	5
Settable polyline bundles	–	–	–	20	20	20	20	20	20
Marker types	5	5	5	5	5	5	5	5	5
Marker sizes	1	1	1	1	1	1	1	1	1
Predefined polymarker bundles	5	5	5	5	5	5	5	5	5
Settable polymarker bundles	–	–	–	20	20	20	20	20	20
Character vectors (see Note 1)	1	1	1	1	1	1	1	1	1
Character expansion factors (see Note 1)	1	1	1	1	1	1	1	1	1
String precision fonts	1	1	1	1	1	1	1	1	1
Character precision fonts	1	1	1	1	1	1	1	1	1
Stroke precision fonts	0	0	0	2	2	2	2	2	2
Predefined text bundles	2	2	2	6	6	6	6	6	6
Settable text bundles	–	–	–	20	20	20	20	20	20
Predefined patterns (see Note 2)	1	1	1	1	1	1	1	1	1
Settable patterns (see Notes 2 and 5)	–	–	–	10	10	10	10	10	10
Hatch styles (see Note 3)	3	3	3	3	3	3	3	3	3
Predefined fill area bundles	5	5	5	5	5	5	5	5	5
Settable fill area bundles	–	–	–	10	10	10	10	10	10
Segment priorities (see Note 4)	–	–	–	2	2	2	2	2	2
Input classes	–	5	5	–	6	6	–	6	6
Length of input queue (see Note 5)	–	–	20	–	–	20	–	–	20
Maximum string buffer size (characters)	–	72	72	–	72	72	–	72	72
Maximum stroke buffer size (points)	–	64	64	–	64	64	–	64	64
Workstations of category OUTPUT or OUTIN	1	1	1	1	1	1	1	1	1
Workstations of category INPUT or OUTIN	–	1	1	–	1	1	–	1	1
Workstation Independent Segment	–	–	–	–	–	–	1	1	1
0 Indicates explicitly defined and non-required at the level. – Indicates not defined at that level. NOTES 1 Relevant only for character and string precision text. 2 Relevant only for workstation supporting pattern interior style. 3 Relevant only for workstation supporting hatch interior style. 4 Relevant only for workstation supporting segment priorities. 5 Since available resources are finite and entries have variable size, it may not always be possible to achieve the minimal values in a particular graphics configuration.									

C.6 Description of the primitives

C.6.1 Introduction

The primitives are discussed in this clause.

Each primitive is named, the parameters are described, data types are listed and a description of implicit relationship is added.

The order in which parameters will occur in a parameter list is not to be assumed from the order in which they are mentioned in this annex but is deferred to clause C.8.

The list of data types is given below:

<i>Parameter data types</i>		<i>Meaning</i>
P	Point	Two NDC normalized device coordinate values representing the x and y coordinates of a point in NDC space.
CI	Colour Index	Index into a table of colour values.
CL	Colour Index List	List of colour indices encoded in different ways.
CD	Colour Direct	Colour definition with R, G and B intensities.
E	Enumerated type	Set of standardized values. The set is defined by enumerating the identifiers that denote the values.
I	Integer	Number with no fractional part.
ID	Identifier	Name or identifier.
IX	Index	Pointer into a table of values other than colour indices.
M	Matrix	Segment transformation matrix.
REC	Record	Data record.
S	String	Sequence of characters.
V	Size value	Size value in NDC space.
R	Real	Real number.

Combinations of simple types can also be used when n is an unspecified number (for example: nP or 2R, 2I). Also, lists of types can be expressed (for example: I, E, R, E). nP with an unspecified number n denotes an implementation or application dependent number of points called a point list. nP with a specified number n denotes individual points.

C.6.2 Geometric primitives

C.6.2.1 Workstation management primitives

C.6.2.1.1 OPEN WORKSTATION

Level: 0a

Parameters: – Workstation identifier (ID)

Description: The workstation state list is allocated and initialized for the specified workstation. The workstation identifier is added to the set of open workstations in the GDS state list. OPEN WORKSTATION ensures that the display surface is cleared, but does not clear the display surface needlessly.

Related primitives: CLOSE WORKSTATION

Discussion: In the following cases this primitive has no effect:

- the specified workstation is already opened;
- the specified workstation is active;
- the specified workstation identifier is invalid.

C.6.2.1.2 CLOSE WORKSTATION

Level: 0a

Parameters: – Workstation identifier (ID)

Description: An implicit UPDATE WORKSTATION, with the update regeneration flag set to PERFORM, is performed for the specified workstation. The workstation state list is deallocated. The workstation identifier is deleted from the set of open workstations in the GDS state list and from the set of associated workstations in the segment state list of every segment containing it. If the set of associated workstations of a segment becomes empty, the segment is deleted. The display surface need not be cleared when CLOSE WORKSTATION is invoked, but it may be cleared.

Related primitives: OPEN WORKSTATION

Discussion: In the following cases this primitive has no effect:

- the specified workstation is closed;
- the specified workstation identifier is invalid;
- the specified workstation is active.

C.6.2.1.3 ACTIVATE WORKSTATION

Level: 0a

Parameters: – Workstation identifier (ID)

Description: The specified workstation is marked active in the workstation state list. The workstation identifier is added to the set of active workstations in the GDS state list.

Related primitives: OPEN WORKSTATION
CLOSE WORKSTATION
DEACTIVATE WORKSTATION

Discussion: Output primitives are sent to and segments are stored on all active workstations. In the following cases this primitive has no effect:

- the specified workstation is activated;
- the specified workstation is closed;
- the specified workstation identifier is invalid.

C.6.2.1.4 DEACTIVATE WORKSTATION

Level: 0a

Parameters: – Workstation identifier (ID)

Description: The specified workstation is marked inactive in the workstation state list. The workstation identifier is deleted from the set of active workstations in the GDS state list.

Related primitives: OPEN WORKSTATION
CLOSE WORKSTATION
ACTIVATE WORKSTATION

Discussion: While a workstation is inactive, it will not process output primitives nor does it store new segments. Segments already stored on this workstation are retained.

In the following cases this primitive has no effect:

- the specified workstation is not activated;
- the specified workstation identifier is invalid.

C.6.2.1.5 CLEAR WORKSTATION

Level: 0a

Parameters: – Workstation identifier (ID)

Description: The effect of this primitive depends on the workstation category:

1) OUTPUT, OUTIN and WISS workstations:

The following actions are executed in the given sequence:

- a) All deferred actions for the specified workstation are executed (without intermediate clearing of the display surface).
- b) The display surface is always cleared.
- c) The current workstation transformation is set to the last defined WORKSTATION WINDOW and WORKSTATION VIEWPORT, if necessary.
- d) All segments stored for the specified workstation are deleted.

2) Other workstations:

No effect.

Related primitives: None.

Discussion: If the workstation identifier parameter is out of range, the primitive is ignored.

C.6.2.1.6 SET DEFAULTS

Level: 0a

Parameters: None.

Description: This primitive places each workstation status or attributes to its default, as defined in clause C.9.

Related primitives: None.

Discussion: None.

C.6.2.1.7 GDS ESCAPE1

Level: 0a

Parameters: – GDS escape identifier (I)
– GDS data record (REC)

Description: The GDS ESCAPE1 primitive allows use of device capabilities not specified by this annex. This primitive is best suited for access to non-standardized control features of graphics devices.

The specific function specified by the GDS escape identifier is invoked. Non-negative values of the GDS escape identifier are reserved for future standardization. Negative values are available for implementation dependent use.

The GDS escape data record depends on the function being performed.

Related primitives: None.

Discussion: None.

C.6.2.2 Output workstation primitives

C.6.2.2.1 Output drawing primitives

C.6.2.2.1.1 POLYLINE

Level: 0a

Parameters: – Point list (2..n) (nP)

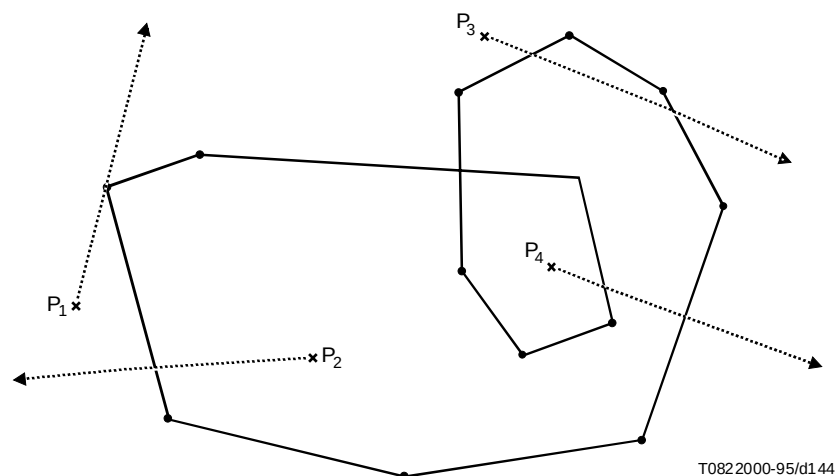
Description:	A line is drawn from the starting point to the second point, ..., from the next-to-last point to the ending point.
Related primitives:	SET POLYLINE INDEX SET ASPECT SOURCE FLAGS SET LINE TYPE SET LINE WIDTH SCALE FACTOR SET POLYLINE COLOUR INDEX
Discussion:	If only one point is specified, the primitive is ignored. The implementation of a zero length line segment is implementation dependent.

C.6.2.2.1.2 POLYMARKER

Level:	0a
Parameters:	– Point list (1..n) (nP)
Description:	The marker corresponding to the currently selected marker type is drawn at each of the points in the point list. If the marker type is one of the pre-defined markers, it is drawn centred at each of the points. Other implementation dependent markers may have other alignments where desired. If the resulting marker is completely within the clipping area, the entire marker is drawn. If any part of the marker would have to be executed outside the clipping rectangle, the result is device dependent.
Related primitives:	SET POLYMARKER INDEX SET ASPECT SOURCE FLAGS SET MARKER TYPE SET MARKER SIZE SCALE FACTOR SET POLYMARKER COLOUR INDEX
Discussion:	None.

C.6.2.2.1.3 FILL AREA

Level:	0a
Parameters:	– Point list (3..n) (nP)
Description:	A boundary of a polygonal region is defined by connecting each vertex to its successor in the ordered point list and connecting the last vertex to the first. The polygonal region may be non-simple. For example, edges are allowed to cross. In this way subareas can be created. Any given point is considered inside the polygon if a straight line from the given point to infinity intersects the polygon edges an odd number of times. If this line passes through a vertex point tangentially, the intersection count is not changed. If a polygon is clipped and subareas are generated, the new boundaries become part of the polygon boundaries. An example is given below:



Points P_3 and P_4 are considered to be outside the polygon, because the intersection count is even ($= 2$). Point P_2 is considered to be inside the polygon, because the intersection count is odd ($= 1$). Point P_1 is considered to be outside the polygon, because the intersection count is even ($= 0$). The line to infinity from P_1 passes through a vertex point tangentially, but this does not affect the intersection count.

A non-degenerate polygon (one with three or more vertices, not all of which are colinear) is displayed with interior as specified by the SET FILL AREA INDEX, SET ASPECT SOURCE FLAGS and SET FILL AREA INTERIOR STYLE primitives. If a point is within the polygon, it is included in the area to be filled subject to the following rules for boundaries. The boundary is drawn for interior style HOLLOW and is not drawn for other interior styles.

Related primitives: SET FILL AREA INDEX
SET ASPECT SOURCE FLAGS
SET FILL AREA INTERIOR STYLE
SET FILL AREA STYLE INDEX
SET PATTERN REPRESENTATION
SET PATTERN VECTORS
SET PATTERN REFERENCE POINT
SET FILL AREA COLOUR INDEX

Discussion: If less than three points are specified, the primitive is ignored.

If the list contains only colinear vertices, a straight line is drawn through them.

If all specified points coincide, a dot is displayed.

C.6.2.2.1.4 TEXT

Level: 0a

Parameters: – Text position (P)
– Character string (S)

Description: The character codes specified in the string are interpreted to obtain the associated symbols. Characters are displayed on the viewing surface as specified by the text attributes.

The characters are dimensioned according to the SET CHARACTER VECTORS, font-dependent character aspect ratio and SET CHARACTER EXPANSION FACTOR and are oriented according to SET CHARACTER VECTORS. The direction of the character placement in the string relative to SET CHARACTER VECTORS is controlled by SET TEXT PATH.

Related primitives: SET TEXT FONT AND PRECISION
SET TEXT INDEX
SET ASPECT SOURCE FLAGS
SET CHARACTER EXPANSION FACTOR
SET CHARACTER SPACING
SET TEXT COLOUR INDEX
SET CHARACTER VECTORS
SET TEXT PATH
SET TEXT ALIGNMENT

Discussion: None.

C.6.2.2.1.5 CELL ARRAY

Level: 0a

Parameters: – Parallelogram (P, Q, R) (3P)
– Number of cells in P-R direction m (I)
– Number of cells in Q-R direction n (I)
– Cell colour index list (CL)

Description: The points P, Q and R define a parallelogram. P and Q are the end points of a diagonal of the parallelogram. R defines a third corner.

This parallelogram is subdivided in $m * n$ contiguous parallelograms in the following way. The side PR of the parallelogram is subdivided into m intervals of equal size. The side QR of the parallelogram is subdivided into n intervals of equal size. The grid, implied by this subdivision, consists of $m * n$ equally dimensioned cells.

The cell colour index list consists of $m * n$ colour indices, conceptually an array of dimensions m and n representing respectively the column and row dimensions. Array element (1, 1) is mapped to the cell associated with P and array element (m, 1) is mapped to the cell associated with R. Array element (m, n) is mapped to the cell associated with Q. Hence, the colour elements are mapped within rows running from P to R, and with the rows incrementing in order from R to Q. This is illustrated in Figure C.13. No current colour setting is changed.

Related primitive: None.

Discussion: The parameters and actions of this primitive supersede any setting of any other primitive. Specifically, individual or bundled colour specifiers are ignored.

If the number of indices in the colour index list is less than $m * n$, the last index is repeated until the end of the list. If the number of indices in the colour index list is greater than $m * n$, the exceeding indices are ignored.

If the number of cells (parameters m and n) are less than or equal to 0, the primitive is ignored.

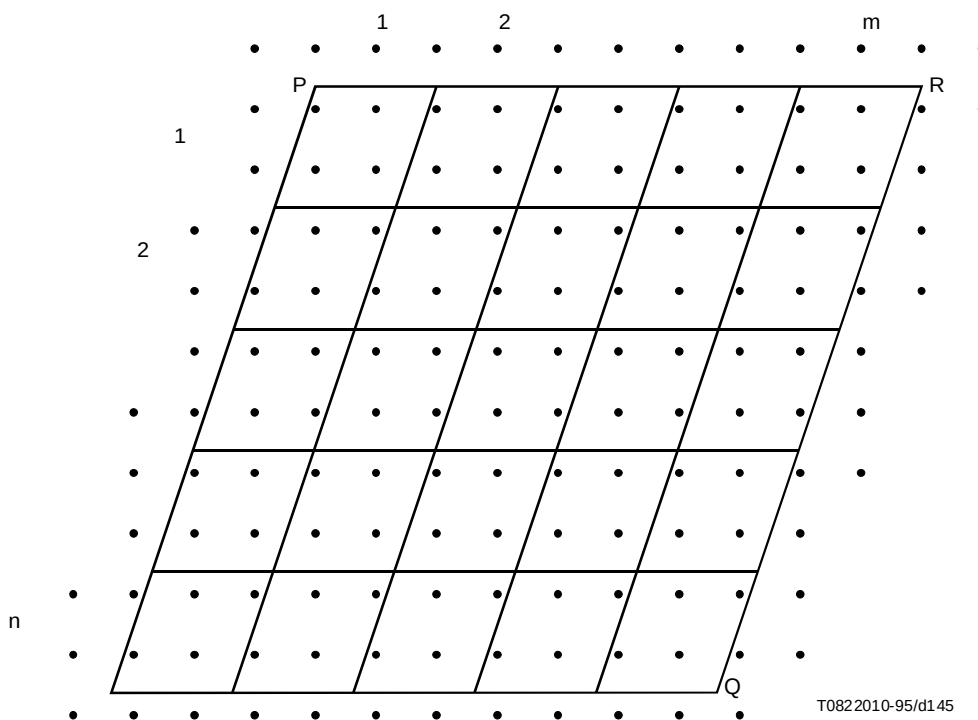


Figure C.13/T.101 – m by n parallelograms mapped onto the display surface

C.6.2.2.1.6 GDP

Level: 0a

Parameters: – GDP identifier (I)
– Point list (nP)
– Data record (REC)

Description: A Generalized Drawing Primitive (GDP) is specified by the identifier, the data of the point list and the data record. The appearance of the GDP is determined by zero or more of the attribute sets of the standardized output primitives, depending on the particular GDP. These attributes are listed in C.5.1.2.

Non-negative values of the identifier are reserved for standardization and negative values are available for private use.

Related primitives: Not standardized.

Discussion: The GDP is displayed on all active workstations capable of doing so.

The points specified as parameters are transformed after the interpretation of the points is performed by the active workstations. For example, a GDP, which defines a circle, would appear as an ellipse when the transformation has different scaling for the two axes.

The resulting output of the GDP is clipped against the clipping rectangle.

C.6.2.2.1.6.1 GDP (rectangle)

Level: 0a

Parameters: – GDP identifier (I)
– (= rectangle)
Two points (2P)

Description: A rectangle is specified by a pair of points. These points are the opposite corners of the rectangle having its sides parallel to the axes.

The rectangle is displayed with interior style as defined by SET FILL AREA INDEX, SET ASPECT SOURCE FLAGS and SET FILL AREA INTERIOR STYLE.

Related primitives: Same as for FILL AREA.

Discussion: If the two points define a line parallel to either the x or y axis, the degenerated form of the rectangle is the line.

If the two points are identical, the degenerate form of the rectangle is the defined point.

C.6.2.2.1.6.2 GDP (circle)

Level: 0a

Parameters: – GDP identifier (I)
– (= circle)
– Centre (P)
– Radius (V)

Description: A circle of the specified radius at the specified centre position as displayed with interior style as defined by SET FILL AREA INDEX, SET ASPECT SOURCE FLAGS and SET FILL AREA INTERIOR STYLE.

Related primitives: Same as for FILL AREA.

Discussion: If the radius parameter has the value 0, a dot is displayed at the centre position.

If the radius parameter is negative, the primitive is ignored.

C.6.2.2.1.6.3 GDP (circular arc 3 point)

Level:	0a
Parameters:	– GDP identifier (I) – (= circular arc 3 point) – Starting point, intermediate point, ending point (3P)
Description:	A circular arc is displayed from the starting point, through the specified intermediate point, to the specified ending point.
Related primitives:	Same as for POLYLINE.
Discussion:	If the starting point and the ending point are identical, a circle is displayed which passes through the specified points and has a diameter equal to the distance from the starting point to the intermediate point. The resulting circle is not considered to be a closed area, so it does not have an interior style. If the intermediate point coincides with the starting or ending point, a straight line between the two points is displayed. If the three points coincide, a dot is displayed. If the three points are collinear, a straight line from the starting point through the intermediate point to the ending point is drawn.

C.6.2.2.1.6.4 GDP (circular arc 3 point chord)

Level:	0a
Parameters:	– GDP identifier (I) – (= circular arc 3 point chord) – Starting point, intermediate point, ending point (3P)
Description:	A circular arc 3 point chord defined by three points (starting point, intermediate point and ending point) is a boundary closed by the circular arc specified by the three points and the chord from the starting point to the ending point. Such a circular arc 3 point chord is displayed with interior style as defined by SET FILL AREA INDEX, SET ASPECT SOURCE FLAGS and SET FILL AREA INTERIOR STYLE.
Related primitives:	Same as for FILL AREA.
Discussion:	If the starting point and ending point are identical, a filled area bounded by the circle which passes through the specified points and has a diameter equal to the distance from the starting point to the intermediate point is displayed. If the intermediate point coincides with the starting or ending point, a straight line between the two points is drawn. If the three points coincide, a dot is displayed. If the three points are collinear, a straight line from the starting point through the intermediate point to the ending point is drawn.

C.6.2.2.1.6.5 GDP (circular arc 3 point pie)

Level:	0a
Parameters:	– GDP identifier (I) – (= circular arc 3 point pie) – Starting point, intermediate point, ending point (3P)
Description:	A circular arc 3 point pie defined by three points (starting point, intermediate point, ending point) is a boundary closed by the arc specified by the three points and the two radii from starting point (respectively ending point) to the computed centre.

Such a circular arc 3 point pie is displayed with interior style as defined by SET FILL AREA INDEX, SET ASPECT SOURCE FLAGS and SET FILL AREA INTERIOR STYLE.

Related primitives: Same as for FILL AREA.

Discussion: If the starting point and ending point of a circular arc 3 point pie are identical, a filled area bounded by the circle which passes through the specified points and has a diameter equal to the distance from the starting point to the intermediate point is displayed.

If the intermediate point coincides with the starting or ending point, a straight line between the two points is drawn.

If the three points coincide, a dot is displayed.

If the three points are collinear, the primitive is ignored.

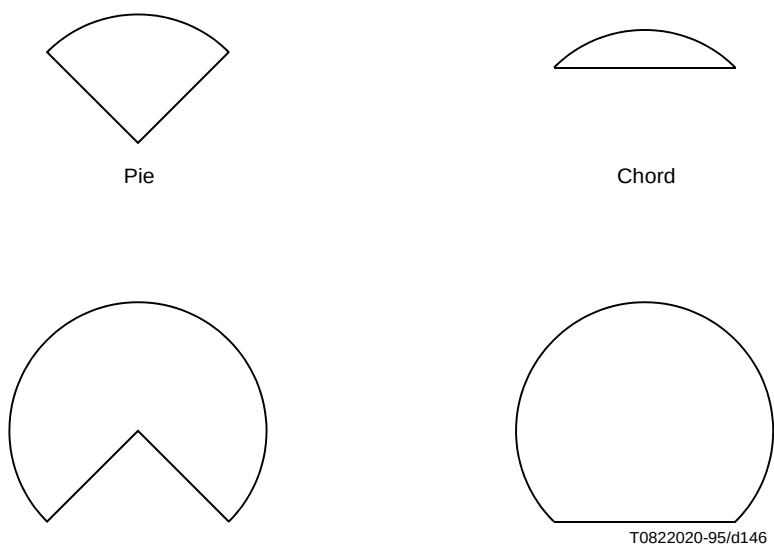


Figure C.14/T.101 – GDP specifications for “arc pie” and “arc chord”

C.6.2.2.1.6.6 GDP (circular arc centre)

Level: 0a

- Parameters:
- GDP identifier (I)
(= circular arc centre)
 - Centre (P)
 - Radius (V)
 - Start vector (P)
 - End vector (P)

Description: The radius parameter and the centre parameter together define a circle. The fourth parameter, together with the centre parameter, defines the start vector. The fifth parameter, together with the centre parameter, defines the end vector.

The starting (respectively ending) point of the circular arc centre is obtained by measuring a distance equal to the radius parameter along the start (respectively end) vector. The circular arc centre is drawn counterclockwise along the circle from starting point to ending point.

Related primitives: Same as POLYLINE.

Discussion: If the radius parameter has the value 0, a dot is displayed at the centre point.

If the radius parameter is negative, the primitive is ignored.

If the start vector and the end vector coincide, the primitive is ignored.

If a segment transformation is applied to the points defining a circular arc centre, the transformed points again define a circular arc centre in the same way as described above. This may result in unwanted effects.

C.6.2.2.1.6.7 GDP (circular arc centre chord)

Level: 0a

Parameters:

- GDP identifier (I)
(= circular arc centre chord)
- Centre (P)
- Radius (V)
- Start vector (P)
- End vector (P)

Description: A circular arc centre chord is a boundary closed by the circular arc, specified by the parameters, and the chord from starting point of the circular arc to ending point of the circular arc (see definition of circular arc centre for the construction of the circular arc, starting point and ending point).

Such a circular arc centre chord is displayed with an interior style as defined by SET FILL AREA INDEX, SET ASPECT SOURCE FLAGS and SET FILL AREA INTERIOR STYLE.

Related primitives: Same as FILL AREA.

Discussion: If the radius parameter has the value 0, a dot is displayed at the centre point.

If the radius parameter is negative, the primitive is ignored.

If the start vector and the end vector coincide, the primitive is ignored.

If a segment transformation is applied to the points defining a circular arc centre chord, the transformed points again define a circular arc centre chord in the same way as described above. This may result in unwanted effects.

C.6.2.2.1.6.8 GDP (circular arc centre pie)

Level: 0a

Parameters:

- GDP identifier (I)
(=circular arc centre pie)
- Centre (P)
- Radius (V)
- Start vector (P)
- End vector (P)

Description: A circular arc centre pie is a boundary closed by the circular arc, specified by the parameters, and the two radii from starting point (respectively ending point) of the circular arc to the centre (see definition of circular arc centre for the construction of the circular arc, starting point and ending point).

Such a circular arc centre pie is displayed with an interior style as defined by SET FILL AREA INDEX, SET ASPECT SOURCE FLAGS and SET FILL AREA INTERIOR STYLE.

Related primitives: Same as FILL AREA.

Discussion: If the radius parameter has the value 0, a dot is displayed at the centre point.

If the radius parameter is negative, the primitive is ignored.

If the start vector and the end vector coincide, the primitive is ignored.

If a segment transformation is applied to the points defining a circular arc centre pie, the transformed points again define a circular arc centre pie in the same way as described above. This may result in unwanted effects.

C.6.2.2.1.6.9 GDP (ellipse)

Level: 0a

Parameters:

- GDP identifier (I)
(= ellipse)
- Centre point (P)
- Endpoints (X1, Y1) and (X2, Y2) (2P)

Description: An ellipse is specified by a Conjugate Diameter Pair (CDP). If the centre point is (XM, YM), the endpoints of an ellipse Conjugate Diameter Pair are defined by the two endpoints. The CDP vector components, relative to the centre point, are defined as follows:

$$DX1 = X1 - XM$$

$$DY1 = Y1 - YM$$

$$DX2 = X2 - XM$$

$$DY2 = Y2 - YM$$

The CDP vector components are the coefficients of the parametric equations:

$$X = XM + DX1 * \cos(t) + DX2 * \sin(t)$$

$$Y = YM + DY1 * \cos(t) + DY2 * \sin(t)$$

When $t = 0$, (X, Y) is the endpoint of the first conjugate diameter, when $t = \pi/2$, (X, Y) is the endpoint of the second conjugate diameter, and when $t = 2 * \pi$, the ellipse is closed and (X, Y) is again the endpoint of the first conjugate diameter.

The ellipse so defined is displayed with interior style as specified by SET FILL AREA INDEX, SET ASPECT SOURCE FLAGS and SET FILL AREA INTERIOR STYLE.

Related primitives: Same as for FILL AREA.

Discussion: If the three points coincide, a dot is displayed.

If the three points are collinear, a straight line connecting the two endpoints is displayed.

If any two of the three points are coinciding, the primitive is ignored.

C.6.2.2.1.6.10 GDP (elliptic arc)

Level: 0a

Parameters:

- GDP identifier (I)
(= elliptic arc)
- Centre point (P)
- Endpoints (X1, Y1) and (X2, Y2) (2P)
- T_start, T_end (2R)

Description: The centre point and the two endpoints together define an ellipse (see C.6.2.2.1.6.5).

T_start and T_end correspond to two points on the ellipse. The defined elliptic arc is that which begins at T_start and goes to T_end in the direction established by the coefficients of the parametric equations (the CDP vector components DX1, DY1, DX2, DY2).

The elliptic arc is displayed with the attributes defined for POLYLINE.

Related primitives: Same as for POLYLINE.

Discussion: If the centre point and the endpoints coincide, a dot is displayed.

If the centre point and the endpoints are collinear, a straight line connecting the two endpoints is displayed.

If any two of the centre point and the two endpoints are coinciding, the primitive is ignored.

If T_start equals T_end, the full ellipse is displayed.

C.6.2.2.1.6.11 GDP (elliptic arc chord)

Level: 0a

Parameters:

- GDP identifier (I)
(= elliptic arc chord)
- Centre point (P)
- Endpoints (X1, Y1) and (X2, Y2) (2P)
- T_start, T_end (2R)

Description: An elliptic arc chord is a boundary closed by the elliptic arc, defined by the parameters, and the chord from the starting to the ending point of the elliptic arc (see C.6.2.2.1.6.6).

The elliptic arc chord is displayed with interior style as defined by SET FILL AREA INDEX, SET ASPECT SOURCE FLAGS and SET FILL AREA INTERIOR STYLE.

Related primitives: Same as for FILL AREA.

Discussion: If the centre point and the endpoints coincide, a dot is displayed.

If the centre point and the endpoints are collinear, a straight line connecting the two endpoints is displayed.

If any two of the centre point and the two endpoints are coinciding, the primitive is ignored.

If T_start equals T_end, the full ellipse is displayed.

C.6.2.2.1.6.12 GDP (elliptic arc pie)

Level: 0a

Parameters:

- GDP identifier (I)
(= elliptic arc pie)
- Centre point (P)
- Endpoints (X1, Y1) and (X2, Y2) (2P)
- T_start, T_end (2R)

Description: An elliptic arc pie is a boundary closed by the elliptic arc, defined by the parameters and the lines from the starting and ending points of this elliptic arc to the centre point [see GDP (elliptic arc)].

The elliptic arc pie is displayed with an interior style as defined by SET FILL AREA INDEX, SET ASPECT SOURCE FLAGS and SET FILL AREA INTERIOR STYLE.

Related primitives: Same as for FILL AREA.

Discussion: If the centre point and the endpoints coincide, a dot is displayed.

If the centre point and the endpoints are collinear, a straight line connecting the two endpoints is displayed.

If any two of the centre point and the two endpoints are coinciding, the primitive is ignored.

If T_start equals T_end, the full ellipse is displayed.

C.6.2.2.1.6.13 GDP (spline)

Level: 0a

Parameters:

- GDP identifier (I)
(= spline)
- Point list (3..n) (nP)

Description: A smooth curve is drawn based on the specified points. This curve, known as a uniform quadratic B-spline, is displayed with the attributes as defined for the display of POLYLINE.

Related primitives: Same as for POLYLINE.

Discussion: None.

C.6.2.2.2 Output primitives related to display element attributes

C.6.2.2.2.1 SET POLYLINE REPRESENTATION

Level: 1a

Parameters:

- Workstation identifier (ID)
- Polyline index (IX)
- Line type (I)
- Line width scale factor (R)
- Polyline colour index (CI)

Description: In the polyline bundle table of the specified workstation, the given index is associated with the specified parameters:

- Line type;
- Line width scale factor;
- Polyline colour index.

The polyline bundle table in the workstation has predefined entries. Any table entry may be redefined.

Which of the aspects in an entry that are used depends upon the setting of the correspond ASF's.

Related primitives: POLYLINE
 SET POLYLINE INDEX
 GDP (spline)
 GPD (circular arc 3 point)
 GDP (circular arc centre)
 GDP (elliptic arc)

Discussion: If one or more parameters are invalid or out of range, the primitive is ignored.

C.6.2.2.2.2 SET POLYLINE INDEX

Level: 0a

Parameters:

- Polyline index (IX)

Description: The polyline index is set to the value specified by the parameter.

Related primitives: SET ASPECT SOURCE FLAGS

Discussion: If the parameter value is out of range, the default value is set.

C.6.2.2.2.3 SET LINE TYPE

Level: 0a

Parameters:

- Line type (I)
 (One of: SOLID, DASHED, DOTTED, DASHED-DOTTED, <other, implementation dependent>)

Description: The line type is set to the value specified by the parameter.

When the 'line type ASF' is 'INDIVIDUAL', subsequent display elements using POLYLINE attributes are displayed with this line type.

When the 'line type ASF' is 'BUNDLED', this primitive does not affect the display of subsequent display elements using POLYLINE attributes until the ASF returns to 'INDIVIDUAL'.

Related primitives: POLYLINE
GDP (circular arc 3 point)
GDP (circular arc centre)
GDP (elliptic arc)
GDP (spline)

Discussion: If the parameter value is out of range, the default value is set.

C.6.2.2.2.4 SET LINE WIDTH SCALE FACTOR

Level: 0a

Parameters: – Line width scale factor (R)

Description: The line width scale factor is set to the value specified by the parameter.

When the 'line width scale factor ASF' is 'INDIVIDUAL', subsequent display elements using POLYLINE attributes are displayed with this line width scale factor.

When the 'line width scale factor ASF' is 'BUNDLED', this primitive does not affect the display of subsequent display elements using POLYLINE attributes until the ASF returns to 'INDIVIDUAL'.

Related primitives: POLYLINE
GDP (circular arc 3 point)
GDP (circular arc centre)
GDP (elliptic arc)
GDP (spline)

Discussion: If the line width scale factor parameter has the value 0.0, the default line width scale factor is set.

If the line width scale factor parameter is negative, the primitive is ignored.

C.6.2.2.2.5 SET POLYLINE COLOUR INDEX

Level: 0a

Parameters: – Polyline colour index (CI)

Description: The polyline colour index is set to the value specified by the parameter.

When the 'polyline colour ASF' is 'INDIVIDUAL', subsequent display elements using POLYLINE attributes are displayed using this polyline colour index.

When the 'polyline colour ASF' is 'BUNDLED', this primitive does not affect the display of subsequent display elements using POLYLINE attributes until the ASF returns to 'INDIVIDUAL'.

Related primitives: POLYLINE
GDP (circular arc 3 point)
GDP (circular arc centre)
GDP (spline)
GDP (elliptic arc)

Discussion: If the parameter value is out of range, the default value is set.

C.6.2.2.2.6 SET POLYMARKER REPRESENTATION

Level: 1a

Parameters: – Workstation identifier (ID)
– Polymarker index (IX)
– Marker type (I)
– Marker size scale factor (R)
– Polymarker colour index (CI)

Description: In the polymarker bundle table of the specified workstation, the given polymarker index is associated with the specified parameters:

- Marker type;
- Marker size scale factor;
- Polymarker colour index.

The polymarker bundle table in the workstation has predefined entries. Any table entry (including the predefined entries) may be redefined.

Which of the aspects in the entry are used depends upon the setting of the corresponding ASF's.

Related primitives: POLYMARKER
SET POLYMARKER INDEX

Discussion: If one or more parameters are invalid or out of range, the primitive is ignored.

C.6.2.2.2.7 SET POLYMARKER INDEX

Level: 0a

Parameters: – Polymarker index (IX)

Description: The polymarker index is set to the value specified by the parameter.

Related primitives: SET ASPECT SOURCE FLAGS

Discussion: If the parameter value is out of range, the default value is set.

C.6.2.2.2.8 SET MARKER TYPE

Level: 0a

Parameters: – Marker type (I)
(One of: DOT, PLUS, ASTERISK, CIRCLE, DIAGONAL CROSS, <other, implementation dependent>)

Description: The marker type is set to the value specified by the parameter.

When the 'marker type ASF' is 'INDIVIDUAL', subsequent POLYMARKER elements are displayed with this marker type.

When the 'marker type ASF' is 'BUNDLED', this primitive does not affect the display of subsequent POLYMARKER elements until the ASF returns to 'INDIVIDUAL'.

The marker types produce centred symbols as indicated:

- DOT
- + PLUS
- * ASTERISK
- CIRCLE
- X DIAGONAL CROSS

Related primitives: POLYMARKER

Discussion: If the parameter value is out of range, the default value is set.

C.6.2.2.2.9 SET MARKER SIZE SCALE FACTOR

Level: 0a

Parameters: – Marker size scale for (R)

Description: The marker size scale factor is set to the value specified by the parameter.

When the 'marker size scale factor ASF' is 'INDIVIDUAL', subsequent POLYMARKER elements are displayed with this marker size scale factor.

When the 'marker size scale factor ASF' is 'BUNDLED', this primitive does not affect the display of subsequent POLYMARKER elements until the ASF returns to 'INDIVIDUAL'.

Related primitives: POLYMARKER

Discussion: If the marker size scale factor parameter has the value 0.0, the default marker size scale factor is set.
If the marker size scale factor parameter is negative, the primitive is ignored.

C.6.2.2.2.10 SET POLYMARKER COLOUR INDEX

Level: 0a

Parameters: – Polymarker colour index (CI)

Description: The polymarker colour index is set to the value specified by the parameter.

When the 'polymarker colour ASF' is 'INDIVIDUAL', subsequent POLYMARKER elements are displayed with this polymarker colour index.

When the 'polymarker colour ASF' is 'BUNDLED', this primitive does not affect the display of subsequent POLYMARKER elements until the ASF returns to 'INDIVIDUAL'.

Related primitives: POLYMARKER

Discussion: If the parameter value is out of range, the default value is set.

C.6.2.2.2.11 SET FILL AREA REPRESENTATION

Level: 1a

Parameters: – Workstation identifier (ID)
– Fill area index (IX)
– Fill area interior style (E)
(One of: HOLLOW, SOLID, PATTERN, HATCH)
– Fill area style index (I)
(interior style = HATCH)
or
– Fill area style index (IX)
(interior style = PATTERN)
– Fill area colour index (CI)

Description: In the fill area bundle table of the specified workstation, the given fill area index is associated with the specified parameters:

- Fill area interior style;
- Fill area style index (hatch or pattern);
- Fill area colour index.

The fill area bundle table in the workstation has predefined entries. Any table entry (including the predefined entries) may be redefined.

Which of the aspects in the entry are used depends upon the setting of the corresponding ASF's.

Related primitives: FILL AREA
SET FILL AREA INDEX
GDP (rectangle)
GDP (circle)
GDP (circular arc 3 point chord)
GDP (circular arc centre chord)
GDP (circular arc 3 point pie)
GDP (circular arc centre pie)
GDP (ellipse)
GDP (elliptic arc chord)
GDP (elliptic arc pie)

Discussion: If one or more parameters are invalid or out of range, the primitive is ignored.

C.6.2.2.2.12 SET FILL AREA INDEX

Level: 0a

Parameters: – Fill area index (IX)

Description: The fill area index is set to the value specified by the parameter.

Related primitives: SET ASPECT SOURCE FLAGS

Discussion: If the parameter value is out of range, the default value is set.

C.6.2.2.2.13 SET FILL AREA INTERIOR STYLE

Level: 0a

Parameters: – Fill area interior style (E)
(One of: HOLLOW, SOLID, PATTERN, HATCH)

Description: The interior style of the display elements which define closed boundaries is set to the value specified by the parameter.

When the 'fill area interior style ASF' is 'INDIVIDUAL', subsequent elements are displayed with this interior style.

When the 'fill area interior style ASF' is 'BUNDLED', this element does not affect the display of these subsequent display elements until the ASF returns to 'INDIVIDUAL'.

The fill area interior style is used to determine in what style the area is to be filled:

- HOLLOW: No filling. The boundary is drawn using the fill area colour index currently selected (either BUNDLED or INDIVIDUALLY, depending on the corresponding 'fill area colour ASF').
- SOLID: Fill the interior using the fill area colour index currently selected (either BUNDLED or INDIVIDUALLY, depending on the corresponding 'fill area colour ASF').
- PATTERN: Fill the interior using the fill area style index currently selected (either BUNDLED or INDIVIDUALLY, depending on the corresponding 'fill area style index ASF').
- HATCH: Fill the interior using the fill area colour index and the fill area style index currently selected (either BUNDLED or INDIVIDUALLY, depending on the corresponding 'fill area colour ASF' and 'fill area style index ASF').

Related primitives: FILL AREA
GDP (rectangle)
GDP (circle)
GDP (circular arc 3 point chord)
GDP (circular arc centre chord)
GDP (circular arc 3 point pie)
GDP (circular arc centre pie)
GDP (ellipse)
GDP (elliptic arc chord)
GDP (elliptic arc pie)
SET FILL AREA STYLE INDEX

Discussion: If the parameter value is out of range, the default value is set.

C.6.2.2.2.14 SET FILL AREA COLOUR INDEX

Level: 0a

Parameters: – Fill area colour index (CI)

Description: The fill area colour index is set to the value specified by the parameter.

When the 'fill area colour ASF' is 'INDIVIDUAL', subsequent display elements which define closed boundaries are filled using this colour index.

When the 'fill area colour ASF' is 'BUNDLED', this primitive does not affect the display of subsequent display elements which define closed boundaries until the ASF returns to 'INDIVIDUAL'.

The fill area colour index is only significant if the Fill area interior style is either 'SOLID', 'HATCH' or 'HOLLOW'.

Related primitives: FILL AREA
 GDP (rectangle)
 GDP (circle)
 GDP (circular arc 3 point chord)
 GDP (circular arc centre chord)
 GDP (circular arc 3 point pie)
 GDP (circular arc centre pie)
 GDP (ellipse)
 GDP (elliptic arc chord)
 GDP (elliptic arc pie)

Discussion: If the parameter value is out of range, the default value is set.

C.6.2.2.2.15 SET FILL AREA STYLE INDEX

Level: 0a

Parameters: – Fill area style index (I)
 if (interior style = HATCH)
or
 – Fill area style index (IX)
 if (interior style = PATTERN)

Description: The fill area style index is set to the value specified by the parameter.

When the 'fill area style index ASF' is 'INDIVIDUAL', subsequent display elements which define closed boundaries are displayed using this fill area style index.

When the 'fill area style index ASF' is 'BUNDLED', this primitive does not affect the display of subsequent display elements which define closed boundaries until the ASF returns to 'INDIVIDUAL'.

Related primitive: FILL AREA
 GDP (rectangle)
 GDP (circle)
 GDP (circular arc 3 point chord)
 GDP (circular arc centre chord)
 GDP (circular arc 3 point pie)
 GDP (circular arc centre pie)
 GDP (ellipse)
 GDP (elliptic arc chord)
 GDP (elliptic arc pie)
 SET FILL AREA INTERIOR STYLE

Discussion: If the parameter value is out of range, the default value is set.

C.6.2.2.2.16 SET PATTERN REPRESENTATION

Level: 1a

Parameters: – Workstation identifier (ID)
 – Pattern index (IX)
 – Deltax (DX) (I)
 – Deltay (DY) (I)
 – Pattern cell colour index list (CL)

Description:	The deltax and deltax values define a horizontal by vertical (DX by DY) array into which colour values are mapped.
Related primitives:	FILL AREA GDP (rectangle) GDP (circle) GDP (circular arc 3 point chord) GDP (circular arc centre chord) GDP (circular arc 3 point pie) GDP (circular arc centre pie) GDP (ellipse) GDP (elliptic arc chord) GDP (elliptic arc pie) SET FILL AREA INTERIOR STYLE
Discussion:	If one or more parameters are invalid or out of range, the primitive is ignored. If the number of indices in the colour index list is less than $DX * DY$, the last index is repeated until the end of the list. If the number of indices in the colour index list is greater than $DX * DY$, the exceeding indices are omitted. If deltax or deltax are less than or equal to 0, the primitive is ignored.

C.6.2.2.2.17 SET PATTERN REFERENCE POINT

Level	0a
Parameters:	– Reference point (P)
Description:	The pattern reference point is set to the value specified by the parameter. When the currently selected interior style is PATTERN, this value is used in conjunction with the pattern vectors for displaying filled areas. When the currently selected interior style is HATCH, it is implementation dependent if the pattern reference point is used for displaying filled areas.
Related primitives:	FILL AREA GDP (rectangle) GDP (circle) GDP (circular arc 3 point chord) GDP (circular arc centre chord) GDP (circular arc 3 point pie) GDP (circular arc centre pie) GDP (ellipse) GDP (elliptic arc chord) GDP (elliptic arc pie) SET FILL AREA INTERIOR STYLE SET PATTERN REPRESENTATION SET PATTERN VECTORS
Discussion:	None.

C.6.2.2.2.18 SET PATTERN VECTORS

Level:	0a
Parameters:	– Height vector (P) – Width vector (P)
Description:	The origin of the NDC space and the first point define the pattern height vector. The origin of the NDC space and the second point define the pattern width vector. When the currently selected interior style is PATTERN, these pattern vectors are used in conjunction with the pattern reference point for displaying filled areas.

Related primitives: FILL AREA
 GDP (rectangle)
 GDP (circle)
 GDP (circular arc 3 point chord)
 GDP (circular arc centre chord)
 GDP (circular arc 3 point pie)
 GDP (circular arc centre pie)
 GDP (ellipse)
 GDP (elliptic arc chord)
 GDP (elliptic arc pie)
 SET FILL AREA INTERIOR STYLE
 SET PATTERN REPRESENTATION
 SET PATTERN REFERENCE POINT

Discussion: If the two vectors are coinciding or at 180 degrees, the primitive is ignored.

C.6.2.2.2.19 SET TEXT REPRESENTATION

Level: 1a

Parameters: – Workstation identifier (ID)
 – Text index (IX)
 – Text font (I)
 – Text precision (E)
 – Character expansion factor (R)
 – Character spacing (R)
 – Text colour index (CI)

Description: In the text bundle table of the specified workstation, the given text index is associated with the specified parameters:

- Text font and precision;
- Character expansion factor;
- Character spacing;
- Text colour index.

The text bundle table in the workstation has predefined entries. Any table entry (including the predefined entries) may be redefined.

Which of the aspects in the entry are used depends upon the setting of the corresponding ASF's.

Related primitives: TEXT
 SET TEXT INDEX

Discussion: If one or more parameters are invalid or out of range, the primitive is ignored.

C.6.2.2.2.20 SET TEXT INDEX

Level: 0a

Parameters: – Text index (IX)

Description: The text index is set to the value specified by the parameter.

Related primitives: SET ASPECT SOURCE FLAGS

Discussion: If the parameter value is out of range, the default value is set.

C.6.2.2.2.21 SET TEXT FONT AND PRECISION

Level: 0a

Parameters: – Text font (I)
 – Text precision (E)
 (One of: STRING, CHARACTER, STROKE)

Description: The text font and precision are set to the values specified by the parameters.

When the 'Text font and precision ASF' is 'INDIVIDUAL', subsequent TEXT display elements are displayed with this text precision and using this text font.

When the 'Text font and precision ASF' is 'BUNDLED', this primitive does not affect the display of subsequent TEXT elements until the ASF returns to 'INDIVIDUAL'.

The accuracy of execution of text attributes can be controlled by one of three values for text precision.

If 'STRING' precision is specified, the text string is generated in the requested text font and is positioned by aligning the text string at the given text position. The length of the character vectors and the character expansion factor are evaluated as closely as reasonable, given the capabilities of the workstation. The direction of the character vectors, the text path, the text alignment and the character spacing need not be used. Clipping is done in an implementation dependent way.

If 'CHARACTER' precision is specified, the text string is generated in the requested text font. For the representation of each individual character, both the length and the direction of the character vectors and the character expansion factor are evaluated as closely as possible, in a workstation dependent way. The spacing used between character bodies is evaluated exactly. The character body, for this purpose, is an ideal character body, calculated precisely from the text aspects and the font dimensions. The position of the text string is determined by the text alignment and the text position. Clipping is performed at least on a character by character basis.

If 'STROKE' precision is specified, the text string is displayed at the text position by applying all text aspects. The text string is clipped exactly at the clipping rectangle.

Related primitives: TEXT

Discussion: If a parameter value is out of range, the default value of that parameter is set.

C.6.2.2.2.22 SET CHARACTER EXPANSION FACTOR

Level: 0a

Parameters – Character expansion factor (R)

Description: The character expansion factor is set to the value specified by the parameter.

When the 'Character expansion factor ASF' is 'INDIVIDUAL', subsequent TEXT display elements are displayed with this character expansion factor.

When the 'Character expansion factor ASF' is 'BUNDLED', this primitive does not affect the display of subsequent TEXT display elements until the ASF returns to 'INDIVIDUAL'.

The Character expansion factor specifies the deviation of the width/height ratio of the characters from the ratio indicated by the font designer.

The Character expansion factor is a non-dimensional scalar. The desired resulting character width is the product of the length of the character width vector times the width/height ratio for the character times the character expansion factor.

Related primitives: TEXT
SET CHARACTER VECTORS

Discussion: If the character expansion factor has a value less than or equal to 0, the default character expansion factor will be set.

C.6.2.2.2.23 SET CHARACTER SPACING

Level:	0a
Parameters:	– Character spacing (R)
Description:	The character spacing is set to the value specified by the parameter. When the 'Character spacing ASF' is 'INDIVIDUAL', subsequent TEXT display elements are displayed with this character spacing. When the 'Character spacing ASF' is 'BUNDLED', this primitive does not affect the display of subsequent TEXT display elements until the ASF returns to 'INDIVIDUAL'. The parameter represents the desired space to be added between characters of a text string. It is specified as a fraction of the current character height vector attribute. The space is added along the text path. A negative value implies that characters may overlap.
Related primitives:	TEXT SET CHARACTER VECTORS
Discussion:	The parameter is compared to the dimensions of available intercharacter space in the device. The available value next smaller than or equal to the specified value is selected. If no such value is available, the next larger value may be selected.

C.6.2.2.2.24 SET TEXT COLOUR INDEX

Level	0a
Parameters	– Text colour index (CI)
Description:	The text colour index is set as specified by the parameter. When the 'Text colour ASF' is 'INDIVIDUAL', subsequent TEXT display elements are displayed using this text colour index. When the 'Text colour ASF' is 'BUNDLED', this primitive does not affect the display of subsequent TEXT display elements until the ASF returns to 'INDIVIDUAL'.
Related primitives:	TEXT
Discussion:	If the parameter value is out of range, the default value is set.

C.6.2.2.2.25 SET TEXT PATH

Level:	0a
Parameters:	– Text path (E) (One of: RIGHT, LEFT, UP, DOWN)
Description:	The text path is set to the value specified by the parameter. Subsequent TEXT display elements are displayed with this text path. This primitive sets the value of the text path attribute, specifying the writing direction of a text string relative to the character height vector and the character width vector. 'RIGHT' means in the direction of the character width vector. 'LEFT' means 180 degrees from the character width vector. 'UP' means in the direction of the character height vector. 'DOWN' means 180 degrees from the character height vector.
Related primitives:	TEXT SET CHARACTER VECTORS
Discussion:	If the parameter value is out of range, the default value is set.

C.6.2.2.26 SET CHARACTER VECTORS

Level: 0a

Parameters: – Height vector (P)
– Width vector (P)

Description: The origin of the NDC space and the first point define the character height vector. The origin of the NDC space and the second point define the character width vector.

The direction of the two vectors defines both the orientation and the skew of the character body and the sense of the text path in subsequent TEXT display elements.

The length of the character height vector defines the height of the character within the character body and is also used as a scaling factor when calculating the desired space to be added between the characters. The length of the width vector is used as a scaling factor when calculating the width of the character within the character body.

The character expansion factor is a non-dimensional scalar. The desired resulting character width is the product of the length of the character width vector times the width/height ratio for the character times the character expansion factor.

Related primitives: TEXT
SET TEXT PATH
SET CHARACTER SPACING
SET CHARACTER EXPANSION FACTOR
SET TEXT ALIGNMENT

Discussion: If the two vectors are coincident or at 180 degrees, the primitive is discarded.

When the primitive is processed, it is transformed by whatever transformation the workstation is using to map NDC coordinates to device coordinates and the resulting value is compared to the set of available character heights in the device. The available value next smaller than or equal to the transformed value is selected. If no such value is available, the next larger value may be selected.

C.6.2.2.27 SET TEXT ALIGNMENT

Level: 0a

Parameters: – Horizontal alignment (E)
(One of: NORMAL, LEFT, CENTRE, RIGHT)
– Vertical alignment (E)
(One of: NORMAL, TOP, CAP, HALF, BASE, BOTTOM)

Description: The text alignment is set to the value specified by the parameter. Subsequent display elements using the TEXT attributes are displayed with this text alignment.

Related primitives: TEXT
SET CHARACTER VECTORS

Discussion: The 'NORMAL' parameter values are dependent on the text path at the time of the elaboration of the TEXT display element.

<i>PATH</i>	<i>NORMAL HORIZONTAL</i>	<i>NORMAL VERTICAL</i>
RIGHT	LEFT	BASE
LEFT	RIGHT	BASE
UP	CENTRE	BASE
DOWN	CENTRE	TOP

If one of the parameter values is out of range, the default value is set.

C.6.2.2.2.28 SET COLOUR REPRESENTATION

Level: 0a

Parameters:

- Workstation identifier (ID)
- Starting entry in the colour table (CI)
- Colour data (nCD)

Description: This primitive is used to load colour tables. The first parameter specifies the workstation. The second parameter defines the first table entry to be loaded.

The number of bits used to define an intensity is defined by the 'unit resolution' parameter of the last SET COLOUR HEADER primitive processed.

The end of the colour data parameter (and therefore the number of entries to be loaded n) is implicitly indicated by the receipt of the next primitive to be processed.

Related primitives: SET COLOUR HEADER

Discussion: If the first or second parameters are invalid or out of range, the primitive is ignored. If the last colour data are incomplete these last, truncated colour data are ignored.

C.6.2.2.2.29 SET ASPECT SOURCE FLAGS

Level: 0a

Parameters:

- Line type ASF (E)
(One of: INDIVIDUAL, BUNDLED)
- Line width scale factor ASF (E)
(One of: INDIVIDUAL, BUNDLED)
- Polyline colour ASF (E)
(One of: INDIVIDUAL, BUNDLED)
- Marker type ASF (E)
(One of: INDIVIDUAL, BUNDLED)
- Marker size scale factor ASF (E)
(One of: INDIVIDUAL, BUNDLED)
- Polymarker colour ASF (E)
(One of: INDIVIDUAL, BUNDLED)
- Text font and precision ASF (E)
(One of: INDIVIDUAL, BUNDLED)
- Character expansion factor ASF (E)
(One of: INDIVIDUAL, BUNDLED)
- Character spacing ASF (E)
(One of: INDIVIDUAL, BUNDLED)
- Text colour ASF (E)
(One of: INDIVIDUAL, BUNDLED)
- Fill area interior style ASF (E)
(One of: INDIVIDUAL, BUNDLED)
- Fill area style index ASF (E)
(One of: INDIVIDUAL, BUNDLED)
- Fill area colour ASF (E)
(One of: INDIVIDUAL, BUNDLED)

Description: The Aspect Source Flags (ASFs) are set to the values specified by the parameters.

Related primitives: None.

Discussion: If one of the parameter values is out of range, the default value is set.

C.6.2.2.3 Transformation primitives

C.6.2.2.3.1 SET WORKSTATION WINDOW

Level: 0a

Parameters:

- Workstation identifier (ID)
- Window limits (2P)

Description: The parameters specify a rectangle in the unit square of the NDC space used for the display of images.

The 'requested workstation window' entry in the workstation state list of the specified workstation is set to the value specified by the parameter.

If the 'dynamic modification accepted' entry for workstation transformation in the workstation description table is set to IMM or if the 'display surface empty' entry in the workstation state list is set to EMPTY, then the 'current workstation window' entry in the workstation state list is set to the value specified by the parameter and the 'workstation transformation update state' entry is set to NOTPENDING. Otherwise, the 'workstation transformation update state' entry in the workstation state list is set to PENDING and the 'current workstation window' entry is not changed.

Related primitives: SET WORKSTATION VIEWPORT

Discussion: If the points are outside the unit square, the default workstation window is set.

If the workstation identifier parameter is out of range, the primitive is ignored.

C.6.2.2.3.2 SET WORKSTATION VIEWPORT

Level: 0a

Parameters:

- Workstation identifier (ID)
- Viewport limits: $XMIN < XMAX$, $YMIN < YMAX$ (4R)

Description: The parameters specify a rectangle in the DC space.

The 'requested workstation viewport' entry in the workstation state list of the specified workstation is set to the value specified by the parameter.

If the 'dynamic modification accepted' entry for workstation transformation in the workstation description table is set to IMM or if the 'display surface empty' entry in the workstation state list is set to EMPTY, then the 'current workstation viewport' entry in the workstation state list is set to the value specified by the parameter and the 'workstation transformation update state' entry is set to NOTPENDING. Otherwise, the 'workstation transformation update state' entry in the workstation state list is set to PENDING and the 'current workstation viewport' entry is not changed.

Related primitives: SET WORKSTATION WINDOW

Discussion: If the points are outside the display space, the default workstation viewport is set.

If the workstation identifier parameter is out of range, the primitive is ignored.

C.6.2.2.4 Clipping primitives

C.6.2.2.4.1 SET CLIPPING RECTANGLE

Level: 0a

Parameters: – Clipping rectangle limits (2P)

Description: The parameters specify a rectangle in the unit square of the NDC space, which defines the clipping rectangle.

Related primitives: None

Discussion: This primitive is necessary to provide local clipping areas within the NDC space. For example, high quality text could be sent to the device as text string and clipping rectangle so that the interpreter can clip character strokes.

If the points are outside the unit square, the default clipping rectangle is set.

C.6.2.2.5 Control primitives

C.6.2.2.5.1 UPDATE WORKSTATION

Level: 0a

Parameters: – Workstation identifier (ID)
– Update regeneration flag (E)
(One of: PERFORM, POSTPONE)

Description: All deferred actions for the specified workstation are executed (without intermediate clearing of the display surface). If the update regeneration flag is set to PERFORM and the 'new frame action necessary at update' entry in the workstation state list is YES, then the following actions are executed in the given sequence:

- a) The display surface is cleared only if the 'display surface empty' entry in the workstation state list is NOTEMPTY. The entry is set to EMPTY.
- b) If the 'workstation transformation update state' entry in the workstation state list is PENDING, the 'current workstation window' and 'current workstation viewport' entries in the workstation state list are assigned the values of the 'requested workstation window' and 'requested workstation viewport' entries. The 'workstation transformation update state' entry is set to NOTPENDING.
- c) All visible segments stored on this workstation (i.e. contained in the 'set of stored segments for this workstation' entry of the workstation state list) are redisplayed. The action typically causes the 'display surface empty' entry in the workstation state list to be set to NOTEMPTY.
- d) The 'new frame action necessary at update' entry of the workstation state list is set to NO.

If the update regeneration flag is PERFORM, UPDATE WORKSTATION suspends the effect of SET DEFERRAL STATE. In that case, it is equivalent to the following sequence of primitives:

- INQUIRE WORKSTATION STATE;
save deferral state;
- SET DEFERRAL STATE (ASAP, ALLOWED);
set deferral state to saved value.

If the value of the 'new frame action necessary at update' entry is NO or the update regeneration flag is POSTPONE, UPDATE WORKSTATION merely initiates the transmission of blocked data. If the value of the 'new frame necessary at update' entry is YES and the update regeneration flag is PERFORM, UPDATE WORKSTATION behaves as REDRAW ALL SEGMENTS ON WORKSTATION.

The 'new frame action necessary at update' entry in the workstation state list is set to YES during deferred action generation if both of the following are true (see C.5.5):

- a) an action causing modification of the picture is actually deferred on that workstation;
- b) the workstation display surface does not allow modification of the image without redrawing the whole picture (for example: plotter storage tube display).

Related primitives: SET DEFERRAL STATE

Discussion: If the workstation identifier parameter is out of range, the primitive is ignored.

C.6.2.2.5.2 SET DEFERRAL STATE

Level: 1a

Parameters:

- Workstation identifier (ID)
- Deferral mode (E)
(One of: ASAP, BNIL, BNIG, ASTI)
- Implicit regeneration flag (E)
(One of: SUPPRESSED, ALLOWED)

Description: The deferral mode and implicit regeneration flag of the specified workstation are set to the values specified by the parameters.

Depending on the new value of deferral mode, deferred output may be unblocked. If the new value of implicit regeneration flag is ALLOWED and the 'new frame action necessary at update' entry in the workstation state list is YES, an action equivalent to REDRAW ALL SEGMENTS ON WORKSTATION is performed.

Related primitives: UPDATE WORKSTATION

Discussion: If the workstation identifier parameter is out of range, the primitive is ignored.

C.6.2.2.5.3 EMERGENCY CLOSE

Level: 0a

Parameters:

- None.

Description: The graphics configuration is emergency closed. The following actions are performed (if possible):

- a) CLOSE SEGMENT (if open);
- b) UPDATE WORKSTATION with update regeneration flag PERFORM for all open workstation;

- c) DEACTIVATE WORKSTATION for all active workstations;
- d) CLOSE WORKSTATION for all open workstations.

Related primitives: None.

Discussion: None.

C.6.2.3 Segment related primitives

C.6.2.3.1 WDSS related primitives

C.6.2.3.1.1 CREATE SEGMENT

Level: 1a

Parameters: – Segment name (ID)

Description: A segment is opened with the name <segment name>. All subsequent output primitives belong to this segment until the first CLOSE SEGMENT primitive.

Related primitives: CLOSE SEGMENT

Discussion: None.

C.6.2.3.1.2 CLOSE SEGMENT

Level: 1a

Parameters: – None.

Description: No further primitives will be added to the current open segment. If no segment was open, this primitive will have no effect.

Related primitives: CREATE SEGMENT

Discussion: None.

C.6.2.3.1.3 RENAME SEGMENT

Level: 1a

Parameters: – Old segment name (ID)
– New segment name (ID)

Description: Each occurrence of old segment name is replaced by new segment name.

If old segment name is the name of the open segment, the name of the open segment is set to the new segment name.

Related primitives: None.

Discussion: If the old segment name doesn't exist in a workstation, the primitive has no effect on that workstation.

C.6.2.3.1.4 DELETE SEGMENT FROM WORKSTATION

Level: 1a

Parameters: – Workstation identifier (ID)
– Segment name (ID)

Description: The named segment is deleted from the specified workstation and the segment name is removed from the set of segment names in use for this workstation.

Related primitives: None.

Discussion: If the named segment doesn't exist in the specified workstation, the primitive has no effect.
If the workstation identifier parameter is out of range, the primitive is ignored.

C.6.2.3.1.5 DELETE SEGMENT

Level: 1a

Parameters: – Segment name (ID)

Description: The named segment and the segment name are deleted from all workstation segment storages.

Related primitives: None.

Discussion: If the named segment doesn't exist in a workstation, the primitive has no effect on that workstation.

C.6.2.3.1.6 REDRAW ALL SEGMENTS ON WORKSTATION

Level: 1a

Parameters: – Workstation identifier (ID)

Description: The following actions are executed in the given sequence:

- a) All deferred actions for the specified workstation are executed (without intermediate clearing of the display surface).
- b) The display surface is cleared if the 'display surface empty' entry in the workstation state list is NOTEMPTY. The entry is set to EMPTY.
- c) The current workstation transformation is set to the last defined workstation window and workstation viewport, if the 'workstation transformation update state' entry in the workstation state list is PENDING. The entry is set to NOTPENDING.
- d) All visible segments stored on this workstation are redisplayed.

Related primitives: None

Discussion: If the workstation identifier parameter is out of range, the primitive is ignored.

C.6.2.3.1.7 SET HIGHLIGHTING

Level: 1a

Parameters: – Segment name (ID)
– Highlighting flag (E)
(One of: NOTHIGHLIGHTED, HIGHLIGHTED)

Description: The highlighting attribute of the named segment is set to the value specified by the parameter.

Related primitives: None.

Discussion: None.

C.6.2.3.1.8 SET VISIBILITY

Level: 1a

Parameters: – Segment name (ID)
– Visibility flag (E)
(One of: VISIBLE, INVISIBLE)

Description: The visibility attribute of the named segment is set to the value specified by the parameter.

Related primitives: None.

Discussion: None.

C.6.2.3.1.9 SET SEGMENT TRANSFORMATION

Level: 1a

Parameters: – Segment name (ID)
– Transformation matrix (M)

Description: The segment transformaion matrix is set to the value specified by the parameter. When a segment is displayed, the coordinates of its display elements are transformed by applying the following matrix multiplication to them:

$$\begin{array}{c} \begin{array}{|c|} \hline x' \\ \hline \\ \hline y' \\ \hline \end{array} = \begin{array}{|ccc|} \hline M11 & M12 & M13 \\ \hline M21 & M22 & M23 \\ \hline \end{array} \times \begin{array}{|c|} \hline x \\ \hline y \\ \hline 1 \\ \hline \end{array} \end{array}$$

The original coordinates are (x, y), the transformed coordinates are (x', y'), both in NDC.

The values M13 and M23 of the transformation matrix are in NDC coordinates, the other values are unitless. For geometric attributes which are vectors (for example: CHARACTER VECTORS), the values M13 and M23 are ignored.

The function can be used to transform a stored segment.

The segment transformation (conceptually takes place in NDC space. This transformation is not cumulative, i.e. it always applies to the segment as originally created.

Related primitives: INSERT SEGMENT
REDRAW ALL SEGMENTS ON WORKSTATION

Discussion: None.

C.6.2.3.1.10 SET SEGMENT PRIORITY

Level: 1a

Parameters: – Segment name (ID)
– Segment priority (R)

Description: The 'segment priority' entry in the segment state list of the named segment is set to the value specified by the parameter. Segment priority affects the display of segments and pick input if segments overlap, in which case precedence is given to segments with higher priority. If segments with the same priority overlap, the result is implementation dependent.

The use of segment priority applies only to workstations where the entry 'number of segment priorities supported' in the workstation description table is greater than 1 or equal to 0 (indicating that a continuous range of priorities is supported).

If 'number of segment priorities supported' is greater than 1, the range [0.1] for segment priority is mapped onto the range 1 to 'number of segment priorities supported' for a specific workstation before being used. If 'number of segment priorities supported' is equal to 0, the implementation allows all values of segment priority to be differentiated.

This feature is intended to address appropriate hardware capabilities only. It cannot be used to force software checking of interference between segments on non-raster displays.

Related primitives: REDRAW ALL SEGMENTS ON WORKSTATION

Discussion: None.

C.6.2.3.2 WISS related primitives

C.6.2.3.2.1 ASSOCIATE SEGMENT WITH WORKSTATION

Level: 2a

Parameters: – Workstation identifier (ID)
– Segment name (ID)

Description: The segment is sent to the specified workstation in the same way as if the workstation were active when the segment was created. Clipping rectangles are copied unchanged.

Related primitives: None.

Discussion: If the segment is not present in the WISS or if it is already associated with the specified workstation, the primitive has no effect.

C.6.2.3.2.2 COPY SEGMENT TO WORKSTATION

Level: 2a

Parameters: – Workstation identifier (ID)
– Segment name (ID)

Description: The primitives in the specified segment are sent to the indicated workstation after segment transformation and clipping at the clipping rectangle stored with each primitive.

The primitives are not stored in a segment.

All display elements keep the values of the display element attributes, that were assigned to them when they were created, for their whole lifetime.

In particular, when segments are copied, the values of the primitive attributes within the copied segments are unchanged.

Related primitives: None.

Discussion: If the segment is not present in the WISS or if the specified workstation is the WISS, this primitive is ignored.

C.6.2.3.2.3 INSERT SEGMENT

Level: 2a

Parameters: – Segment name (ID)
– Transformation matrix (M)

Description: Having been transformed, as described below, the primitives contained in the segment are drawn on the display surface and, eventually, stored in the open segment (see also C.5.4.6).

The coordinates of the primitives contained in the inserted segment are transformed first by any segment transformation specified for it, and secondly by applying the following matrix multiplication to them:

$$\begin{array}{c} \begin{array}{|c|} \hline x' \\ \hline \\ \hline y' \\ \hline \end{array} = \begin{array}{|ccc|} \hline M11 & M12 & M13 \\ \hline M21 & M22 & M23 \\ \hline \end{array} \times \begin{array}{|c|} \hline x \\ \hline y \\ \hline 1 \\ \hline \end{array} \end{array}$$

The original coordinates are (x, y), the transformed coordinates are (x', y'), both in NDC.

The values M13 and M23 of the transformation matrix are in NDC coordinates, the other values are unitless. For geometric attributes which are vectors (for example: CHARACTER VECTORS), the values M13 and M23 are ignored.

Other than the segment transformation, attributes of the inserted segment are ignored.

All clipping rectangles in the inserted segment are ignored, each primitive processed is assigned a new clipping rectangle which is the current clipping rectangle.

All primitives processed by a single invocation of INSERT SEGMENT receive the same clipping rectangle. All primitives keep the values of the primitive attributes, that were assigned to them when they were created.

In particular, where segments are inserted, the values of the primitive attributes within the inserted segments are unchanged. The values of primitive attributes used in the creation of subsequent primitives within the segment into which the insertion takes place are unaffected by the insertion.

Related primitives: SET SEGMENT TRANSFORMATION

Discussion: None.

C.6.2.4 Input primitives

A Videotex environment will contain no workstations of the type INPUT, so this subclause is not applicable for Videotex purposes.

C.6.2.5 Inquire primitives

A Videotex environment will support no inquire primitives, so this subclause is not applicable for Videotex purposes.

C.6.2.6 Protocol descriptor primitives

C.6.2.6.1 SET DOMAIN RING

Level: 0a

Parameters: – Angular resolution factor (E)
(One of: RESOLUTION_0, RESOLUTION_1, RESOLUTION_2, RESOLUTION_3)
– Basic Radius (I)

Description: This primitive specifies the precision of the coordinate encoding, when using incremental mode.

Related primitives: SET COORDINATE PRECISION

Discussion: If the basic Radius parameter value is 0, the default basic Radius according to the current granularity code (as set with the SET COORDINATE PRECISION primitive) is selected (see clause C.9).

If the basic Radius parameter is negative, the primitive is ignored.

If the angular resolution factor parameter value is out of range, the default value is set.

C.6.2.6.2 SET COLOUR HEADER

Level: 0a

Parameters: – Unit resolution (1, 2 .. 9) (I)
– Coding method (E)
(Only: RGB_CODING)

Description: The parameters have the following meaning:

- the first parameter specifies the number of bits used to encode the intensity of each colour component in the RGB colour definition.
- the last parameter specifies the coding method used.

Related primitives: SET COLOUR REPRESENTATION

Discussion: If the selected unit resolution is out of range, the default unit resolution is selected.

C.6.2.6.3 SET COORDINATE PRECISION

Level: 0a

Parameters:

- Magnitude code (I)
- Granularity code (I)
- Default exponent (I)
- Explicit exponent allowed (E)

(One of: ALLOWED, FORBIDDEN)

Description: The magnitude code parameter specifies the largest possible magnitude of positive or negative NDC coordinates and size values which can be encoded (the number of bits which may occur to the left of the binary radix point in the representation of a coordinate or size value as a binary fraction plus one bit for the sign bit).

The granularity code parameter specifies the smallest non-zero value that can be expressed with this precision. This value is called the Basic Grid Unit (BGU). It does this by specifying the smallest exponent that is permitted in the coded representation of a coordinate or a size value.

The default exponent parameter specifies the exponent value, which is used when encoding a point, in case:

- the exponent is omitted in the encoding, *and*
- the encoded point is a single point or the first point of a point list.

It also specifies the exponent value, which is used when encoding a size value, in case the exponent is omitted in the encoding.

The explicit exponent allowed parameter specifies whether or not an exponent part is allowed in the coded representation of a coordinate or a size value.

For example, if the magnitude code = +11 and the granularity code = –16, the coordinates in the range from –111111111.1111111111111111 (binary) to +111111111.1111111111111111 (binary) can be encoded with each encoded coordinate being a multiple of 2^{*-16} (binary 0.0000000000000001), the BGU.

Related primitives: SET DOMAIN RING

Discussion: The magnitude code must be greater than the granularity code. The magnitude code must be greater than 0. The default exponent must be greater than or equal to the granularity code. If either of these conditions are not met the primitive is ignored.

This primitive applies only to coordinates and size values.

C.6.2.6.4 SET REAL PRECISION

Level: 0a

Parameters:

- Magnitude code (I)
- Granularity code (I)
- Default exponent (I)
- Explicit exponent allowed (E)

(One of: ALLOWED, FORBIDDEN)

Description: The magnitude code parameter specifies the largest possible magnitude of positive or negative real numbers which can be encoded (the number of bits may occur to the left of the binary radix point in the representation of a real number as a binary fraction plus one bit for the sign bit).

The granularity code parameter specifies the smallest non-zero real number that can be expressed with this precision. It does this by specifying the smallest exponent that is permitted in the coded representation of a real number.

The default exponent parameter specifies the exponent, which is used when encoding a real number, in case the exponent is omitted in the encoding.

The explicit exponent allowed parameter specifies whether or not an exponent part is allowed in the coded representation of a real number.

For example, if the magnitude code = +11 and the granularity code = -16, the real numbers in the range from -111111111.1111111111111111 (binary) to +111111111.1111111111111111 (binary) can be encoded with real numbers encoded being a multiple of 2^{-16} (binary 0.0000000000000001).

Related primitives: None.

Discussion: The magnitude code must be greater than the granularity code. The magnitude code must be greater than 0. The default exponent must be greater than or equal to the granularity code. If either of these conditions are not met the primitive is ignored.

This primitive applies only to real numbers.

C.6.2.6.5 SET COLOUR INDEX PRECISION

Level: 0a

Parameters: – Number of bits (I)

Description: The number of bits parameter specifies the number of bits necessary to encode the colour indices.

Related primitives: None

Discussion: None.

C.7 Encoding principles

The encoding of the geometric display is independent of the encoding of other sets, e.g. alphamosaic or photographic display. The geometric display, in general, is selected by means of the appropriate US sequence, as described in part 0 of this annex. The specification of different levels within the geometric display is described in Annex IV.

The encoding of primitives is defined in terms of a 7-bit code. When used in an 8-bit environment, bit 8 of each octet shall be zero (except within the datatype 'string').

This clause deals with the detailed encoding principles of:

- The opcodes of the primitives;
- The operands of the primitives.

Each primitive is coded according to the following rules:

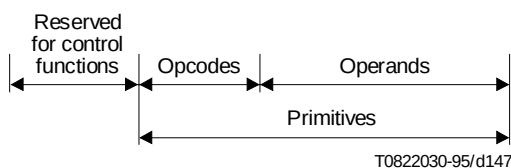
- a primitive is composed of one opcode and operands as required;
- the operands are encoded in column 2 or 3 of the 7-bit Code Table;
- operands are encoded in columns 4 up to 7. (However, the coded representation of a 'string' operand may include bit combinations from other columns of the Code Table – see the description of string operands in C.7.2.5.)

The start of a character string is indicated by START OF STRING (SOS) represented by ESC 5/8 (in a 7-bit environment, ESC = 1/11) or 9/8 (in a 8-bit environment).

A character string is terminated by the delimiter STRING TERMINATOR (ST) represented by the escape sequence ESC 5/12 (in a 7-bit environment, ESC = 1/11) or 9/12 (in a 8-bit environment). Only the datatypes String and Record use character strings.

Table C.5/T.101 – 7-bit Code Table as used for GDS coding

					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
						0	1	2	3	4	5	6	7
b ₄	b ₃	b ₂	b ₁										
0	0	0	0	0									
0	0	0	1	1									
0	0	1	0	2									
0	0	1	1	3									
0	1	0	0	4									
0	1	0	1	5									
0	1	1	0	6									
0	1	1	1	7									
1	0	0	0	8									
1	0	0	1	9									
1	0	1	0	10									
1	0	1	1	11									
1	1	0	0	12									
1	1	0	1	13									
1	1	1	0	14									
1	1	1	1	15									



C.7.1 Encoding principles of the opcode

The encoding technique used when encoding opcodes supplies:

- the basic opcode set;
- extension opcode sets.

The description of this encoding technique is, therefore, divided into two parts;

- description of the encoding technique of the basic opcode set;
- description of the extension mechanism.

C.7.1.1 Encoding technique of the basic opcode set

The basic opcode set consists of single-byte and double-byte opcodes. The general structure of an opcode is shown in Figure C.15.

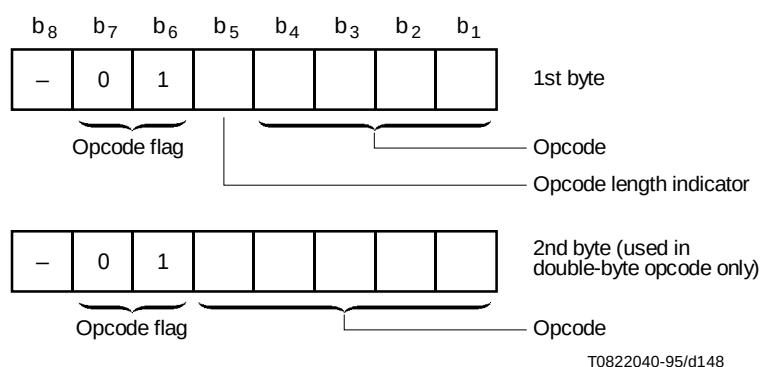


Figure C.15/T.101 – Opcode encoding structure

For single-byte opcodes the opcode length indicator, bit b₅, is ZERO (opcodes of column 2). Bits b₄ to b₁ are used to encode the opcode.

For double-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 are used to encode the opcode (first byte of the opcode is from column 3, the second byte being from column 2 or 3).

The code EXTEND OPCODE SPACE (EOS, 3/15) is used in a different sense (see C.7.1.2).

The basic opcode set, supplied by this encoding technique consists of 496 opcodes, being:

- 16 single-byte opcodes (from column 2);
- $15 \times 32 = 480$ double-byte opcodes (first byte from column 3 except code 3/15, second byte from column 2 or column 3).

C.7.1.2 Extension mechanism

The basic opcode set can be extended with an unlimited number of extension opcode sets by means of the EXTEND OPCODE SPACE code (EOS, 3/15).

The N-th extension opcode set consists of opcodes of the basic opcode set, prefixed with n times the code EOS. The three possible formats of an opcode from the N-th extension opcode set are:

Opcode format	Extension codes	Basic opcode set codes
1	<EOS> ... <EOS> n times	<2/x>
2	<EOS> ... <EOS> n times	<3/y> <2/z>
3	<EOS> ... <EOS> n times	<3/y> <3/z>

<EOS> = 3/15
 x = 0,1,...,15
 y = 0,1,...,14
 z = 0,1,...,15
 n = 0,1,.....

n = 0 selects the basic set.
 n = 1 selects the first extension opcode set.
 n = N selects the N-th extension opcode set.

The number of opcodes supplied by this encoding technique (basic opcode set plus extension opcode sets) is:

- 16 single-byte opcodes from the basic opcode set (opcode format 1, n = 0);
- 480 double-byte opcodes from the basic opcode set (opcode format 2 and 3, n = 0)
 16 double byte opcodes from the 1st extension opcode set (opcode format 1, n = 1);
- 480 N-byte opcodes from extension opcode set N – 2 (opcode format 2 and 3, n = N – 2)
 16 N-byte opcodes from extension opcode set N – 1 (opcode format 1, n = N – 1).

C.7.2 Encoding principles of the operands

The operand part of a primitive may contain one or more operands, each operand consisting of one or more bytes. The general format of an operand byte is given in Figure C.16.

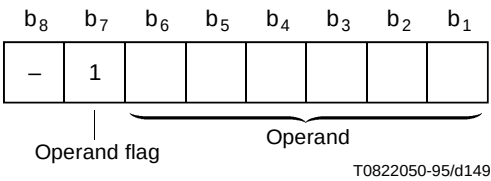


Figure C.16/T.101 – Operand encoding structure

The encoding of the operands may make use of the following *DATATYPES*:

- a) Identifier (ID);
- b) Enumerated type (E);
- c) String (S);
- d) Record (REC);
- e) Point (P);
- f) Colour index (CI);
- g) Index (IX);
- h) Size value (V)
- i) Integer number (I);
- j) Real number (R);
- k) Matrix (M);
- l) Colour direct (CD);
- m) Colour list (CL).

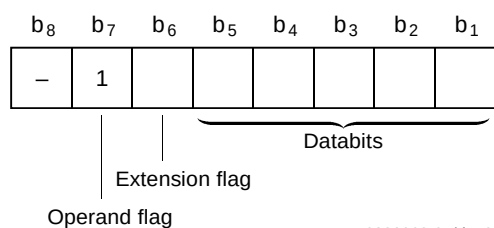
To encode the data types five *STRUCTURES* are available:

- 1) Basic format;
- 2) Real format;
- 3) Bitstream format;
- 4) String format;
- 5) Record format.

These five structures will be explained in the following subclause.

C.7.2.1 Basic format

Each Basic format operand is coded as a sequence of one or more bytes, which structure is shown in Figure C.17.



T0822060-95/d150

Figure C.17/T.101 – Basic format structure

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 Basic format is used to encode:

- Identifiers (ID);
- Enumerated types (E);

- Colour indices (CI);
- Indices other than colour indices (IX);
- Integer numbers (I).

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.

Signed integers are encoded using the modulus and sign notation. The range of integer numbers is subdivided into a non-negative range and a negative range, using bit b_5 of the first byte as a sign bit. If bit b_5 is set to ZERO the integer is non-negative; if bit b_5 is set to ONE the integer is negative. PLUS ZERO is considered to be non-negative. MINUS ZERO is used in incremental mode encoding (see C.7.2.2.3.3). Bits b_4 to b_1 of the first byte and bits b_5 to b_1 of subsequent bytes are used to encode the modulus of the integer.

In some situations the range of integers is used to indicate two categories of parameters, e.g. private and standardized use. Non-negative values indicate standardized use and negative values indicate private use.

The datatype Enumerated types is encoded in the same manner as an Integer. The datatypes Identifier, Colour index and Index are encoded in the Basic format without a sign bit.

In Figures C.18 to C.25, some examples are given.

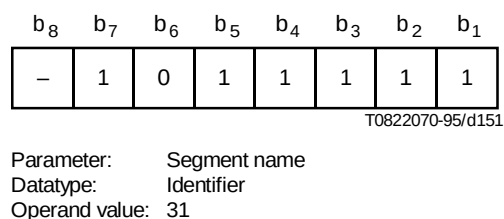


Figure C.18/T.101 – Identifier encoding

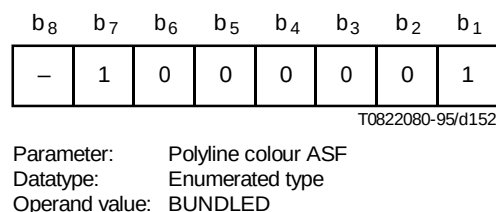


Figure C.19/T.101 – Enumerated type encoding

b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
–	1	0	1	0	0	0	0

T0822090-95/d153

Parameter: Polyline colour index
 Datatype: Colour index
 Operand value: 16

Figure C.20/T.101 – Colour index encoding

b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
–	1	0	0	0	0	1	0

T0822100-95/d154

Parameter: Line type
 Datatype: Integer
 Operand value: +2 (standardized line type 2)

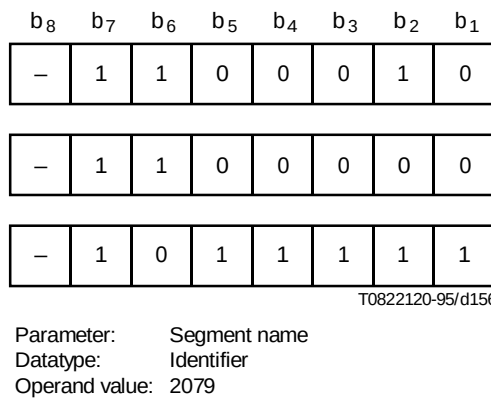
**Figure C.21/T.101 – Integer encoding
for standardized use**

b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
–	1	0	1	0	0	1	0

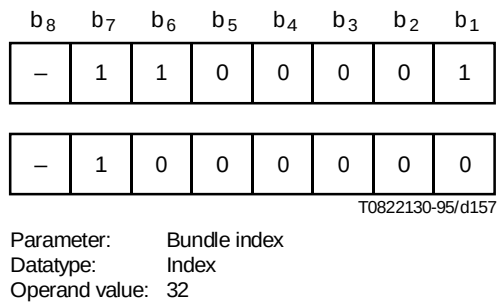
T0822110-95/d155

Parameter: Line type
 Datatype: Integer
 Operand value: –2 (private line type 2)

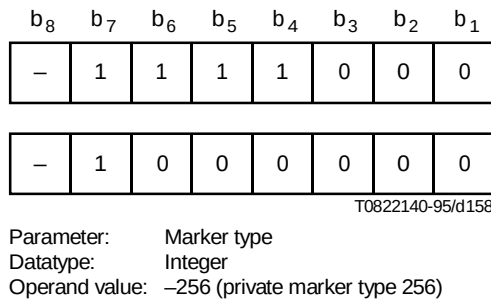
**Figure C.22/T.101 – Integer encoding
for private use**



**Figure C.23/T.101 – Identifier encoding
(more than one byte)**



**Figure C.24/T.101 – Index encoding
(more than one byte)**



**Figure C.25/T.101 – Integer encoding
for private use (more than one byte)**

C.7.2.2 Real format

Each real format consists of a mantissa part and an optional exponent part, both coded in Basic format. The exponent is the power of two by which the mantissa is to be multiplied. The exponent may be implicitly defined as a default exponent, which is then omitted in the Real format or the exponent may be coded explicitly as the second part of the Real format.

$$\langle \text{real format} \rangle = \langle \text{mantissa part} \rangle [\langle \text{exponent part} \rangle]$$

Whether or not the exponent part is explicitly coded as a part of the Real format depends on:

- the current 'explicit exponent allowed' value, which can be set by the 'explicit exponent allowed' parameter of the SET COORDINATE PRECISION and SET REAL PRECISION primitives;
- the value of the 'exponent follows' flag in the mantissa part of a Real format.

The Real format is used to encode:

- Real numbers (R);
- Size values (V);
- Points (P);
- Matrices (M).

C.7.2.2.1 Mantissa

The mantissa is an integer, which is coded in the Basic format. In the first byte of the mantissa, bit b_5 is used as the sign bit: ZERO for a non-negative mantissa, ONE for a negative mantissa. The MINUS ZERO code for a mantissa is not permitted and reserved for future use.

If the current 'explicit exponent allowed' value is ALLOWED, bit b_4 of the first byte of a mantissa is used as the 'exponent follows' bit: ONE if an explicit exponent follows the mantissa, ZERO as no exponent follows the mantissa.

If the current 'explicit exponent allowed' value is FORBIDDEN, bit b_4 of the first byte of a mantissa is used as databit.

Figures C.26 and C.27 show the general format of a mantissa. Figure C.28 shows an example of a multiple-byte mantissa.

C.7.2.2.2 Exponent

If the current 'explicit exponent allowed' value is ALLOWED and the 'exponent follows' bit in a mantissa is ONE, then an explicit exponent follows the mantissa. The exponent is coded as an integer in Basic format.

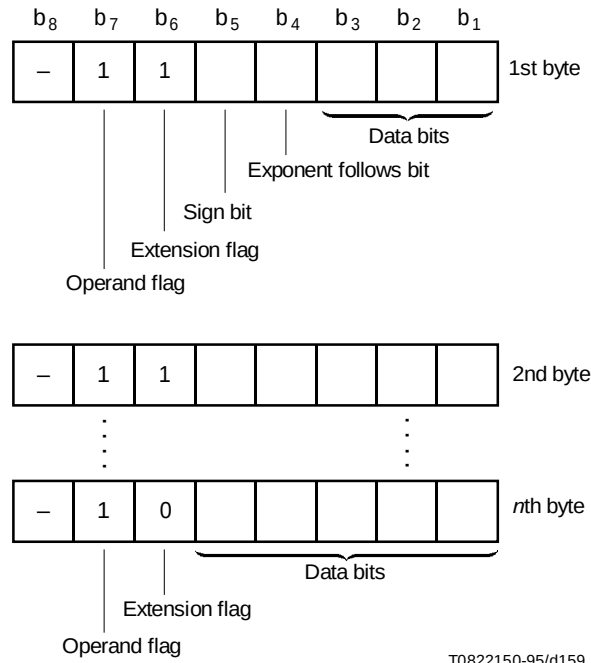
If the current 'explicit exponent allowed' value is FORBIDDEN or if the current 'explicit exponent allowed' value is ALLOWED, together with the 'exponent follows' bit in the mantissa part having the value ZERO, then the exponent is omitted from the coding and an implicit default exponent value is assumed. The default value for an exponent depends on the data type being encoded.

If an exponent is omitted in a real number, a default value determined by SET REAL PRECISION is assumed.

If an exponent is omitted in a size value, a default value determined by SET COORDINATE PRECISION is assumed.

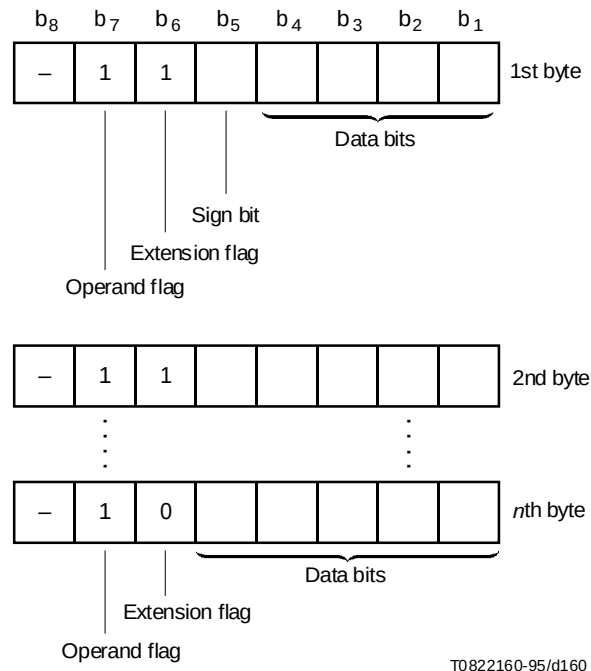
If an exponent is omitted in the x-component of a point, the exponent used for the x-component of the preceding point is assumed. Likewise, the exponent in a y-coordinate defaults to the value of the exponent in the preceding y-coordinate. For individual points (points not part of a point list) and for the first point of a point list, an omitted exponent assumes a default value determined by SET COORDINATE PRECISION.

If an exponent is omitted in one of the elements of a matrix, the default value, which is assumed for the exponent, depends on the matrix element which is concerned (see C.7.2.2.4).



Explicit exponent allowed = ALLOWED

Figure C.26/T.101 – Encoding of a mantissa using the basic format



Explicit exponent allowed = FORBIDDEN

Figure C.27/T.101 – Encoding of a mantissa using the basic format

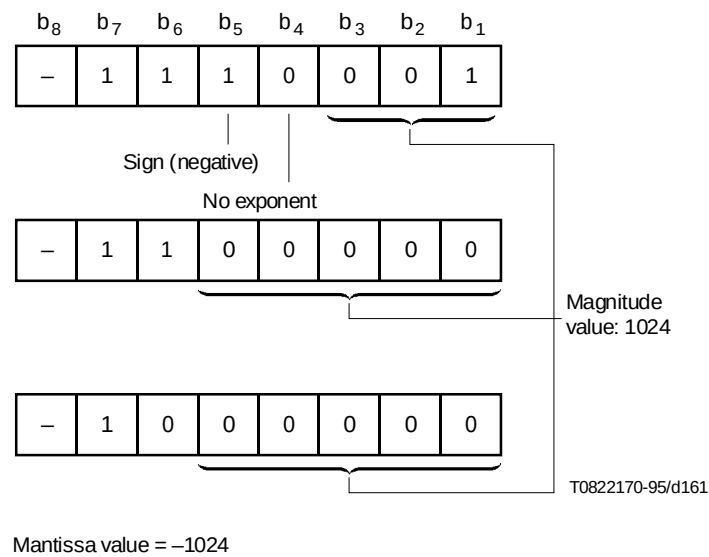


Figure C.28/T.101 – A multiple-byte mantissa

C.7.2.2.3 Points and point lists

Point lists may be coded with either one or two coding structures or a mixture of the two. These structures are called 'Displacement Mode' and 'Incremental Mode'. Individual points (datatype P) and the first point of each point list, however, must always be coded in Displacement Mode.

C.7.2.2.3.1 Displacement mode

In Displacement Mode, each point is coded as a sequence of two size value parameters. The first size value gives the x-component of the point's displacement from the preceding point, while the second size value specifies the y-component of that displacement.

$$\langle P \rangle = \langle V: \text{delta } x \rangle \langle V: \text{delta } y \rangle$$

For individual points (datatype P) and for the first point of each point list, the displacement values delta x and delta y are displacements measured from the origin. For points after the first point in a point list, each displacement is measured from the preceding point of the point list (this is true regardless whether the preceding point was coded in Displacement Mode or in Incremental Mode).

C.7.2.2.3.2 Incremental mode

The Incremental Mode is defined as a so called Differential Chain Code (DCC). The data in this mode does not reflect actual coordinates, but identifies points on a Ring. A Ring is a set of points on a square which centre is the previously identified point. The first centre point is encoded in Displacement Mode.

A Ring is characterized by its Radius (R in Basic Grid Units), its Angular resolution (by a factor p) and its Direction (D). The maximum number of points on a Ring is 8R. The actual number of points on a Ring with a given Angular resolution factor p follows from:

$$N = \frac{8R}{2p}$$

with

$$p = 0, 1, 2, 3$$

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 and N is odd for the remaining part, the remaining part (with N being odd) is discarded.

The points on the Ring are numbered, starting at the Direction point, from 0 to M-1 for the upper part of the Ring and from -1 to -M for the lower part of the Ring, with $M = N/2$.

Figure C.29 shows a Ring with Radius R = 3 and Angular resolution factor p = 0 respectively 1.

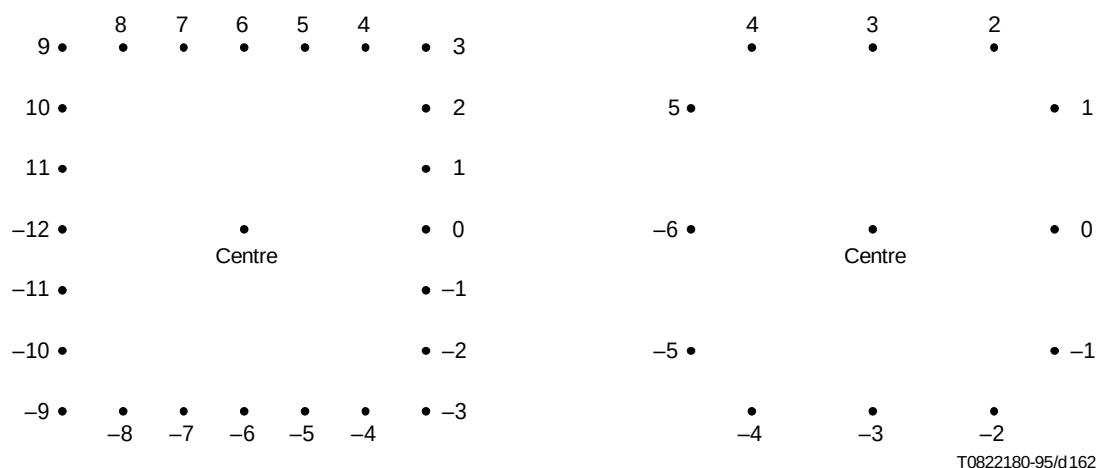


Figure C.29/T.101 – Some example Rings with point numbering

The Direction of a Ring is identified by the position of the point with number ZERO. The initial position of this point is on the positive x-axis, while the Cartesian axes are drawn through the centre point of the Ring. The Direction of the Rings following the initial one is dependent on the direction of the increments. This Direction is determined in the following way:

If P1 is the previous centre point and the current centre point is P2 (P2 is a point on the Ring with the centre in P1). The position of the point with number ZERO on the Ring, with P2 as centre point, is opposite to point P1, this is the Direction of the Ring. So the Direction of the Ring is dependent of the writing direction as indicated by the last increment. The position of the increment on the new Ring (centre P2) is described as the difference between the position of point P2 on the previous Ring and the position of the new point P3 on the current Ring.

In the DCC only the differences between points on the consecutive Rings are coded. Or to state it in another way, the Direction of the Ring is dependent on the direction of the line to be displayed. As shown in Figure C.30, the position of point P3 is defined by the difference:

$$P3 - P2 = -1$$

P3 and P2 being point numbers on the two Rings, numbered as given in Figure C.29. The Direction (position of the point with number ZERO) is defined by D.

The basic Radius is expressed in Basic Grid Units (BGUs). The basic Radius of the Ring, as used in the Incremental mode, is dependent on the granularity code as set with the SET COORDINATE PRECISION primitive. The default basic Radius follows from:

$$\text{default basic Radius} = 2^{\max(0, -8 - \text{granularity code})}$$

The basic Radius as derived from the granularity code may be changed to any value greater than ZERO with the SET DOMAIN RING-primitive. with this primitive one can also change the Angular resolution factor p. The default value for p = 0 and p can only be 0, 1, 2, or 3.

The basic Radius must be equal to or greater than ONE. If the basic Radius is less than ONE, the value of the default basic Radius is assumed for the basic Radius.

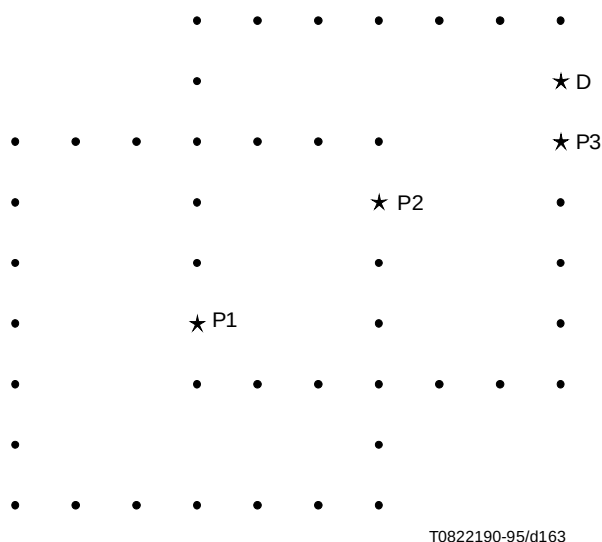


Figure C.30/T.101 – Change of direction with R = 3

The encoding used in Incremental mode makes use of the DCC property by using variable length code-words (Huffman Code). The encoding also allows changing of the basic Radius and the Angular resolution factor temporarily. The Radius actually used (= Rt) can have a value of R, 2R 4R or 8R, where R is the basic Radius as set before entering the Incremental mode. The Angular resolution factor actually used (= pt) p can be 0, 1, 2, or 3.

The Huffman Code table used in the Incremental mode is a fixed length table. To allow the encoding of more points on a Ring two Escapes codes are defined. With these Escape codes the points outside the Huffman Code table can be addressed. The end of the Incremental mode data is indicated by an End of Block value in the Huffman Code table.

The fixed Huffman Code table is given in Table C.6.

Table C.6/T.101 – Huffman Code table for Incremental mode

Length	Code-word	Point number
2	00	0
2	10	1
2	01	-1
4	1100	2
4	1101	-2
6	111000	3
6	111001	-3
6	111010	4
6	111011	-4
8	11110000	5
8	11110001	-5
8	11110010	6
8	11110011	-6
8	11110100	7
8	11110101	-7
8	11110110	8
8	11110111	-8
10	1111100000	9
10	1111100001	-9
10	1111100010	10
10	1111100011	-10
10	1111100100	11
10	1111100101	-11
10	1111100110	12
10	1111100111	-12
10	1111101000	13
10	1111101001	-13
10	1111101010	14
10	1111101011	-14
10	1111101100	15
10	1111101101	-15
10	1111101110	16
10	1111101111	-16
10	1111110000	17
10	1111110001	-17
10	1111110010	18
10	1111110011	-18
10	1111110100	19
10	1111110101	-19
10	1111110110	C1
10	1111110111	-20
10	1111111000	C2
10	1111111001	C3
10	1111111010	C4
10	1111111011	C5
10	1111111100	C6
10	1111111101	IM-ESC 1
10	1111111110	IM-ESC 2
10	1111111111	End of Block

The <End of Block> code from the Huffman Code table identifies the end of the Incremental mode data. Remaining bits in the last Incremental mode data byte have no meaning, they will be ignored.

The Incremental Mode escape codes <IM-ESC 1> and <IM-ESC 2> are used to extend the addressable number of points, e.g. points outside the range -20 to 19. The code <IM-ESC 1> adds +20 or -20 to the following code depending on the sign of that following point. The code <IM-ESC 2> adds +40 or -40 to the following code, depending on the sign. The escape codes can follow each other in any desired order. The following examples demonstrate some possible combinations, [n] is a point number.

<IM-ESC 1> [1]	=	point number	21
<IM-ESC 1> [-1]	=	point number	-21
<IM-ESC 2> [14]	=	point number	54
<IM-ESC 2> [-12]	=	point number	-52
<IM-ESC 1> <IM-ESC 2> [6]	=	point number	(20 + 40 + 6) = 66
<IM-ESC 2> <IM-ESC 1> [-18]	=	point number	(-40 + -20 + -18) = -78

The codes C1 up to C6 are used to change temporarily the parameters that define the Ring to be used to Rt and pt. The values of Rt are taken from the range R, 2R, 4R and 8R, where R is the value of the basic Radius before entering the Incremental mode. The values of pt are taken from the range 0, 1, 2 and 3. The function of these codes is as follows:

- a) C1
Change the Ring parameters, Rt and pt, to the next higher value, e.g. if the Rt = R, the next higher is 2R, if pt = 0 the next higher value is 1. If Rt = 8R or pt = 3 the code <C1> has no effect.
- b) C2
Change the Ring parameters, Rt and pt, to the next lower value. The effect of the code <C2> is the inverse of code <C1>. If Rt = R or pt = 0 the code <C2> has no effect.
- c) C3
Change the basic Radius Rt to the next higher value. The code <C3> has no effect if the current basic Radius Rt = 8R.
- d) C4
Change the Angular resolution factor p to the next higher values. The code <C4> has no effect if the current pt = 3.
- e) C5
Change the basic Radius Rt to the next lower value. The code <C5> has no effect if the current basic Radius = R.
- f) C6
Change the Angular resolution factor pt to the next lower value. The code <C6> has no effect if the current pt = 0.

In addition, those codes (C1 to C6) set the position of the point with number ZERO on the positive x-axis, while the cartesian axes are drawn through the centre point of the Ring. Changing the Ring parameters from R to Rt and from p to pt by means of these codes will have effect up to the next Incremental mode End of Block code. The basic Radius R and the Angular resolution factor p are not affected. This implies that the Ring parameters of subsequent Incremental mode point lists are not affected by these codes.

C.7.2.2.3.3 Incremental mode encoding

The structure of the Incremental mode format is given in Figure C.31.

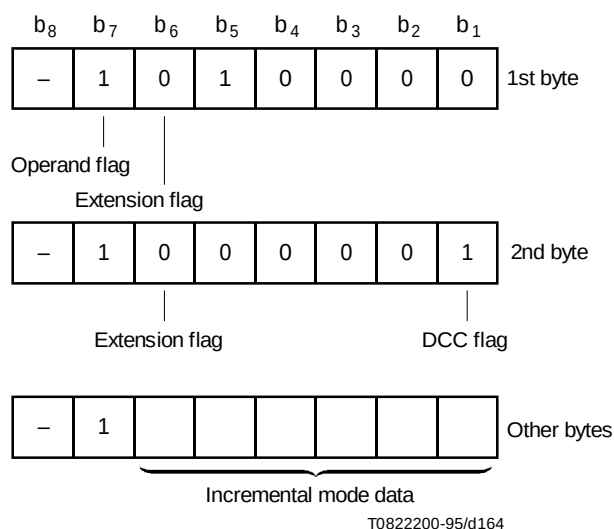


Figure C.31/T.101 – Incremental mode structure

The first byte is according to the Basic format structure indicating an integer value MINUS ZERO. The second byte is structured according to the Basic format structure. Bit b_1 now is set to ONE to identify the use of Differential Chain Coding (DCC). The bits b_5 to b_2 are reserved for future use and now set to ZERO.

The Incremental mode uses variable length code-words. This implies that the code-words do not fit in the Incremental mode data bits (bit b_6 to bit b_1 of the other bytes). The code-words are packed in consecutive bits of the Incremental mode bytes, starting from high numbered bits to lower numbered bits. If the code-word does not fit in one byte, the most significant part is packed in the first byte, the remaining part is packed in the second byte and so on.

The end of Incremental mode data is identified by the <End of Block> code. Remaining bits in the last Incremental mode data byte have no meaning, they will be ignored.

C.7.2.2.4 Matrices

A matrix is a structured datatype consisting of six matrix elements. The six matrix elements are ordered in three columns, each column containing two rows. The matrix elements are numbered M_{xy} , x indicating the row (1..2) and y indicating the column (1..3). Element M_{23} denotes the element in the third column of row 2.

The elements of column 3 (M_{13} and M_{23}) are in NDC coordinates, the other elements are real numbers.

The elements of column 3 (M_{13} and M_{23}) are encoded in the same way as single points, using the Real format – displacement mode. The displacement values are measured from the origin.

The other elements are encoded as real numbers, using the Real format.

A matrix is encoded column by column. This means that M_{11} is the first element to be encoded. The next element to be encoded is M_{21} , then M_{12} , etc.

C.7.2.3 Bitstream format

Each Bitstream format operand is encoded as a sequence of one or more bytes, which structure is shown in Figure C.32.

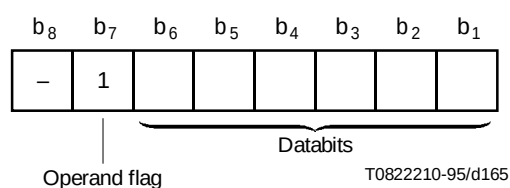


Figure C.32/T.101 – Bitstream format structure

The Bitstream format is used to encode:

- Incremental mode coordinates (see C.7.2.2.3.3);
- Colour direct (see C.7.2.3.1);
- Colour index lists (see C.7.2.4.1.1).

Bitstream data are packed in consecutive databits starting from high numbered bits to lower numbered bits of the first byte for the most significant part of the bitstream data.

The end of a Bitstream format operand cannot be derived from the Bitstream format itself (the format is not self-delimiting).

- Encoding Incremental mode coordinates, the end of the data (which identifies the end of the Bitstream format operand) is identified by the <End of Block> code.
- Encoding Colour index lists, the number of bits needed to encode the Colour index list (which identifies the end of the Bitstream format operand) is set by the SET COLOUR INDEX PRECISION primitive.
- Encoding Colour direct data, the number of bits needed to encode this data (which identifies the end of the Bitstream format operand) is set by the SET COLOUR HEADER primitive.

C.7.2.3.1 Colour direct encoding

In encoding colour data which must be loaded into a colour table (Colour direct), the following applies. The Bitstream format is composed of a number of bytes as derived from the unit resolution parameter of the SET COLOUR HEADER primitive.

Each byte contains two bits for each component (R, G, B) (see Figure C.33). The first byte contains the most significant bits. In the last byte, bit b_3 , b_2 and b_1 may be left unused.

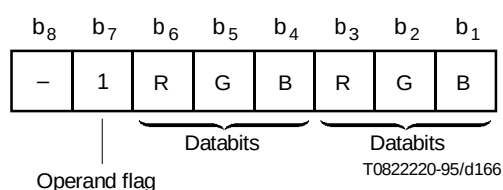


Figure C.33/T.101 – Colour direct encoding

Figure C.34 gives an example of the colour loading with a unit resolution of 4.

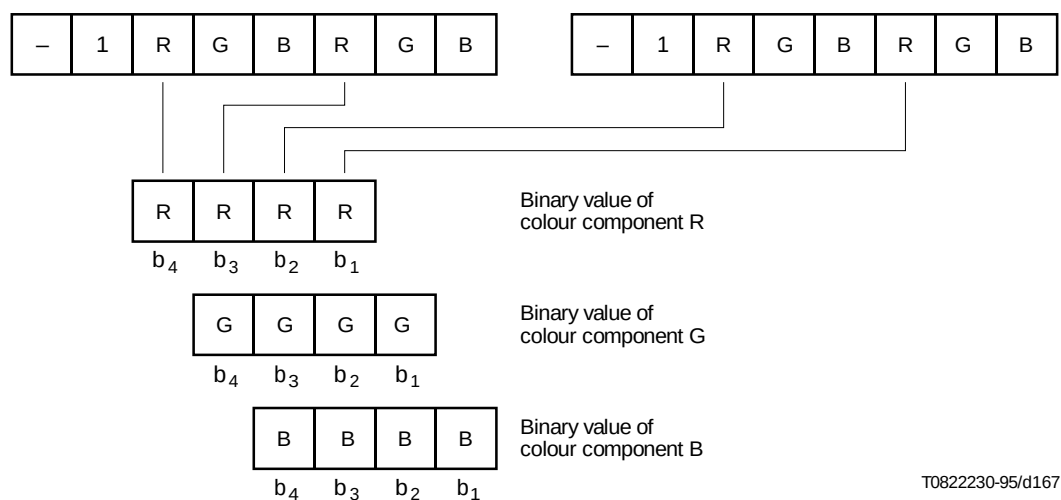


Figure C.34/T.101 – Colour loading with a unit resolution of 4

C.7.2.4 Colour lists

A colour list is a structured datatype consisting of a sequence of colour list elements, each element specifying a colour value. The general format of a colour list is:

$$\langle \text{colour list} \rangle = \underbrace{\langle \text{colour list element} \rangle}_{n \text{ times}} \quad n = 1, 2, \dots$$

The colours can be specified:

- Indexed, with colour list elements representing colour indices pointing into a colour table. The colour list is called a colour index list.
- Direct, with colour list elements representing the RGB components of the colour data. The colour list is called a colour direct list.

There are two ways to encode a colour list. If many adjacent colour list elements have the same colour value, runlength encoding is efficient. However, for short runs of only one or two colour list elements with the same colour value, it is more efficient to encode the colour list elements individually.

When using runlength encoding, for each run the colour of the colour list elements is encoded (run colour), followed by a count telling how many colour list elements are in the run (run count).

C.7.2.4.1 Colour list encoding

The method, used in encoding a colour list is specified in the first byte of a colour list (see Figure C.35).

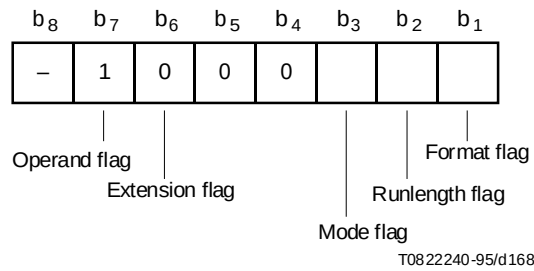


Figure C.35/T.101 – First byte of colour list encoding

The mode flag, bit b_3 , specifies whether the colour list is encoded as a colour index list (mode flag is ZERO) or as a colour direct list (mode flag is ONE).

The runlength flag, bit b_2 , specifies whether runlength encoding is used (runlength flag is ONE) or not (runlength flag is ZERO).

The format flag, bit b_1 , specifies whether the Basic format (format flag is ZERO) or the Bitstream format (format flag is ONE) is used to encode the colour list.

Bits b_5 and b_4 are reserved for future use and now set to ZERO

In the next subclauses the different methods of encoding colour lists are described in more detail.

C.7.2.4.1.1 Individual encoding of colour index lists

This encoding method specifies a colour list by individually encoding all colour list elements as colour indices.

The colour indices can be encoded using either the Basic format or the Bitstream format.

If Basic format encoding is specified, bit b_3 of the first byte set to ZERO, the indices of the Colour index list are encoded using the Basic format described in C.7.2.1 for each separate index.

If Bitstream format encoding is specified, bit b_3 of the first byte set to ONE, the indices of the Colour index list are encoded using the Bitstream format described in C.7.2.3. The length (in bits) of each index is identified by the SET COLOUR INDEX PRECISION primitive. These indices are packed in consecutive bits starting from bit b_6 of the first Bitstream format byte. If an index does not fit in one byte, the most significant part is packed in the first byte, the remaining part is packed in the second byte and so on.

Some examples are given in Figures C.36 and C.37.

C.7.2.4.1.2 Runlength encoding of colour index lists

When using runlength encoding of a colour index list both the run colour and the run count are encoded using the Basic format (respectively encoded as a Colour index datatype and as an Integer datatype), so the format flag, bit b_1 of the first byte, is always set to ZERO.

An example is given in Figure C.38.

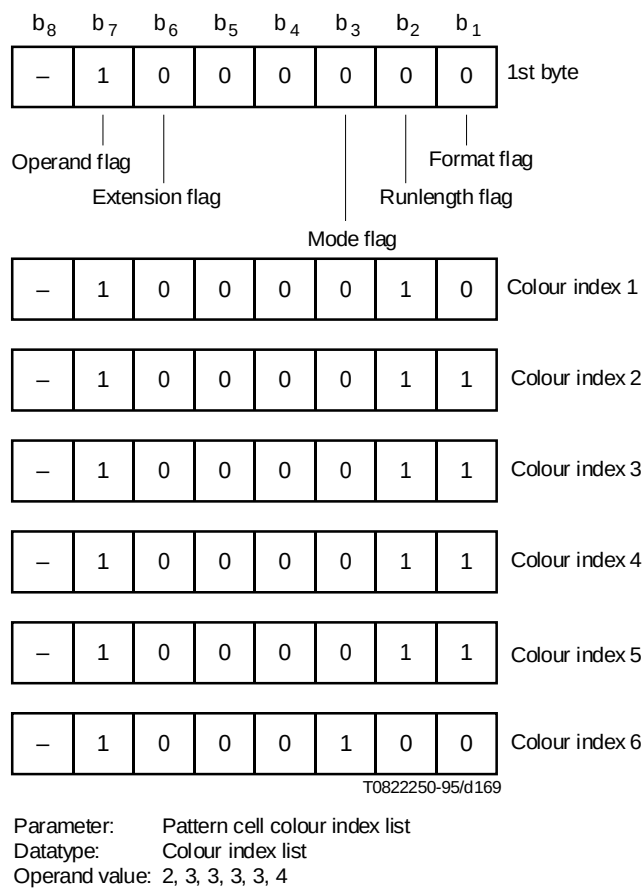


Figure C.36/T.101 – Colour index list individually encoded using the basic format

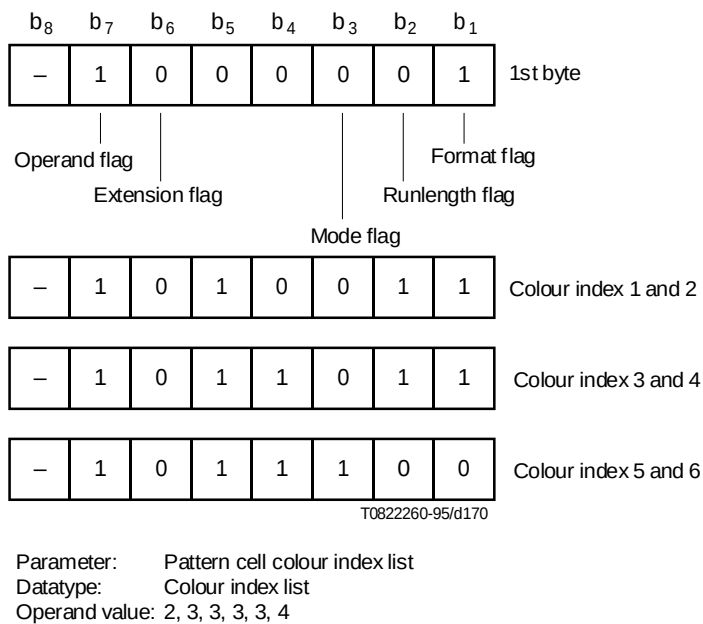


Figure C.37/T.101 – Colour index list individually encoded using the Bitstream format

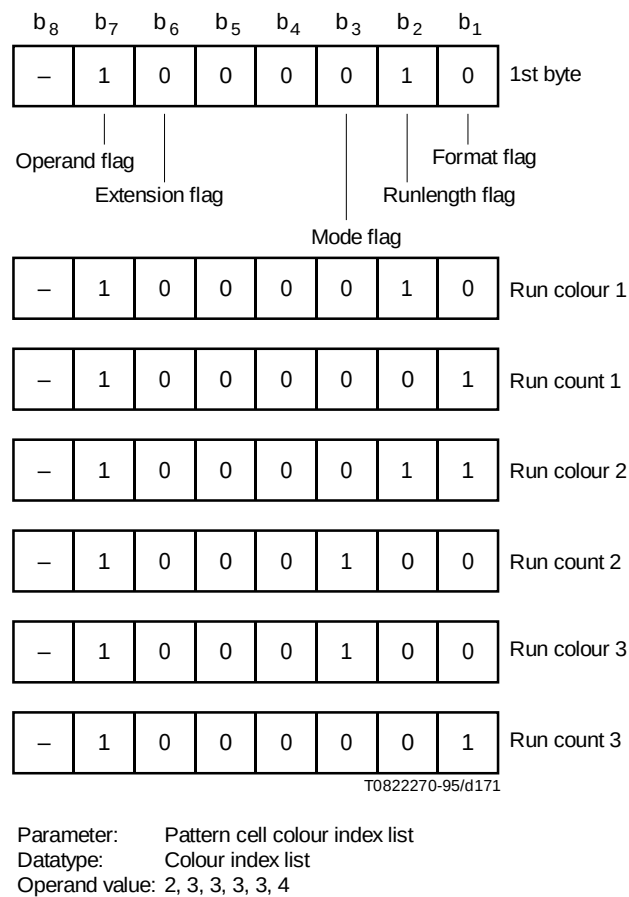


Figure C.38/T.101 – Colour index list encoded using runlength encoding

C.7.2.4.1.3 Encoding of colour direct lists

In this standard only colour index lists are defined, so the mode flag, bit b₃ of the first byte, is always set to ZERO. The encoding of colour direct lists is reserved for future use.

C.7.2.5 String format

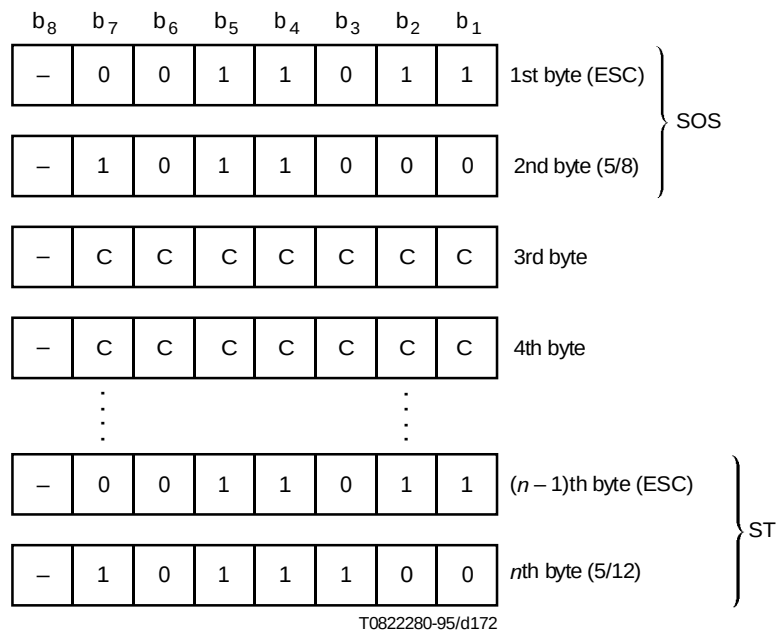
The structure of a String format is given in Figures C.39 and C.40.

The number of bytes needed to encode a string operand equals to the number of characters of the string plus four (for SOS and ST, coded 1/11, 5/8 and 1/11, 5/12) in a 7-bit environment and equals to the number of characters of the string plus two (for SOS and ST, coded 9/8 and 9/12) in a 8-bit environment.

The encoding of a string is the only exception of the general coding rules indicated in C.7.2.

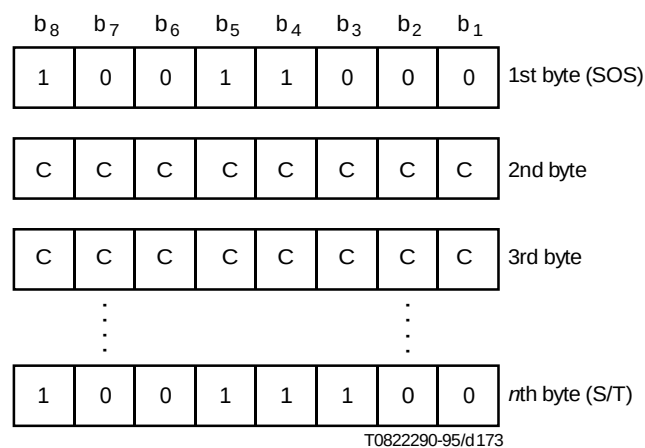
C.7.2.6 Record format

The structure of the record format is implementation dependent. Its format must consist of one or more already defined formats (see C.7.2.1 to C.7.2.5).



NOTE – Databits are marked C.

Figure C.39/T.101 – String format structure in a 7-bit environment



NOTE – Databits are marked C.

Figure C.40/T.101 – String format structure in a 8-bit environment

C.8 Encoding of the primitives

The coding of the primitives is described in the following subclauses. Subclause C.8.1 gives the code of each primitive. Subclause C.8.2 defines the encoding of each primitive indicating the order of the parameters along with their specific data type.

C.8.1 Primitive encoding

The following table gives the opcode for each primitive.

Primitives	Opcode	
	Byte 1	Byte 2
<i>WORKSTATION MANAGEMENT PRIMITIVES</i>		
OPEN WORKSTATION	3/2	3/0
CLOSE WORKSTATION	3/2	3/1
ACTIVATE WORKSTATION	3/2	3/2
DEACTIVATE WORKSTATION	3/2	3/3
CLEAR WORKSTATION	3/2	2/0
SET DEFAULTS	3/2	2/5
GDS EXCAPE1	3/2	2/11
<i>OUTPUT DRAWING PRIMITIVES</i>		
POLYLINE	2/0	–
POLYMARKER	2/2	–
FILL AREA	2/1	–
TEXT	2/4	–
CELL ARRAY	2/3	–
GDP	2/5	–
<i>OUTPUT PRIMITIVES RELATED TO DISPLAY ELEMENT ATTRIBUTES</i>		
SET POLYLINE REPRESENTATION	3/1	3/11
SET POLYLINE INDEX	3/1	2/3
SET LINE TYPE	3/1	2/2
SET LINE WIDTH SCALE FACTOR	3/1	2/1
SET POLYLINE COLOUR INDEX	3/1	2/0
SET POLYMARKER REPRESENTATION	3/1	3/13
SET POLYMARKER INDEX	3/1	2/14
SET MARKER TYPE	3/1	2/12
SET MARKER SIZE SCALE FACTOR	3/1	2/13
SET POLYMARKER COLOUR INDEX	3/1	2/11
SET FILL AREA REPRESENTATION	3/1	3/12
SET FILL AREA INDEX	3/1	2/10
SET FILL AREA INTERIOR STYLE	3/1	2/5
SET FILL AREA COLOUR INDEX	3/1	2/4
SET FILL AREA STYLE INDEX	3/1	2/6
SET PATTERN REPRESENTATION	3/1	3/15
SET PATTERN REFERENCE POINT	3/1	2/9
SET PATTERN VECTORS	3/1	2/8
SET TEXT REPRESENTATION	3/1	3/14
SET TEXT INDEX	3/1	3/6
SET TEXT FONT AND PRECISION	3/1	3/4
SET CHARACTER EXPANSION FACTOR	3/1	3/0
SET CHARACTER SPACING	3/1	3/3
SET TEXT COLOUR INDEX	3/1	2/15
SET TEXT PATH	3/1	3/2
SET CHARACTER VECTORS	3/1	3/1
SET TEXT ALIGNMENT	3/1	3/5
SET COLOUR REPRESENTATION	3/2	3/8
SET ASPECT SOURCE FLAGS	3/1	2/7
<i>TRANSFORMATION PRIMITIVES</i>		
SET WORKSTATION WINDOW	3/2	2/1
SET WORKSTATION VIEWPORT	3/2	2/2

Primitives	Opcode	
	Byte 1	Byte 2
<i>CLIPPING PRIMITIVES</i>		
SET CLIPPING RECTANGLE	3/2	2/3
<i>CONTROL PRIMITIVES</i>		
UPDATE WORKSTATION	3/2	3/4
SET DEFERRAL STATE	3/2	3/5
EMERGENCY CLOSE	3/2	2/6
<i>SEGMENT RELATED PRIMITIVES</i>		
CREATE SEGMENT	3/3	2/0
CLOSE SEGMENT	3/3	2/1
RENAME SEGMENT	3/3	2/2
DELETE SEGMENT FROM WORKSTATION	3/3	2/3
DELETE SEGMENT	3/3	2/8
REDRAW ALL SEGMENTS ON WORKSTATION	3/3	2/9
SET HIGHLIGHTING	3/3	2/6
SET VISIBILITY	3/3	2/7
SET SEGMENT TRANSFORMATION	3/3	2/5
SET SEGMENT PRIORITY	3/3	2/12
ASSOCIATE SEGMENT WITH WORKSTATION	3/3	2/10
COPY SEGMENT TO WORKSTATION	3/3	2/11
INSERT SEGMENT	3/3	2/4
<i>PROTOCOL DESCRIPTOR PRIMITIVES</i>		
SET DOMAIN RING	3/2	2/4
SET COLOUR HEADER	3/2	2/8
SET COORDINATE PRECISION	3/2	2/9
SET REAL PRECISION	3/2	3/9
SET COLOUR INDEX PRECISION	3/2	2/10

C.8.2 Coding of the primitives

The notational conventions used are:

<symbols> = 1 occurrence
 <symbols> (n) = n or more occurrences (with n >= 1)
 <symbols> (= n) = n occurrences (with n >= 1)
 <symbols> (0) = optional, 0 or more occurrences
 [comment] = explanation of a production
 <x : y> = construction x with meaning y
 <symbols> | <symbols> = choice

C.8.2.1 Workstation management primitives

C.8.2.1.1 OPEN WORKSTATION

<OPEN WORKSTATION opcode: 3/2, 3/0>

<identifier : workstation identifier>

C.8.2.1.2 CLOSE WORKSTATION

<CLOSE WORKSTATION opcode : 3/2, 3/1>

<identifier : workstation identifier>

C.8.2.1.3 ACTIVATE WORKSTATION

<ACTIVATE WORKSTATION opcode: 3/2. 3/2>

<identifier : workstation identifier>

C.8.2.1.4 DEACTIVATE WORKSTATION

<DEACTIVATE WORKSTATION opcode: 3/2. 3/3>

<identifier : workstation identifier>

C.8.2.1.5 CLEAR WORKSTATION

<CLEAR WORKSTATION opcode: 3/2. 2/0>

<identifier : workstation identifier>

C.8.2.1.6 SET DEFAULTS

<SET DEFAULTS opcode : 3/2. 2/5>

C.8.2.1.7 GDS ESCAPE1

<GDS ESCAPE1 opcode: 3/2. 2/11

<integer : GDS escape identifier>

<record : GDS data record>

with

<integer : escape identifier> =

<integer : non-negative> [reserved]

| <integer : negative> [private escape]

C.8.2.2 Output workstation primitives

C.8.2.2.1 Output drawing primitives

C.8.2.2.1.1 POLYLINE

<POLYLINE opcode: 2/0>

<point : point list> (2)

C.8.2.2.1.2 POLYMARKER

<POLYMARKER opcode: 2/2>

<point : point list> (1)

C.8.2.2.1.3 FILL AREA

<FILL AREA opcode: 2/1>

<point : point list> (3)

C.8.2.2.1.4 TEXT

<TEXT opcode: 2/4>

<point : text position>

<string : character string>

C.8.2.2.1.5 CELL ARRAY

<CELL ARRAY opcode: 2/3>

<point : parallelogram (P, Q, R)> (= 3)

<integer : number of cells in PQ-direction> [value M]

<integer : number of cells in QR-direction> [value N]

<colour index list : cell colour index list>

C.8.2.2.1.6 GDP

<GDP opcode: 2/5>

```
<integer          : GDP identifier>
with
<integer          : GDP identifier > =
<integer: 0>      [rectangle]
|<integer: 1>      [circle]
|<integer: 2>      [circular arc 3 point]
|<integer: 3>      [circular arc 3 point chord]
|<integer: 4>      [circular arc 3 point pie]
|<integer: 5>      [circular arc centre]
|<integer: 6>      [circular arc centre chord]
|<integer: 7>      [circular arc centre pie]
|<integer: 8>      [ellipse]
|<integer: 9>      [elliptic arc]
|<integer: 10>     [ellipse arc chord]
|<integer: 11>     [ellipse arc pie]
|<integer: 12>     [spline]
|<integer: >12>    [reserved]
|<integer: negative> [private GDP]

if <integer: GDP identifier> = 0    [rectangle]
then
<point           : two points> (= 2)

if <integer: GDP identifier> = 1    [circle]
then
<point           : centre>
<size value      : radius>

if <integer: GDP identifier> = 2    [circular arc 3 point]
then
<point           : starting point>
<point           : intermediate point>
<point           : ending point>
if <integer: GDP identifier> = 3    [circular arc 3 point chord]
then
<point           : starting point>
<point           : intermediate point>
<point           : ending point>

if <integer: GDP identifier> = 4    [circular arc 3 point pie]
then
<point           : starting point>
<point           : intermediate point>
<point           : ending point>

if <integer: GDP identifier> = 5    [circular arc centre]
then
<point           : centre>
<size value      : radius>
<point           : start vector>
<point           : end vector>
```



```

if <integer: GDP identifier> = 6    [circular arc centre chord]

then
<point          : centre>
<size value     : radius>
<point          : start vector>
<point          : end vector>

if <integer: GDP identifier> = 7    [circular arc centre pie]

then
<point          : centre>
<size value     : radius>
<point          : start vector>
<point          : end vector>

if <integer: GDP identifier> = 8    [ellipse]

then
<point          : centre point>
<point          : endpoints> (= 2) [(X1, Y1) and (X2, Y2)]

if <integer: GDP identifier> = 9    [elliptic arc]

then
<point          : centre point>
<point          : endpoints> (= 2) [(X1, Y1) and (X2, Y2)]
<real          : T_start, T_end> (= 2)

if <integer: GDP identifier> = 10   [elliptic arc chord]

then
<point          : centre point>
<point          : endpoints> (= 2) [(X1, Y1) and (X2, Y2)]
<real          : T_start, Y_end> (= 2)

if <integer: GDP identifier> = 11   [elliptic arc pie]

then
<point          : centre point>
<point          : endpoints> (= 2) [(X1, Y1) and (X2, Y2)]
<real          : T_start, T_end> (= 2)

if <integer: GDP identifier> = 12   [spline]

then
<point          : point list> (3)

```

C.8.2.2.2 Output primitives related to display element attributes

C.8.2.2.2.1 SET POLYLINE REPRESENTATION

<SET POLYLINE REPRESENTATION opcode: 3/1, 3/11>

```

<identifier      : workstation identifier>
<index           : polyline index>
<integer         : line type>
<real            : line width scale factor>
<colour index    : polyline colour index>

with
<integer         : line type> =
<integer: 0>    [SOLID]
|<integer: 1>   [DASHED]

```

<integer: 2>	[DOTTED]
<integer: 3>	[DASHED-DOTTED]
<integer: >3>	[reserved]
<integer negative>	[private line type]

C.8.2.2.2.2 SET POLYLINE INDEX

<SET POLYLINE INDEX opcode: 3/1, 2/3>

<index	:	polyline index>
--------	---	-----------------

C.8.2.2.2.3 SET LINE TYPE

<SET LINE TYPE opcode: 3/1, 2/2>

<integer	:	line type>
----------	---	------------

with

<integer	:	line type> =
----------	---	--------------

<integer: 0>	[SOLID]
<integer: 1>	[DASHED]
<integer: 2>	[DOTTED]
<integer: 3>	[DASHED-DOTTED]
<integer: >3>	[reserved]
<integer: negative>	[private line type]

C.8.2.2.2.4 SET LINE WIDTH SCALE FACTOR

<SET LINE WIDTH SCALE FACTOR opcode: 3/1, 2/1>

<real	:	line width scale factor>
-------	---	--------------------------

C.8.2.2.2.5 SET POLYLINE COLOUR INDEX

<SET POLYLINE COLOUR INDEX opcode: 3/1, 2/0>

<colour index	:	polyline colour index>
---------------	---	------------------------

C.8.2.2.2.6 SET POLYMARKER REPRESENTATION

<SET POLYMARKER REPRESENTATION opcode: 3/1, 3/13>

<identifier	:	workstation identifier>
<index	:	polymarker index>
<integer	:	marker type>
<real	:	marker size scale factor>
<colour index	:	polymarker colour index>

with

<integer	:	marker type>=
<integer: 0>	[DOT]	
<integer: 1>	[PLUS SIGN]	
<integer: 2>	[ASTERISK]	
<integer: 3>	[CIRCLE]	
<integer: 4>	[DIAGONAL CROSS]	
<integer: >4>	[reserved]	
<integer: negative>	[private marker type]	

C.8.2.2.2.7 SET POLYMARKER INDEX

<SET POLYMARKER INDEX opcode: 3/1, 2/14>

<index	:	polymarker index>
--------	---	-------------------

C.8.2.2.2.8 SET MARKER TYPE

<SET MARKER TYPE OPCODE: 3/1, 2/12>

<integer : marker type>

with

<integer : marker type> =

<integer: 0> [DOT]

<integer: 1> [PLUS SIGN]

<integer: 2> [ASTERISK]

<integer: 3> [CIRCLE]

<integer: 4> [DIAGONAL CROSS]

<integer: >4> [reserved]

<integer: negative> [private line type]

C.8.2.2.2.9 SET MARKER SIZE SCALE FACTOR

<SET MARKER SIZE SCALE FACTOR opcode: 3/1, 2/13>

<real : marker size scale factor>

C.8.2.2.2.10 SET POLYMARKER COLOUR INDEX

<SET POLYMARKER COLOUR INDEX opcode: 3/1, 2/11>

<colour index : polymarker colour index>

C.8.2.2.2.11 SET FILL AREA REPRESENTATION

<SET FILL AREA REPRESENTATION opcode: 3/1, 3/12>

<identifier : workstation identifier>

<index : fill area index>

<enumerated : fill area interior style>

<integer : fill area style index>

[interior style HATCH]

<index : fill area style index>

[interior style PATTERN]

<colour index : fill area colour index>

with

<enumerated : fill area interior style> =

<enumerated: 0> [HOLLOW]

<enumerated: 1> [SOLID]

<enumerated: 2> [PATTERN]

<enumerated: 3> [HATCH]

<enumerated: >3> [reserved]

<integer : fill area style index> =

[interior style HATCH]

<integer: 0> [vertical lines]

<integer: 1> [horizontal lines]

<integer: 2> [45 degree lines]

<integer: 3> [−45 degree lines]

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

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

<integer: >5> [reserved]

<integer: negative> [private hatch style]

C.8.2.2.2.12 SET FILL AREA INDEX

<SET FILL AREA INDEX opcode: 3/1, 2/10>

<index : fill area index>

C.8.2.2.2.13 SET FILL AREA INTERIOR STYLE

<SET FILL AREA INTERIOR STYLE opcode: 3/1, 2/5>

<enumerated : fill area interior style>

with

<enumerated: 0> [HOLLOW]

<enumerated: 1> [SOLID]

<enumerated: 2> [PATTERN]

<enumerated: 3> [HATCH]

<enumerated: >3> [reserved]

C.8.2.2.2.14 SET FILL AREA COLOUR INDEX

<SET FILL AREA COLOUR INDEX opcode: 3/1, 2/4>

<colour index : fill area colour index>

C.8.2.2.2.15 SET FILL AREA STYLE INDEX

<SET FILL AREA STYLE INDEX opcode: 3/1, 2/6>

<integer : fill area style index>
[interior style HATCH]

<index : fill area style index>
[interior style PATTERN]

with

<integer : fill area style index> =

<integer: 0> [vertical lines]

<integer: 1> [horizontal lines]

<integer: 2> [45 degree lines]

<integer: 3> [−45 degree lines]

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

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

<integer: >5> [reserved]

<integer: negative> [private hatch style]

C.8.2.2.2.16 SET PATTERN REPRESENTATION

<SET PATTERN REPRESENTATION opcode: 3/1, 3/15>

<identifier : workstation identifier>

<index : pattern index>

<integer : deltax> [DX]

<integer : deltay> [DY]

<colour index list : pattern cell colour index list>

C.8.2.2.2.17 SET PATTERN REFERENCE POINT

<SET PATTERN REFERENCE POINT opcode: 3/1, 2/9>

<point : reference point>

C.8.2.2.2.18 SET PATTERN VECTORS

<SET PATTERN VECTORS opcode: 3/1, 2/8>

<point : height vector>
<point : width vector>

C.8.2.2.2.19 SET TEXT REPRESENTATION

<SET TEXT REPRESENTATION opcode: 3/1, 3/14>

<identifier : workstation identifier>
<index : text index>
<integer : text font>
<enumerated : text precision>
<real : character expansion factor>
<real : character spacing>
<colour index : text colour index>

with

<enumerated : text precision> =
<enumerated: 0> [precision STRING]
|<enumerated: 1> [precision CHARACTER]
|<enumerated: 2> [precision STROKE]
|<enumerated: >2> [reserved]
<integer : text font> =
<integer: 0> [font 0]
|<integer: >0> [reserved]
|<integer: negative value> [private font]

C.8.2.2.2.20 SET TEXT INDEX

<SET TEXT INDEX opcode: 3/1, 3/6>

<index : text index>

C.8.2.2.2.21 SET TEXT FONT AND PRECISION

<SET TEXT FONT AND PRECISION opcode: 3/1, 3/4>

<integer : text font>
<enumerated : text precision>

with

<enumerated : text precision> =
<enumerated: 0> [precision STRING]
|<enumerated: 1> [precision CHARACTER]
|<enumerated: 2> [precision STROKE]
|<enumerated: >2> [reserved]
<integer : text font> =
<integer: 0> [font 0]
|<integer: >0> [reserved]
|<integer: negative value> [private font]

C.8.2.2.2.22 SET CHARACTER EXPANSION FACTOR

<SET CHARACTER EXPANSION FACTOR opcode: 3/1, 3/0>

<real : character expansion factor>

C.8.2.2.2.23 SET CHARACTER SPACING

<SET CHARACTER SPACING opcode: 3/1, 3/3>

<real : character spacing>

C.8.2.2.2.24 SET TEXT COLOUR INDEX

<SET TEXT COLOUR INDEX opcode: 3/1, 2/15>

<colour index : text colour index>

C.8.2.2.2.25 SET TEXT PATH

<SET TEXT PATH opcode: 3/1, 3/2>

<enumerated : text path>

with

<enumerated : text path> =

<enumerated: 0> [RIGHT]

<enumerated: 1> [LEFT]

<enumerated: 2> [UP]

<enumerated: 3> [DOWN]

<enumerated >3> [reserved]

C.8.2.2.2.26 SET CHARACTER VECTORS

<SET CHARACTER VECTORS opcode: 3/1, 3/1>

<point : height vector>

<point : width vector>

C.8.2.2.2.27 SET TEXT ALIGNMENT

<SET TEXT ALIGNMENT opcode: 3/1, 3/5>

<enumerated : horizontal alignment>

<enumerated : vertical alignment>

with

<enumerated : horizontal alignment> =

<enumerated: 0> [NORMAL]

<enumerated: 1> [LEFT]

<enumerated: 2> [CENTRE]

<enumerated: 3> [RIGHT]

<enumerated >3> [reserved]

<enumerated : vertical alignment> =

<enumerated: 0> [NORMAL]

<enumerated: 1> [TOP]

<enumerated: 2> [CAP]

<enumerated: 3> [HALF]

<enumerated: 4> [BASE]

<enumerated: 5> [BOTTOM]

<enumerated >5> [reserved]

C.8.2.2.2.28 SET COLOUR REPRESENTATION

<SET COLOUR REPRESENTATION opcode: 3/2, 3/8>

<identifier : workstation identifier>

<colour index : starting entry in the colour table>

<colour direct : colour data> (1)

C.8.2.2.2.29 SET ASPECT SOURCE FLAGS

<SET ASPECT SOURCE FLAGS opcode: 3/1, 2/7>

<enumerated:	line type ASF>
<enumerated:	line width scale factor ASF>
<enumerated:	polyline colour ASF>
<enumerated:	marker type ASF>
<enumerated:	marker size scale factor ASF>
<enumerated:	polymarker colour ASF>
<enumerated:	text font and precision ASF>
<enumerated:	character expansion factor ASF>
<enumerated:	character spacing ASF>
<enumerated:	text colour ASF>
<enumerated:	fill area interior style ASF>
<enumerated:	fill area style index ASF>
<enumerated:	fill area colour ASF>

with

<enumerated	: <attribute> ASF> =
<enumerated: 0>	[INDIVIDUAL]
<enumerated: 1>	[BUNDLED]
<enumerated >1>	[reserved]

C.8.2.2.3 Transformation primitives

C.8.2.2.3.1 SET WORKSTATION WINDOW

<SET WORKSTATION WINDOW opcode: 3/2, 2/1>

<identifier	: workstation identifier>
<point	: window limits> (=2)

C.8.2.2.3.2 SET WORKSTATION VIEWPORT

<SET WORKSTATION VIEWPORT opcode: 3/2, 2/2>

<identifier	: workstation identifier>
<real	: viewport limits> (=4)

C.8.2.2.4 Clipping primitives

C.8.2.2.4.1 SET CLIPPING RECTANGLE

<SET CLIPPING RECTANGLE opcode: 3/2, 2/3>

<point	: clipping rectangle limits> (=2)
--------	-----------------------------------

C.8.2.2.5 Control primitives

C.8.2.2.5.1 UPDATE WORKSTATION

<UPDATE WORKSTATION opcode: 3/2, 3/4>

<identifier	: workstation identifier>
<enumerated	: update regeneration flag>

with

<enumerated	: update regeneration flag > =
<enumerated: 0>	[PERFORM]
<enumerated: 1>	[POSTPONE]
<enumerated >1>	[reserved]

C.8.2.2.5.2 SET DEFERRAL STATE

<SET DEFERRAL STATE opcode: 3/2, 3/5>

<identifier : workstation identifier>
<enumerated : deferral mode>
<enumerated : implicit regeneration flag>

with

<enumerated : deferral mode> =
<enumerated: 0> [ASAP]
|<enumerated: 1> [BNIL]
|<enumerated: 2> [BNIG]
|<enumerated: 3> [ASTI]
|<enumerated >3> [reserved]
<enumerated : implicit regeneration flag> =
<enumerated: 0> [SUPPRESSED]
|<enumerated: 1> [ALLOWED]
|<enumerated: >1> [reserved]

C.8.2.2.5.3 EMERGENCY CLOSE

<EMERGENCY CLOSE opcode: 3/2, 2/6>

C.8.2.3 Segment related primitives

C.8.2.3.1 WDSS related primitives

C.8.2.3.1.1 CREATE SEGMENT

<CREATE SEGMENT opcode: 3/3, 2/0>

<identifier : segment name>

C.8.2.3.1.2 CLOSE SEGMENT

<CLOSE SEGMENT opcode: 3/3, 2/1>

C.8.2.3.1.3 RENAME SEGMENT

<RENAME SEGMENT opcode: 3/3, 2/2>

<identifier : old segment name>
<identifier : new segment name>

C.8.2.3.1.4 DELETE SEGMENT FROM WORKSTATION

<DELETE SEGMENT FROM WORKSTATION opcode: 3/3, 2/3>

<identifier : workstation identifier>
<identifier : segment name>

C.8.2.3.1.5 DELETE SEGMENT

<DELETE SEGMENT opcode: 3/3, 2/8>

<identifier : segment name>

C.8.2.3.1.6 REDRAW ALL SEGMENTS ON WORKSTATION

<REDRAW ALL SEGMENTS ON WORKSTATION opcode: 3/3, 2/9>

<identifier : workstation identifier>

C.8.2.3.1.7 SET HIGHLIGHTING

<SET HIGHLIGHTING opcode: 3/3, 2/6>

<identifier : segment name>
<enumerated : highlighting flag>

with

<enumerated : highlighting flag> =
<enumerated: 0> [NOTHIGHLIGHTED]
|<enumerated: 1> [HIGHLIGHTED]
|<enumerated: >1> [reserved]

C.8.2.3.1.8 SET VISIBILITY

<SET VISIBILITY opcode: 3/3, 2/7>

<identifier : segment name>
<enumerated : visibility flag>

with

<enumerated : visibility flag> =
<enumerated: 0> [VISIBLE]
|<enumerated: 1> [INVISIBLE]
|<enumerated: >1> [reserved]

C.8.2.3.1.9 SET SEGMENT TRANSFORMATION

<SET SEGMENT TRANSFORMATION opcode: 3/3, 2/5>

<identifier : segment name>
<matrix : transformation matrix>

C.8.2.3.1.10 SET SEGMENT PRIORITY

<SET SEGMENT PRIORITY opcode: 3/3, 2/12>

<identifier : segment name>
<real : segment priority>

C.8.2.3.2 WISS related primitives

C.8.2.3.2.1 ASSOCIATE SEGMENT WITH WORKSTATION

<ASSOCIATE SEGMENT WITH WORKSTATION opcode: 3/3, 2/10>

<identifier : workstation identifier>
<identifier : segment name>

C.8.2.3.2.2 COPY SEGMENT TO WORKSTATION

<COPY SEGMENT TO WORKSTATION opcode: 3/3, 2/11>

<identifier : workstation identifier>
<identifier : segment name>

C.8.2.3.2.3 INSERT SEGMENT

<INSERT SEGMENT opcode: 3/3, 2/4>

<identifier : segment name>
<matrix : transformation matrix>

C.8.2.4 Input primitives

A Videotex environment will contain no workstations of the type INPUT, so this subclause is not applicable for Videotex purposes.

C.8.2.5 Inquire primitives

A Videotex environment will support no inquire primitives, so this subclause is not applicable for Videotex purposes.

C.8.2.6 Protocol descriptor primitives

C.8.2.6.1 SET DOMAIN RING

<SET DOMAIN RING opcode: 3/2, 2/4>

<enumerated : angular resolution factor>
<integer : basic Radius>

with

<enumerated : angular resolution factor> =
<enumerated: 0> [resolution 0]
|<enumerated: 1> [resolution 1]
|<enumerated: 2> [resolution 2]
|<enumerated: 3> [resolution 3]
|<enumerated: >3> [reserved]

C.8.2.6.2 SET COLOUR HEADER

<SET COLOUR HEADER opcode: 3/2, 2/8>

<integer : unit resolution>
<enumerated : coding method>

with

<enumerated : coding method> =
<enumerated: 1> [RGB-coding]
|<enumerated: >1> [reserved]

C.8.2.6.3 SET COORDINATE PRECISION

<SET COORDINATE PRECISION opcode: 3/2, 2/9>

<integer : magnitude code>
<integer : granularity coding>
<integer : default exponent>
<enumerated : explicit exponent allowed>

with

<enumerated: 0> [ALLOWED]
|<enumerated: 1> [FORBIDDEN]

C.8.2.6.4 SET REAL PRECISION

<SET REAL PRECISION opcode: 3/2, 3/9>

<integer : magnitude code>
<integer : granularity coding>
<integer : default exponent>
<enumerated : explicit exponent allowed>

with

<enumerated: 0> [ALLOWED]
|<enumerated: 1> [FORBIDDEN]

C.8.2.6.5 SET COLOUR INDEX PRECISION

<SET COLOUR INDEX PRECISION opcode: 3/2, 2/10>

<integer : : number of bits>

C.9 Defaults

The defaults in this clause define the values of parameters to be taken if a particular parameter value is out of range or not yet defined. The purpose of these default values is to be able to process primitives even if some parameter values are not yet defined.

<i>Parameter</i>	<i>Value</i>
ASPECT SOURCE FLAGS (ALL)	INDIVIDUAL
CHARACTER EXPANSION FACTOR	1.0
CHARACTER SPACING	0.0
CHARACTER VECTORS	Implementation dependent
CLIPPING RECTANGLE	(0.0, 0.0) and (1.0, 1.0)
DEFERRAL MODE	ASAP
DOMAIN RING	
Angular resolution factor	RESOLUTION_0
Basic Radius	2**max (0, -8-current granularity code)
FILL AREA BUNDLE TABLE	The entry of the FILL AREA BUNDLE TABLE corresponding to FILL AREA INDEX 1 contains the default values of FILL AREA INTERIOR STYLE, FILL AREA STYLE INDEX and FILL AREA COLOUR INDEX. Non-defined entries use the attributes as specified by FILL AREA INDEX 1.
FILL AREA COLOUR INDEX	1
FILL AREA INDEX	1
FILL AREA INTERIOR STYLE	HOLLOW
FILL AREA STYLE INDEX	1
HIGHLIGHTING FLAG	NOTHIGHLIGHTED
IMPLICIT REGENERATION FLAG	ALLOWED
LINE TYPE	SOLID
LINE WIDTH SCALE FACTOR	1.0
MARKER SIZE SCALE FACTOR	1.0
MARKER TYPE	ASTERISK
PATTERN REFERENCE POINT	(0.0, 0.0)
PATTERN TABLE	
Deltax	1
Deltay	1
Pattern cell colour index list	1
PATTERN VECTORS	(0.0, 1.0) and (1.0, 0.0)
POLYLINE BUNDLE TABLE	The entry of the POLYLINE BUNDLE TABLE corresponding to POLYLINE INDEX 1 contains the default values of LINE TYPE, LINE WIDTH SCALE FACTOR and POLYLINE COLOUR INDEX. Non-defined entries use the attributes as specified by POLYLINE INDEX 1.
POLYLINE COLOUR INDEX	1
POLYLINE INDEX	1
POLYMARKER BUNDLE TABLE	The entry of the POLYMARKER BUNDLE TABLE corresponding to POLYMARKER INDEX 1 contains the default values of MARKER TYPE, MARKER SIZE SCALE

FACTOR and POLYMARKER COLOUR INDEX. Non-defined entries use the attributes as specified by POLYMARKER INDEX 1.

POLYMARKER COLOUR INDEX	1
POLYMARKER INDEX	1
SEGMENT PRIORITY	0.0
SET COORDINATE PRECISION	
exponent	−9
granularity code	−9
magnitude code	4
explicit exponent allowed	FORBIDDEN
SET REAL PRECISION	
exponent	−9
granularity code	−9
magnitude code	4
explicit exponent allowed	FORBIDDEN
SET COLOUR INDEX PRECISION	
number of bits	5
STARTING ENTRY IN THE COLOUR TABLE	1
TEXT ALIGNMENT	NORMAL, NORMAL
TEXT BUNDLE TABLE	The entry of the TEXT BUNDLE TABLE corresponding to TEXT INDEX 1 contains the default values of TEXT FONT AND PRECISION, CHARACTER EXPANSION FACTOR, CHARACTER SPACING and TEXT COLOUR INDEX. Non-defined entries use the attributes as specified by TEXT INDEX 1.
TEXT COLOUR INDEX	1
TEXT FONT AND PRECISION *)	0, STRING
TEXT INDEX	1
TEXT PATH	RIGHT
TRANSFORMATION MATRIX	1.0 0.0 0.0 0.0 1.0 0.0
UNIT RESOLUTION	4
UPDATE REGENERATION FLAG	POSTPONE
VISIBILITY FLAG	VISIBLE
WORKSTATION VIEWPORT *)	The largest square which fits into the display area (see Figure C.9).
WORKSTATION WINDOW *)	(0.0, 0.0) and (1.0, 1.0)

*) The default WORKSTATION WINDOW and WORKSTATION VIEWPORT set by the primitive SET DEFAULTS shall affect only the requested entries in the workstation state lists of all open workstations.

C.10 Conformance

Conformance to a standard means that all of its requirements are met.

GDS addresses those devices which are capable of processing the encoded graphics primitives defined in it. Such devices can vary from each other depending on the application for which they have been specifically designed. To take this into account, GDS provides a set of primitives grouped in levels in a manner consistent with the GKS levels.

Therefore, conformance to GDS means conformance to one of the levels defined in this Recommendation (see 5.10 of GKS DIS 7942).

To be in conformance to one level, for a device, means to offer at least all the functionalities pertaining to this level.

Appendix I

(to Part 2 of Annex C to Recommendation T.101)

Primitives, Workstation Categories, Levels and Options

Table I.1 lists the primitives contained in this Recommendation.

For each primitive is indicated:

- The applicability to workstation categories;
- The level in which it appears;
- The state (mandatory or optional).

The following notational conventions are used:

SS Workstation independent segment storage

O Workstation of category OUTPUT

SN SS is fundamental to the primitive, but the workstation identifier parameter cannot be the one of the SS

M Mandatory

N Non-mandatory

Table I.1/T.101 – Primitives, Workstation categories, Levels and options

Primitives	Level	Applies to		Option
OPEN WORKSTATION	L0a	SS	O	M
CLOSE WORKSTATION	L0a	SS	O	M
ACTIVATE WORKSTATION	L0a	SS	O	M
DEACTIVATE WORKSTATION	L0a	SS	O	M
CLEAR WORKSTATION	L0a	SS	O	M
SET DEFAULTS	L0a	SS	O	M
GDS EXCAPE1	L0a	SS	O	M
POLYLINE	L0a	SS	O	M
POLYMARKER	L0a	SS	O	M
TEXT	L0a	SS	O	M
FILL AREA	L0a	SS	O	M
CELL ARRAY	L0a	SS	O	M
GENERALIZED DRAWING PRIMITIVE:	L0a	SS	O	M
RECTANGLE	L0a	SS	O	N
CIRCLE	L0a	SS	O	N
CIRCULAR ARC 3 POINT	L0a	SS	O	N
CIRCULAR ARC 3 POINT CHORD	L0a	SS	O	N
CIRCULAR ARC 3 POINT PIE	L0a	SS	O	N
CIRCULAR ARC CENTRE	L0a	SS	O	N
CIRCULAR ARC CENTRE CHORD	L0a	SS	O	N
CIRCULAR ARC CENTRE PIE	L0a	SS	O	N
ELLIPSE	L0a	SS	O	N
ELLIPTIC ARC	L0a	SS	O	N
ELLIPTIC ARC CHORD	L0a	SS	O	N
ELLIPTIC ARC PIE	L0a	SS	O	N
SPLINE	L0a	SS	O	N
SET POLYLINE INDEX	L0a	SS	O	M
SET POLYLINE REPRESENTATION	L1a		O	M
SET LINE TYPE	L0a	SS	O	M
SET LINE WIDTH SCALE FACTOR	L0a	SS	O	M

Table I.1/T.101 – Primitives, Workstation categories, Levels and options (fin)

Primitives	Level	Applies to		Option
SET POLYLINE COLOUR INDEX	L0a	SS	O	M
SET POLYMARKER INDEX	L0a	SS	O	M
SET POLYMARKER REPRESENTATION	L1a		O	M
SET MARKER TYPE	L0a	SS	O	M
SET MARKER SIZE SCALE FACTOR	L0a	SS	O	M
SET POLYMARKER COLOUR INDEX	L0a	SS	O	M
SET FILL AREA INDEX	L0a	SS	O	M
SET FILL AREA REPRESENTATION	L1a		O	M
SET FILL AREA INTERIOR STYLE	L0a	SS	O	M
SET FILL AREA STYLE INDEX	L0a	SS	O	M
SET FILL AREA COLOUR INDEX	L0a	SS	O	M
SET PATTERN REPRESENTATION	L1a		O	N
SET PATTERN REFERENCE POINT	L0a	SS	O	N
SET PATTERN VECTORS	L0a	SS	O	N
SET TEXT INDEX	L0a	SS	O	M
SET TEXT REPRESENTATION	L1a		O	M
SET TEXT FONT AND PRECISION	L0a	SS	O	M
SET CHARACTER EXPANSION FACTOR	L0a	SS	O	M
SET CHARACTER SPACING	L0a	SS	O	M
SET TEXT COLOUR INDEX	L0a	SS	O	M
SET TEXT PATH	L0a	SS	O	M
SET CHARACTER VECTORS	L0a	SS	O	M
SET TEXT ALIGNMENT	L0a	SS	O	M
SET COLOUR REPRESENTATION	L0a		O	M
SET ASPECT SOURCE FLAGS	L0a	SS	O	M
SET WORKSTATION WINDOW	L0a		O	M
SET WORKSTATION VIEWPORT	L0a		O	M
SET CLIPPING RECTANGLE	L0a	SS	O	M
UPDATE WORKSTATION	L0a		O	M
SET DEFERRAL STATE	L1a		O	M
EMERGENCY CLOSE	L0a	SS	O	M
CREATE SEGMENT	L1a	SS	O	M
CLOSE SEGMENT	L1a	SS	O	M
RENAME SEGMENT	L1a	SS	O	M
DELETE SEGMENT FROM WORKSTATION	L1a	SS	O	M
DELETE SEGMENT	L1a	SS	O	M
REDRAW ALL SEGMENTS ON WORKSTATION	L1a		O	M
SET HIGHLIGHTING	L1a	SS	O	M
SET VISIBILITY	L1a	SS	O	M
SET SEGMENT TRANSFORMATION	L1a	SS	O	M
SET SEGMENT PRIORITY	L1a	SS	O	N
ASSOCIATE SEGMENT WITH WORKSTATION	L2a	SS	O	M
COPY SEGMENT TO WORKSTATION	L2a	SN	O	M
INSERT SEGMENT	L2a	SS	O	M
SET COLOUR HEADER	L0a	SS	O	N
SET DOMAIN RING	L0a	SS	O	M
SET COORDINATE PRECISION	L0a	SS	O	M
SET REAL PRECISION	L0a	SS	O	M
SET COLOUR INDEX PRECISION	L0a	SS	O	M

Appendix II

(to Part 2 of Annex C to Recommendation T.101)

B-spline curves and ellipses

II.1 Short note on B-spline curves

A spline is a piecewise polynomial function passing through a set of points called knots (see Figure II.1).

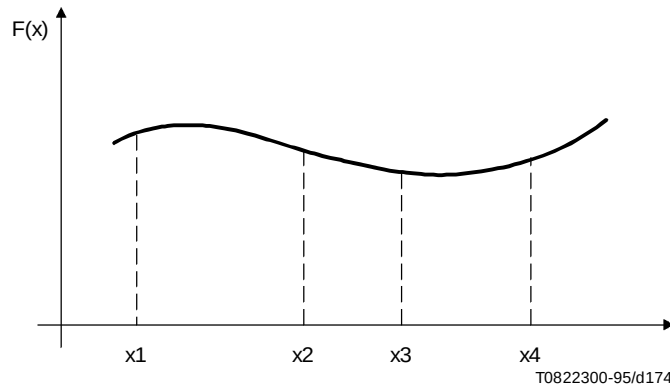


Figure II.1/T.101 – A spline passing through 4 knots

The values $x_1, x_2, \dots, x_k$ are called breakpoints. Between each breakpoint, $f(x)$ is a degree- m polynomial and the j -first derivatives are continuous at each breakpoint. In most applications, polynomials of degree 2 or 3 with $j = 1$ or 2 respectively are sufficient.

Each (x_i, x_{i+1}) defines a sub-interval. B-splines are splines that are zero at all sub-intervals except $m + 1$ of them, where m is the degree of the polynomials. In most cases *uniform* B-splines are used, that are B-splines for which the breakpoints are equally spaced. In the two-dimensional space, a B-spline curve is defined as:

$$P(t) = \sum_{i=1}^n P_i N_{i,m}(t)$$

where $P(t)$ is a point on the curve, points P_i are called guiding points and $N_{i,m}$ is a m -degree B-spline.

For a uniform quadratic B-spline, between two knots, we have:

$$P(t) = \left(\frac{P_{i+2} + P_i}{2} - P_{i+1} \right) t^2 + (P_{i+1} - P_i) t + \frac{1}{2} (P_{i+1} + P_i)$$

The corresponding knots are:

$$\frac{P_i + P_{i+1}}{2} \quad \text{et} \quad \frac{P_{i+1} + P_{i+2}}{2}$$

An example of such a curve is given in Figure II.2. Knots are located on the middle of the segments joining the breakpoints and the curve is tangent to the segment at this point.

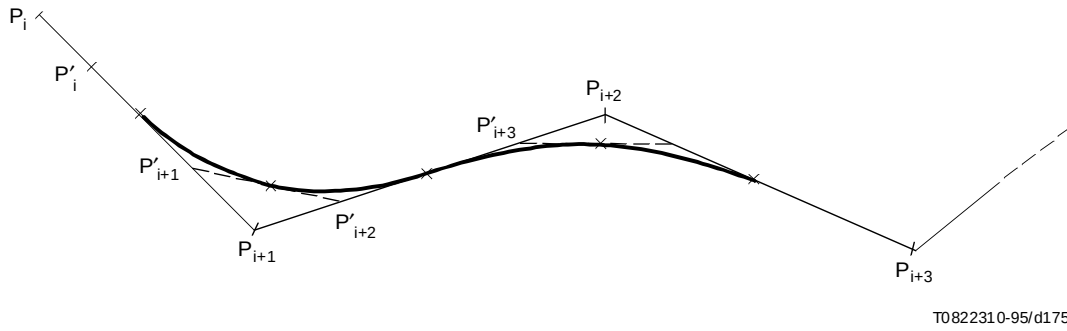


Figure II.2/T.101 – Uniform quadratic B-spline curve

Such spline curves can easily be generated using the sub-division or refinement properties of B-splines [1].

We can apply this theory to quadratic B-splines by replacing P_i , P_{i+1} and P_{i+2} by the set of four points: P'_i , P'_{i+1} , P'_{i+2} , P'_{i+3}

$$P'_i = \frac{1}{2} \left(P_i + \frac{P_i + P_{i+1}}{2} \right)$$

$$P'_{i+1} = \frac{1}{2} \left(\frac{P_{i+1} + P_i}{2} + P_{i+1} \right)$$

$$P'_{i+2} = \frac{1}{2} \left(P_{i+1} + \frac{P_{i+1} + P_{i+2}}{2} \right)$$

$$P'_{i+3} = \frac{1}{2} \left(\frac{P_{i+1} + P_{i+2}}{2} + P_{i+2} \right)$$

The new guiding points will produce the same curve as the former ones but they introduce a supplementary knot:

$$\frac{P_{i+1} + P_{i+2}}{2}$$

Thus, the original curve segment has been divided into two parts. Furthermore the new guiding points are closer to the curve than the former ones (see Figure II-2).

By simply repeating this procedure, until the curve segments reach the size of a pixel, the spline curve can be drawn. Only very simple integer arithmetic is needed at each sub-division step (addition and shift). An algorithm of this type is given in [2]. Note that in this algorithm, the given end-points of the curve are no guiding points but knots.

The coordinates as specified in the GDP (spline) – primitive will be considered as guiding points of a uniform quadratic B-spline curve. The curve can, thus, easily be generated using the above mentioned sub-division technique.

- [1] LANE (J. M.), RIESENFELD (R. F.): A theoretical Development for the Computer Generation and Display of Piecewise Polynomial Surfaces, *I.E.E.E. Trans. on P.A.M.I.*, Vol. PAMI – 2, n° 1, pp 35-46, January 1980.
- [2] CHAIKIN (G. M.): An algorithm for High-Speed Curve Generation, *Computer Graphics and Image Processing*, Vol. 3, pp 346-349, 1974.

II.2 Ellipse primitives

A central question is how to represent a generally oriented ellipse such that the necessary properties of the picture are preserved across all graphical transformations. Unfortunately, the major and minor axes cannot be used for an ellipse in a general orientation since, as shown in Figure II.3 below, these axes do not remain mutually orthogonal (do not remain axes) across a scaling transformation which does not preserve aspect ratio.

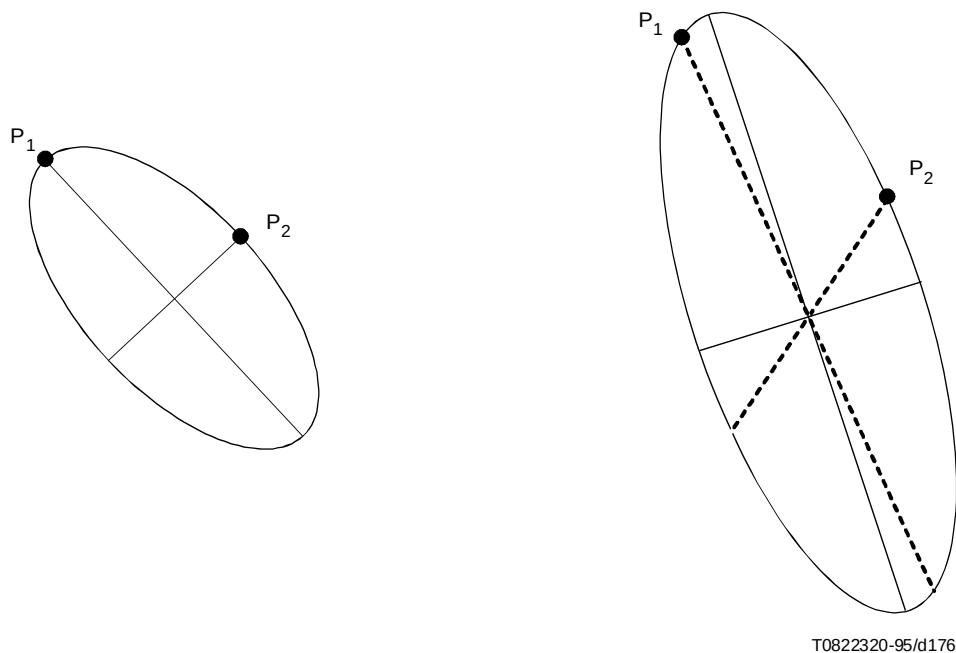


Figure II.3/T.101 – The scaling of an ellipse and its axes such that $X' = 2X$ and $Y' = Y$

The problem can be solved by utilizing the fact that any Conjugate Diameter Pair (CDP) of the ellipse remains a CDP across any graphical transformation.

A CDP is a pair D, d of diameters of the ellipse such that a tangent to the ellipse at each endpoint of a diameter is parallel to the other diameter. Note that the four tangents to the ellipse at the endpoints of the CDP form a parallelogram whose sides are bisected by the endpoints.

This is demonstrated below in Figure II.4 in which the ellipse has been scaled by a factor of two in the Y-direction only.

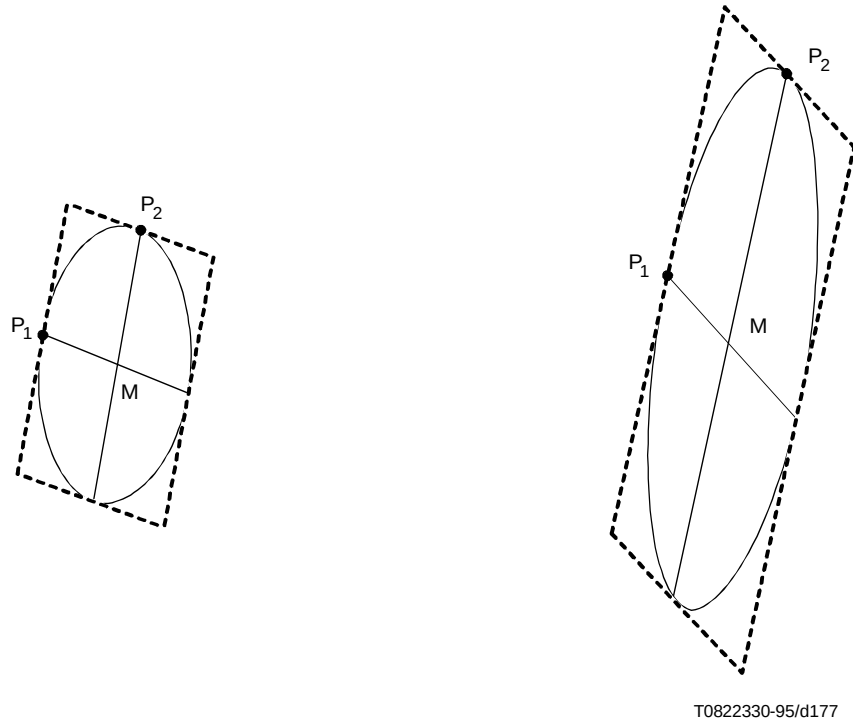


Figure II.4/T.101 – Ellipses defined by a Conjugate Diameter Pair

Thus, any CDP can be used to represent an ellipse. Note that the (mutually perpendicular) major and minor axes of an ellipse, and any pair of perpendicular diameters of a circle are CDPs, although they do not remain perpendicular across a transformation.

To thus represent an ellipse, we need only three points:

- the centre point M (X_m , Y_m) of the ellipse;
- two endpoints P1 (X_1 , Y_1) and P2 (X_2 , Y_2) of a CDP.

The CDP vector components, relative to the centre point, are defined as follows:

$$\begin{aligned} DX1 &= X1 - X_m \\ DY1 &= Y1 - Y_m \\ DX2 &= X2 - X_m \\ DY2 &= Y2 - Y_m \end{aligned}$$

The CDP vector components are the coefficients of the parametric equations:

$$\begin{aligned} X &= X_m + DX1 * \cos(t) + DX2 * \sin(t) \\ Y &= Y_m + DY1 * \cos(t) + DY2 * \sin(t) \end{aligned}$$

in which t runs from 0 to $2 * \pi$.

Appendix III

(to Part 2 of Annex C to Recommendation T.101)

Cross references

III.1 Functional order

Primitives

Subclause number

Description Encoding

WORKSTATION MANAGEMENT PRIMITIVES

OPEN WORKSTATION	C.6.2.1.1	C.8.2.1.1
CLOSE WORKSTATION	C.6.2.1.2	C.8.2.1.2
ACTIVATE WORKSTATION	C.6.2.1.3	C.8.2.1.3
DEACTIVATE WORKSTATION	C.6.2.1.4	C.8.2.1.4
CLEAR WORKSTATION	C.6.2.1.5	C.8.2.1.5
SET DEFAULTS	C.6.2.1.6	C.8.2.1.6
GDS EXCAPE1	C.6.2.1.7	C.8.2.1.7

OUTPUT DRAWING PRIMITIVES

POLYLINE	C.6.2.2.1.1	C.8.2.2.1.1
POLYMARKER	C.6.2.2.1.2	C.8.2.2.1.2
FILL AREA	C.6.2.2.1.3	C.8.2.2.1.3
TEXT	C.6.2.2.1.4	C.8.2.2.1.4
CELL ARRAY	C.6.2.2.1.5	C.8.2.2.1.5
GDP	C.6.2.2.1.6	C.8.2.2.1.6

OUTPUT PRIMITIVES RELATED TO DISPLAY ELEMENT ATTRIBUTES

SET POLYLINE REPRESENTATION	C.6.2.2.2.1	C.8.2.2.2.1
SET POLYLINE INDEX	C.6.2.2.2.2	C.8.2.2.2.2
SET LINE TYPE	C.6.2.2.2.3	C.8.2.2.2.3
SET LINE WIDTH SCALE FACTOR	C.6.2.2.2.4	C.8.2.2.2.4
SET POLYLINE COLOUR INDEX	C.6.2.2.2.5	C.8.2.2.2.5
SET POLYMARKER REPRESENTATION	C.6.2.2.2.6	C.8.2.2.2.6
SET POLYMARKER INDEX	C.6.2.2.2.7	C.8.2.2.2.7
SET MARKER TYPE	C.6.2.2.2.8	C.8.2.2.2.8
SET MARKER SIZE SCALE FACTOR	C.6.2.2.2.9	C.8.2.2.2.9
SET POLYMARKER COLOUR INDEX	C.6.2.2.2.10	C.8.2.2.2.10
SET FILL AREA REPRESENTATION	C.6.2.2.2.11	C.8.2.2.2.11
SET FILL AREA INDEX	C.6.2.2.2.12	C.8.2.2.2.12
SET FILL AREA INTERIOR STYLE	C.6.2.2.2.13	C.8.2.2.2.13
SET FILL AREA COLOUR INDEX	C.6.2.2.2.14	C.8.2.2.2.14
SET FILL AREA STYLE INDEX	C.6.2.2.2.15	C.8.2.2.2.15
SET PATTERN REPRESENTATION	C.6.2.2.2.16	C.8.2.2.2.16
SET PATTERN REFERENCE POINT	C.6.2.2.2.17	C.8.2.2.2.17
SET PATTERN VECTORS	C.6.2.2.2.18	C.8.2.2.2.18
SET TEXT REPRESENTATION	C.6.2.2.2.19	C.8.2.2.2.19
SET TEXT INDEX	C.6.2.2.2.20	C.8.2.2.2.20
SET TEXT FONT AND PRECISION	C.6.2.2.2.21	C.8.2.2.2.21
SET CHARACTER EXPANSION FACTOR	C.6.2.2.2.22	C.8.2.2.2.22
SET CHARACTER SPACING	C.6.2.2.2.23	C.8.2.2.2.23

SET TEXT COLOUR INDEX	C.6.2.2.2.24	C.8.2.2.2.24
SET TEXT PATH	C.6.2.2.2.25	C.8.2.2.2.25
SET CHARACTER VECTORS	C.6.2.2.2.26	C.8.2.2.2.26
SET TEXT ALIGNMENT	C.6.2.2.2.27	C.8.2.2.2.27
SET COLOUR REPRESENTATION	C.6.2.2.2.28	C.8.2.2.2.28
SET ASPECT SOURCE FLAGS	C.6.2.2.2.29	C.8.2.2.2.29
<i>TRANSFORMATION PRIMITIVES</i>		
SET WORKSTATION WINDOW	C.6.2.2.3.1	C.8.2.2.3.1
SET WORKSTATION VIEWPORT	C.6.2.2.3.2	C.8.2.2.3.2
<i>CLIPPING PRIMITIVES</i>		
SET CLIPPING RECTANGLE	C.6.2.2.4.1	C.8.2.2.4.1
<i>CONTROL PRIMITIVES</i>		
UPDATE WORKSTATION	C.6.2.2.5.1	C.8.2.2.5.1
SET DEFERRAL STATE	C.6.2.2.5.2	C.8.2.2.5.2
EMERGENCY CLOSE	C.6.2.2.5.3	C.8.2.2.5.3
<i>SEGMENT RELATED PRIMITIVES</i>		
CREATE SEGMENT	C.6.2.3.1.1	C.8.2.3.1.1
CLOSE SEGMENT	C.6.2.3.1.2	C.8.2.3.1.2
RENAME SEGMENT	C.6.2.3.1.3	C.8.2.3.1.3
DELETE SEGMENT FROM WORKSTATION	C.6.2.3.1.4	C.8.2.3.1.4
DELETE SEGMENT	C.6.2.3.1.5	C.8.2.3.1.5
REDRAW ALL SEGMENTS ON WORKSTATION	C.6.2.3.1.6	C.8.2.3.1.6
SET HIGHLIGHTING	C.6.2.3.1.7	C.8.2.3.1.7
SET VISIBILITY	C.6.2.3.1.8	C.8.2.3.1.8
SET SEGMENT TRANSFORMATION	C.6.2.3.1.9	C.8.2.3.1.9
SET SEGMENT PRIORITY	C.6.2.3.1.10	C.8.2.3.1.10
ASSOCIATE SEGMENT WITH WORKSTATION	C.6.2.3.2.1	C.8.2.3.2.1
COPY SEGMENT TO WORKSTATION	C.6.2.3.2.2	C.8.2.3.2.2
INSERT SEGMENT	C.6.2.3.2.3	C.8.2.3.2.3
<i>PROTOCOL DESCRIPTOR PRIMITIVES</i>		
SET DOMAIN RING	C.6.2.6.1	C.8.2.6.1
SET COLOUR HEADER	C.6.2.6.2	C.8.2.6.2
SET COORDINATE PRECISION	C.6.2.6.3	C.8.2.6.3
SET REAL PRECISION	C.6.2.6.4	C.8.2.6.4
SET COLOUR INDEX PRECISION	C.6.2.6.5	C.8.2.6.5

III.2 Alphabetic order

Primitives

Subclause number

	<i>Description</i>	<i>Encoding</i>
ACTIVATE WORKSTATION	C.6.2.1.3	C.8.2.1.3
ASSOCIATE SEGMENT WITH WORKSTATION	C.6.2.3.2.1	C.8.2.3.2.1
CELL ARRAY	C.6.2.2.1.5	C.8.2.2.1.5
CLEAR WORKSTATION	C.6.2.1.5	C.8.2.1.5
CLOSE SEGMENT	C.6.2.3.1.2	C.8.2.3.1.2

CLOSE WORKSTATION	C.6.2.1.2	C.8.2.1.2
COPY SEGMENT TO WORKSTATION	C.6.2.3.2.2	C.8.2.3.2.2
CREATE SEGMENT	C.6.2.3.1.1	C.8.2.3.1.1
DEACTIVATE WORKSTATION	C.6.2.1.4	C.8.2.1.4
DELETE SEGMENT	C.6.2.3.1.5	C.8.2.3.1.5
DELETE SEGMENT FROM WORKSTATION	C.6.2.3.1.4	C.8.2.3.1.4
EMERGENCY CLOSE	C.6.2.2.5.3	C.8.2.2.5.3
FILL AREA	C.6.2.2.1.3	C.8.2.2.1.3
GDP	C.6.2.2.1.6	C.8.2.2.1.6
GDS ESCAPE1	C.6.2.1.7	C.8.2.1.7
INSERT SEGMENT	C.6.2.3.2.3	C.8.2.3.2.3
OPEN WORKSTATION	C.6.2.1.1	C.8.2.1.1
POLYLINE	C.6.2.2.1.1	C.8.2.2.1.1
POLYMARKER	C.6.2.2.1.2	C.8.2.2.1.2
REDRAW ALL SEGMENTS ON WORKSTATION	C.6.2.3.1.6	C.8.2.3.1.6
RENAME SEGMENT	C.6.2.3.1.3	C.8.2.3.1.3
SET ASPECT SOURCE FLAGS	C.6.2.2.2.29	C.8.2.2.2.29
SET CLIPPING RECTANGLE	C.6.2.2.4.1	C.8.2.2.4.1
SET CHARACTER EXPANSION FACTOR	C.6.2.2.2.22	C.8.2.2.2.22
SET CHARACTER SPACING	C.6.2.2.2.23	C.8.2.2.2.23
SET CHARACTER VECTORS	C.6.2.2.2.26	C.8.2.2.2.26
SET COLOUR HEADER	C.6.2.6.2	C.8.2.6.2
SET COLOUR REPRESENTATION	C.6.2.2.2.28	C.8.2.2.2.28
SET DEFAULTS	C.6.2.1.6	C.8.2.1.6
SET DEFERRAL STATE	C.6.2.2.5.2	C.8.2.2.5.2
SET DOMAIN RING	C.6.2.6.1	C.8.2.6.1
SET FILL AREA COLOUR INDEX	C.6.2.2.2.14	C.8.2.2.2.14
SET FILL AREA INDEX	C.6.2.2.2.12	C.8.2.2.2.12
SET FILL AREA INTERIOR STYLE	C.6.2.2.2.13	C.8.2.2.2.13
SET FILL AREA REPRESENTATION	C.6.2.2.2.11	C.8.2.2.2.11
SET FILL AREA STYLE INDEX	C.6.2.2.2.15	C.8.2.2.2.15
SET HIGHLIGHTING	C.6.2.3.1.7	C.8.2.3.1.7
SET LINE TYPE	C.6.2.2.2.3	C.8.2.2.2.3
SET LINE WIDTH SCALE FACTOR	C.6.2.2.2.4	C.8.2.2.2.4
SET MARKER SIZE SCALE FACTOR	C.6.2.2.2.9	C.8.2.2.2.9
SET MARKER TYPE	C.6.2.2.2.8	C.8.2.2.2.8
SET PATTERN REFERENCE POINT	C.6.2.2.2.17	C.8.2.2.2.17
SET PATTERN REPRESENTATION	C.6.2.2.2.16	C.8.2.2.2.16
SET PATTERN VECTORS	C.6.2.2.2.18	C.8.2.2.2.18
SET POLYLINE COLOUR INDEX	C.6.2.2.2.5	C.8.2.2.2.5
SET POLYLINE INDEX	C.6.2.2.2.2	C.8.2.2.2.2
SET POLYLINE REPRESENTATION	C.6.2.2.2.1	C.8.2.2.2.1
SET POLYMARKER COLOUR INDEX	C.6.2.2.2.10	C.8.2.2.2.10
SET POLYMARKER INDEX	C.6.2.2.2.7	C.8.2.2.2.7
SET POLYMARKER REPRESENTATION	C.6.2.2.2.6	C.8.2.2.2.6
SET SEGMENT PRIORITY	C.6.2.3.1.10	C.8.2.3.1.10

SET SEGMENT TRANSFORMATION	C.6.2.3.1.9	C.8.2.3.1.9
SET TEXT ALIGNMENT	C.6.2.2.2.27	C.8.2.2.2.27
SET TEXT COLOUR INDEX	C.6.2.2.2.24	C.8.2.2.2.24
SET TEXT FONT AND PRECISION	C.6.2.2.2.21	C.8.2.2.2.21
SET TEXT INDEX	C.6.2.2.2.20	C.8.2.2.2.20
SET TEXT PATH	C.6.2.2.2.25	C.8.2.2.2.25
SET TEXT REPRESENTATION	C.6.2.2.2.19	C.8.2.2.2.19
SET VISIBILITY	C.6.2.3.1.8	C.8.2.3.1.8
SET WORKSTATION VIEWPORT	C.6.2.2.3.2	C.8.2.2.3.2
SET WORKSTATION WINDOW	C.6.2.2.3.1	C.8.2.2.3.1
TEXT	C.6.2.2.1.4	C.8.2.2.1.4
UPDATE WORKSTATION	C.6.2.2.5.1	C.8.2.2.5.1

Appendix IV

(to Part 2 of Annex C to Recommendation T.101)

Selection of the geometric display

The geometric display is selected by means of the US sequence (VPCE, Videotex Presentation Control Element):

<US> <3/2> <y>

as described in part 0 of T/CD 6.1. US is the UNIT SEPARATOR control and is coded 1/15. The <y> indicates the highest level of primitives, which is embedded in the geometric data following the US sequence. The currently defined values of <y> are given in Table IV.1.

After this US sequence (VPCE) all data is regarded as geometric data.

Table IV.1/T.101 – Relation between <y> code and levels

<y> code	Highest level
2/1	L0a
2/2	L1a
2/3	L2a

Part 3 – Photographic display

CONTENTS

	<i>Page</i>
C.1 Introduction	409
C.1.1 Protocol principles	409
C.1.2 Coding/Protocol definitions	410
C.1.3 Coding principles	410
C.2 Photographic Pixel Header Unit	411
C.2.1 <SDC> : Select display composition	412
C.2.2 <SCM>: Select Coding Method	417
C.3 Photographic Pixel Transfer Unit	417
C.3.1 <ORG> : Origin	417
C.3.2 <ARE> : Area	417
C.3.3 <DAT> : Pixel Data	418
C.4 Photographic Table Header Unit	418
C.4.1 <SET> : Table Set	419
C.4.2 <SIZ>: Table Size	419
C.5 Photographic Table Transfer Unit	419
C.5.1 <ID>: Identity	419
C.5.2 <LOC>: Location	419
C.5.3 <DAT>: Data	420
Appendix I – DPCM Image Coding – One Dimensional	420
Appendix II – DCT Image Coding – Two Dimensional	422

C.1 Introduction

The photographic facility allows for the transmission and display of an image consisting of individually defined picture elements (pixels) with many grey/colour levels. The image may be subjectively similar to a still broadcast quality television picture. Colour television techniques may be used to define the image and digital signal processing techniques may be used to compress the image for storage and transmission.

The protocol allows for many different photographic videotex systems to be specified but recommendations are given based on the CCIR digital television studio standard (Recommendation 601).

A transmission mode allowing all the presentation level bits to be used for photographic data (transparent mode) provides an efficient means of transmitting the relatively large amounts of data needed for photographic images.

C.1.1 Protocol principles

The transmission of a photographic image is accomplished using two Videotex Presentation Data Elements (VPDEs). For each VPDE type there are two subtypes: a header and a transfer unit. Their functions are outlined below.

- Pixel header unit – Gives the parameters defining the composition of the displayed image and the method of coding used for storage and transmission.

- Pixel transfer unit – Contains the actual data describing the image (grey/colour levels of each pixel).
- Table header unit – Specifies the type and format of tables used in the decoding process (e.g. set up a quantizer).
- Table transfer unit – Contains the actual values to be loaded into the tables (e.g. fill the quantizer).

C.1.2 Coding/Protocol definitions

In this part of the Recommendation the following definitions apply:

photographic parameter: A photographic parameter is a quantity that conveniently characterizes a particular aspect of the transmission or display of the photographic image (e.g. display resolution).

parameter field: A parameter field is the complete coding specification of a parameter. It consists of a parameter identifier and one or more parameter values.

parameter identifier: A parameter identifier introduces a parameter field and defines the particular parameter being specified.

component: Certain aspects of the display or transmission may have to split into separate parts, these are referred to as components. (e.g. colour components Y, U and V).

delimiter: A delimiter may be used to separate parameter values or data for different components.

C.1.3 Coding principles

The coding scheme provides for unambiguous identification of videotex control codes (columns 0, 1), photovideotex parameters (columns 2, 3) and allows 6 bits (columns 4, 5, 6, 7) to be used for data. A diagram of the code table is shown in Figure C.1.

- Column 2 – Identifies the parameter being specified (see Figure C.2);
- Column 3 – Is used to specify:
 - a value in decimal form.
 - to give a 'type number';
 - to separate parameters/data for different components.

Codes 3/0 to 3/9 represent decimal values 0 to 9.

3/11 is used to delimit decimal values. Leading zeroes may be omitted.

3/12 may be used if desired within a parameter to separate different components.

If a default value exists for a parameter it is assigned the 3/0 type value.

When a parameter has several components, the values for each component are specified in sequence. If the value of a trailing component(s) is(are) the same as the previous values then it(they) may be omitted. If a whole parameter is omitted, then it is assumed:

- that its value is implicit from other information;
- it is a default value; or
- the parameter is not applicable in this particular case.

Data may be coded using columns 4, 5, 6 and 7 of the code table. The protocol also allows for all code bits received at the presentation level to be used for data (transparency).

					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 ₁	b ₂	b ₃	b ₄			0	1	2	3	4	5	6	7
0	0	0	0	00	C0 Codes	Select Display Composition	Decimal Values	Pixel Data					
0	0	0	1	01									
0	0	1	0	02									
0	0	1	1	03									
0	1	0	0	04									
0	1	0	1	05									
0	1	1	0	06									
0	1	1	1	07									
1	0	0	0	08		Spare	Value Delimiter						
1	0	0	1	09									
1	0	1	0	10									
1	0	1	1	11									
1	1	0	0	12		Position and Transfer							
1	1	0	1	13									
1	1	1	0	14									
1	1	1	1	15									

T0822340-95/d178

Figure C.1/T.101 – Photovideotex Code Table

C.2 Photographic Pixel Header Unit

The header unit will take the form

US 3/4 2/0 <SDC> <SCM>

The definitions of parameter fields above are given below. The header applies for all following pixel transfer units until another header is sent or until the end of the session.

C.2.1 <SDC> : Select display composition

This field specifies the composition of the photographic display. It can contain up to 4 parameters.

<SDC> : <COM> <RES> <BPC> <STR>

		PARAMETER CODE		DEFAULT
SELECT DISPLAY COMPOSITION	-----	2/0	Components	(Y, U*, V*)
		2/1	Resolution	(4:2:2)
		2/2	Bits/Pixel/Component	(8/8/8)
		2/3	Sampling Structure	(Orthogonal Coincident)
SELECT CODING METHOD	-----	2/4	Image Coding	(Linear PCM)
PIXEL TRANSFER		2/12	Origin	(Top Left of Defined Display Area)
		2/13	Area	(Defined Display Area)
VALUES	-----	3/0	0	
		3/1	1	
		3/2	2	
		3/3	3	
		3/4	4	
		3/5	5	
		3/6	6	
		3/7	7	
		3/8	8	
		3/9	9	
DELIMITERS	-----	3/11	Decimal Delimiter	
		3/12	Component Delimiter	

* CCIR colour difference signals.

Figure C.2/T.101 – Hixel Parameter Codes and Defaults

C.2.1.1 <COM> : Display Components

A displayed image may be formed from one or more components. For a monochrome image only one component is needed but colour requires three. Sets of different component possibilities are given in the table below and a code is assigned to each.

<COM> = display component identifier, component type number
 = 2/0 3/C
 3/C = 3/0 colour YU*V*
 3/1 monochrome
 :
 :
 3/15 (for later allocation)
 * CCIR colour difference signals.

C.2.1.1.1 CCIR Monochrome and colour component

A colour image is defined as being comprised of a luminance (brightness) and a pair of colour difference (colouring) components. A monochrome image contains the luminance (brightness) component only.

The luminance signal is obtained from gamma-corrected primary signals, R, G, B and corresponds to the equation:

$$Y = 0.299R + 0.587G + 0.114B$$

The colour-difference signal components are then defined as:

$$R-Y = 0.701R - 0.587G - 0.114B$$

$$B-Y = -0.299R - 0.587G + 0.886B$$

The colour-difference signals have a range of 0.701 to –0.701 for R-Y, and 0.886 to –0.886 for B-Y. To restore the signal excursion of the colour-difference signals to unity (i.e. +0.5 to –0.5), coefficients are applied to the R-Y and B-Y. The weighted colour-difference signals U* and V* are then defined as

$$U^* = 0.564 (B-Y)$$

$$V^* = 0.713 (R-Y)$$

C.2.1.2 <RES> : Display resolution

Display resolution is defined as the number of pixels horizontally and vertically in the defined area.

The CCIR Recommendation 601 specifies for the digital television studio standard a 13.5 MHz luminance sampling frequency and a 6.75 MHz chrominance sampling frequency for 625 and 525 line systems. The actual number of pixels is dependent on the size of the area. For this existing standard and for other systems based on it a shorter coding can be used to specify the horizontal and vertical resolution. The CCIR nomenclature for sampling frequencies is used where the frequencies of the three components are expressed in sequence and relative to 3.375MHz (e.g. 13.5 / 6.75 / 6.75 MHz = 4:2:2).

<RES> = resolution ident, resolution type number
 = 2/1 3/R
 3/R = 3/0 4:2:2 (CCIR studio standard)
 3/1 2:1:1
 : (for later allocation)
 3/15 Decimally defined value – see below.

C.2.1.2.1 Decimally Defined Resolution

Other resolutions may be specified if required by specifying in decimal form the number of pixels horizontally and vertically. If the pixel is formed from more than one component the resolution of each component is specified in sequence in descending order of resolution. The highest resolution component is referred to as the first component. The resolution of the other lower resolution components are specified as a fraction of the resolution of the first component and are coded as the reciprocal of the fraction (e.g. 1/4 is specified as 4).

<RES> = resolution ident, number of horiz pixels, number of vert pixels
= 2/1 3/15 ...3/uHa 3/11 ...3/uVa 3/11 (1st component)
... :
...3/uHn 3/11 ...3/uVn 3/11 (nth component)

C.2.1.3 <BPC> : Bits per Display Component

This parameter gives the number of grey or colour levels a pixel may have. The number of levels available for each component is expressed in terms of the number of bits of storage per pixel per component if stored in an uncompressed PCM form. Normally this will be a value in the range 1 to 9 and can be specified by a single code value.

<BPC> = bits/pixel/comp ident, no of bits/pixel/component
= 2/2 ...3/Ba 3/11 (1st component)
... :
...3/Bn 3/11 (nth component)

where:

3/B = 3/0 8 bits/pixel (default)
3/1 1 bit/pixel
3/2 2 bits/pixel
3/9 9 bits/pixel
3/15 Decimally defined value – see below.

C.2.1.3.1 Decimally Defined Bits per Display Component

The number of bits per component may if necessary be specified in full decimal form.

<BPC> = bits/pixel/comp ident, no of bits/pixel/component (in decimal form)
= 2/2 3/15 ...3/uBa 3/11 (1st component)
... :
...3/uBn 3/11 (nth component)

C.2.1.3.2 CCIR Level Assignment

The CCIR Recommendation defines certain reference binary levels for a uniformly quantized PCM image having 8 bits per sample. Luminance samples are represented by a positive binary number and colour difference samples by a offset binary number. The total nominal excursion of the luminance signal corresponds to 220 quantization levels, with black corresponding to level 16, and nominal white to level 235 (see Figure C.3). There is an unequal quantization margin above and below the nominal signal, because there is a greater variation in the nominal white level than in the nominal black level and the effect of clipping the overshoot will be more preceptible in the white region.

LEVEL		BINARY	HEX
255	-----	(11111111)	FF
235	-----White-----	(11101011)	EB
16	-----Black-----	(00010000)	10
0	-----	(00000000)	0
LUMINANCE CODING RANGE			
LEVEL		BINARY	HEX
255	-----	(11111111)	FF
240	-----Maximum-----	(11110000)	F0
128	-----Zero-----	(10000000)	80
16	-----Minimum-----	(00010000)	10
0	-----	(00000000)	0
COLOUR DIFFERENCE CODING RANGE			

Figure C.3/T.101 – Level Assignment

Given that the luminance signal is to occupy 220 levels and that black is to be at level 16, the digital luminance signal Y_d may be calculated by:

$$Y_d = 219Y + 16$$

Y is the luminance analogue signal of any colour, expressed as a fraction of unity.

Y_d is the corresponding level number after quantization to the nearest integer value.

The colour-difference signals each occupy 225 levels in the central part of the quantization scale, with zero signal corresponding to level 128 (see Figure C.3).

Given that the colour-difference signals are to occupy 225 levels and that zero level is to be 128, the decimal values of the colour-difference signals, V^* , U^* may be calculated:

$$V^*_d = 224 [0.713 (R-Y)] + 128$$

$$U^*_d = 224 [0.564 (B-Y)] + 128$$

V^*_d U^*_d are the corresponding level numbers after quantization to the nearest integer value.

$R-Y$, $B-Y$ are the colour-difference analogue values of any colour expressed as a fraction of unity.

C.2.1.4 <STR> : Sampling Structure

The structure parameter defines the spatial and temporal relationship between pixels on adjacent lines and fields, see Figure C.4 (Sampling Structures). The relationship between samples of the first component is specified first followed by the relative structure of the other components to the first.

$$= \frac{2}{3} \quad \frac{3}{S} \quad \frac{3}{R}$$

3/S	=	3/0 line orthogonal field orthogonal 3/1 line orthogonal field quincunx 3/2 line quincunx field orthogonal 3/3 line orthogonal single field 3/4 line quincunx single field : : {for later 3/15 {allocation	} interlaced
3/R	=	3/0 coincident 3/1 alternate samples 3/2 sequential line : : {for later 3/15 {allocation	

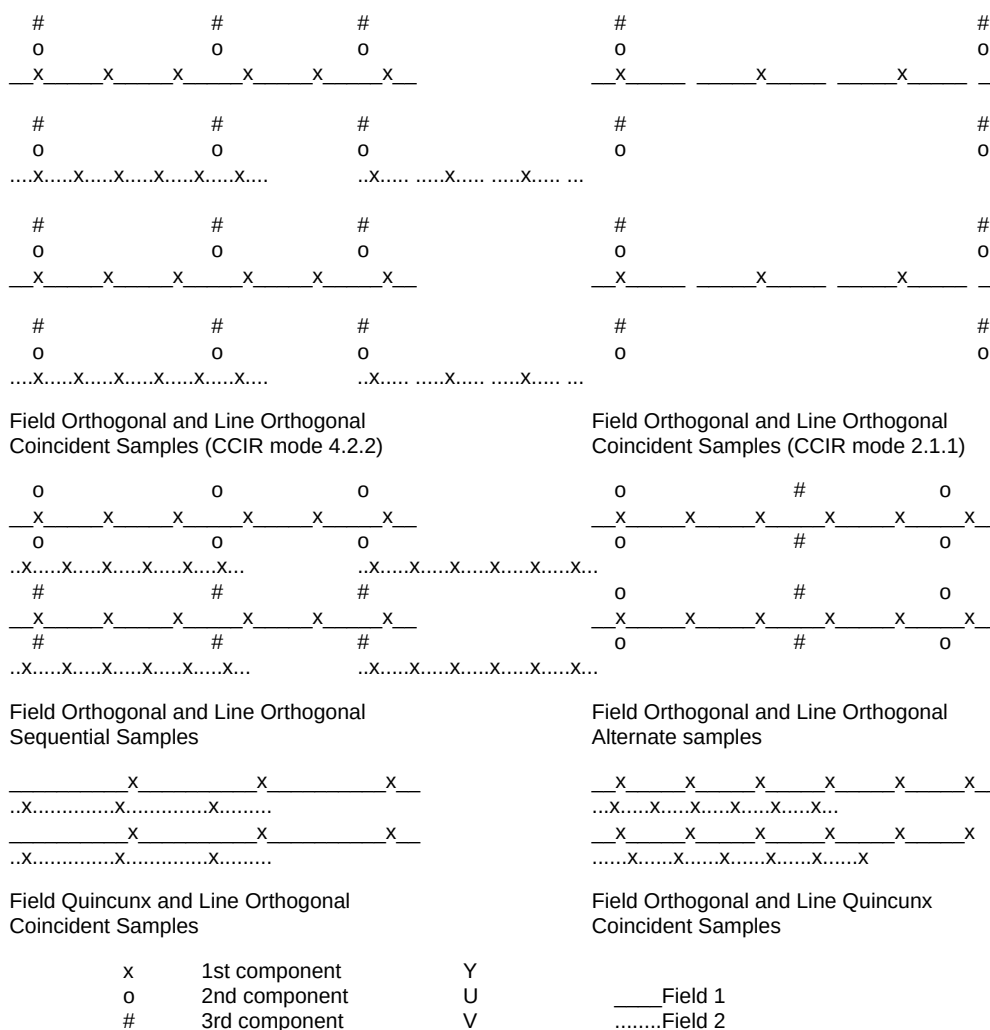


Figure C.4/T.101 – Sampling Structures

C.2.2 <SCM>: Select Coding Method

The way in which data is coded for storage and/or transmission, is described in this subclause. At present, only one parameter field is specifiable, other parameters are given in the descriptions of particular coding techniques. Later, when other generally useful parameters have been identified these may be made individually specifiable.

<SCM> : <ICT>

C.2.2.1 <ICT>: Image Coding Techniques

A photographic image is normally encoded using digital signal processing techniques (e.g. pulse code modulation or a mathematical transform). The various methods offer advantages such as high compression, a desirable image build-up, good quality, or be suitable for a certain type of image. A table of image coding techniques is given and a code is assigned to each. A subtype may be used to distinguish between different techniques of the same type. Each technique may have an independent set of subtypes specified in a list.

<ICT> = coding ident, coding type number
= 2/4 <TY> <STY> <SSTY>
<TY> : Type <STY> : Subtype <SSTY> : Subsubtype
3/1 dpcm ----- 3/0 one dimensional
| (Appendix I)
|
3/2 transform ----- 3/1 Cosine ----- 3/0 two dimensional
| (Appendix II)
|
:
: {for future
3/15 {allocation

The details of the recommended coding methods are given in Appendices I and II. A particular coding technique may implicitly specify certain parameters such as the number of bits per sample or the sequence for transmission.

C.3 Photographic Pixel Transfer Unit

This VPDE carries the data defining the grey/colour level of the pixels forming the photographic image and specifies where the image is to be located on the display. It takes the form:

US 3/4 <ORG> <ARE> <DAT>

C.3.1 <ORG> : Origin

The origin is the first pixel position to which the data following will refer. This is the top left corner of the rectangular area as defined below. It is specified in terms of the horizontal and vertical pixel position of the first component with respect to the Defined Display Area. (see C.2.1.2). See Figure C.5 for Full Screen, Origin and Defined Display Area relationships.

<ORG> = origin ident, horiz pix pos, vert pix pos
= 2/12 ...3/uX 3/11 ...3/uY 3/11

C.3.2 <ARE> : Area

This defines a rectangular area to be filled by the photographic data following. The width and height of the area are specified in terms of the number of pixels of the first component. A non-rectangular image can be constructed from small rectangles, if desired of only one line's height. Alternatively, a mask of the required shape may be created on an outer layer.

<ARE> = area ident, area width, area height
 = 2/13 ...3/uAW 3/11 ...3/uAD 3/11

When an area has been completely filled by all components the origin is assumed to be set back to the origin of the area.

Other methods of defining the area to be filled by the image are for further study.

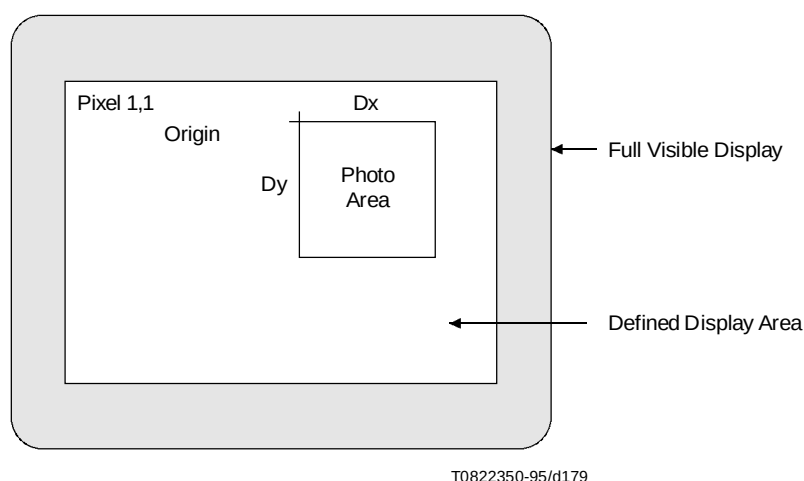


Figure C.5/T.101 – Display Areas

C.3.3 <DAT> : Pixel Data

C.3.3.1 Within the Code Table

Following the introduction variables described above any codes from columns 4, 5, 6 or 7 will be automatically interpreted as data, the least significant 6 bits of each code being considered as a continuous bit stream containing concatenated data values.

<DAT> = 4/H 5/H 6/H 7/H ...

C.3.3.2 Transparent Data

As photographic images contain a relatively large amount of data it is desirable for increased efficiency to use all the presentation level code bits for actual data (8 bits per character). In such a mode all codes pass uninterrupted by the normal presentation level control codes and the mode is thus termed transparent.

The transparent mode is entered using the TRANSPARENT data VPDE (see Part 7).

C.4 Photographic Table Header Unit

For certain photovideotex schemes various tables are needed in the decoding process whose contents have to be changed for different images. The table header unit allows for a set of tables to be specified. Figure C.6 shows examples of table structures.

The header unit will take the form:

US 3/5 2/0 <SET> <SIZ>

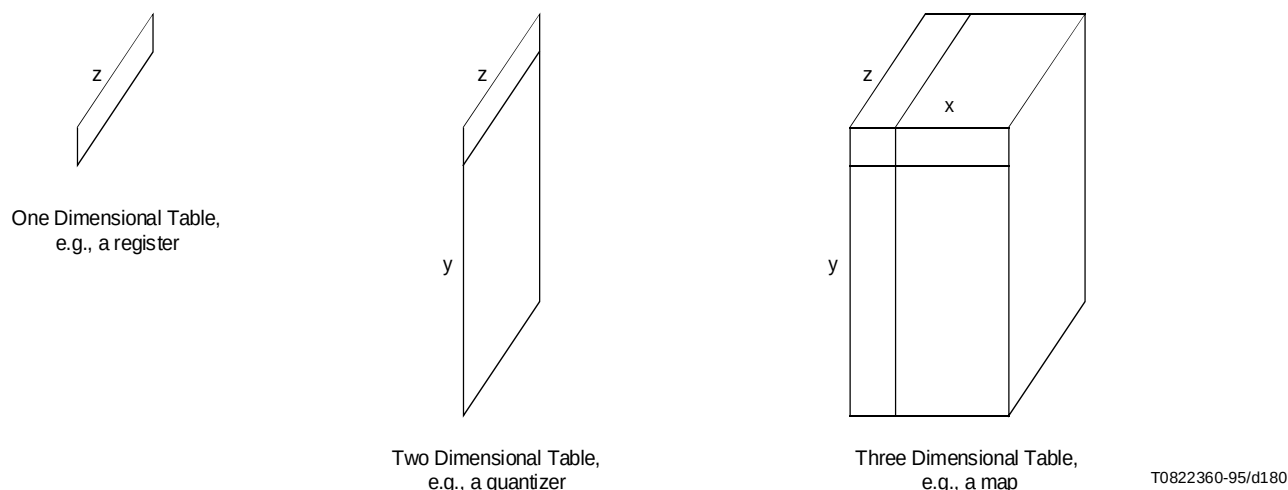


Figure C.6/T.101 – Parameter Table Structure

C.4.1 <SET> : Table Set

The use to which this table is put is defined within a particular coding technique description. The parameter field specifies the table type identity number and the number of tables of that type required. The parameter field is defined within the particular coding technique.

<SET> = set ident, table type number, number of tables
 = 2/1 ...3/uT 3/11 ...3/uN 3/11

C.4.2 <SIZ>: Table Size

A generalized table is defined in tridimensional form. If a table has only two dimensions (e.g. a quantizer) the third dimension is omitted.

<SIZ> = table size ident, table depth (Z), height (Y), width (X)
 = 2/2 ...3/uZ 3/11 ...3/uY 3/11 ...3/uX 3/11

C.5 Photographic Table Transfer Unit

A table transfer unit is used to fill a previously defined table(s).

US 3/5 <ID> <LOC> <DAT>

C.5.1 <ID>: Identity

A particular set of tables is identified using its table set type number and table number within the set. Where there are a number of tables of a given type they will be filled in sequence. If only one value is given it is assumed to be the first table of the set that is addressed.

<ID> = table identification ident, set type number, table number
 = 2/1 ...3/uT 3/11 ...3/uN 3/11

C.5.2 <LOC>: Location

A particular location or bit within a table may be addressed by using its XYZ coordinates as used for the table dimensions. See Figure C.7.

<LOC> = location ident, XYZ address
 = 2/2 ...3/uX 3/11 ...3/uY 3/11 ...3/uZ 3/11

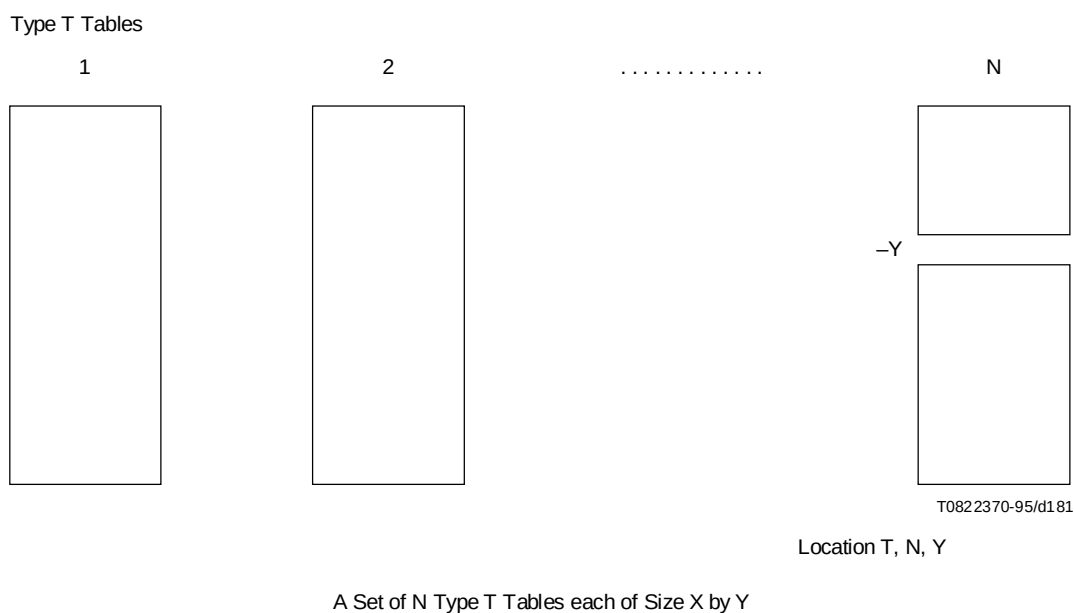


Figure C.7/ T.101 – Parameter Table Addressing

C.5.3 <DAT>: Data

C.5.3.1 Within The Code Table

Table data is coded using the 6 least significant bits of codes from columns 4, 5, 6 and 7.

<DAT> = 4/H 5/H 6/H 7/H ...

C.5.3.2 Transparent Data

A table may be filled using a transparent mode as described in C.3.3.2.

Appendix I

(to Part 3 of Annex C to Recommendation T.101)

DPCM Image Coding – One Dimensional

I.1 Introduction

Differential pulse code modulation (DPCM) is a digital signal coding technique in which the differences between adjacent digitally encoded sample values are used for storage and/or transmission. DPCM is also referred to as predictive coding since the difference between the present value and a predicted value may be used.

A coding scheme may be derived using a difference value based on a one dimensional or two dimensional prediction. The quantizer relating difference codes to actual value changes may be fixed or adaptive (i.e. changed according to image characteristics).

I.2 One Dimensional – Previous Element Prediction DPCM

I.2.1 General

This scheme provides for 50% data compression. It is relatively simple to decode, allowing for cheap and/or high speed decoding. The image is built up pixel by pixel, line by line in two scans. The first scan fills the whole area with a monochrome picture which is then coloured in the second scan.

I.2.2 Compression Technique

The pixel colour information is described using the television signal components luminance (Y) and chrominance (U and V). The prediction for the next sample value (P_{n+1}) is that it will be the same as the present value on the same TV line. The prediction is reset at the start of each line to a mid-range value. For luminance and chrominance this is the value 128 (decimal). The difference or error (D) between the prediction (P) and the actual value (V) is coded and used for storage and/or transmission.

$$\text{Transmitted value } D = V - P = V - V$$

Data compression is achieved by using a non-uniform quantizer. A 16-level quantizer is used and so 4 bits/sample are transmitted. The scheme is non-adaptive and thus the quantizer is fixed as shown below. The same quantizer is used for luminance and both chrominance signals.

The value for display is reconstructed by adding the difference/error to the prediction:

$$\text{Display value } V_{n+1} = P_{n+1} + D_{n+1} = V_n + D_{n+1}$$

<i>Input difference</i>	<i>Transmitted code</i>	<i>Output difference (D)</i>
0-2	0	1
3-6	1	4
7-12	2	9
13-21	3	16
22-35	4	27
36-61	5	44
62-99	6	79
100-255	7	120
-(1-2)	8	-1
-(3-6)	9	-4
-(7-12)	10	-9
-(13-21)	11	-16
-(22-35)	12	-27
-(36-61)	13	-44
-(62-99)	14	-79
-(100-255)	15	-120

I.2.3 Transmission Sequence

The luminance values for the area are transmitted first. Each byte contains two samples. Starting at the origin, values are sent in sequence pixel by pixel, line by line. When the area has been filled by one component the next pixel position is reset to the origin of the area. The chrominance values are sent in UV pairs in sequence pixel by pixel, line by line. Again two samples are sent per byte.

Y11	Y12		Y1N
Y21	Y22		Y2N
:			:
YM1	YM2		YMN
UV11	UV13		UV1N/2
UV21	UV23		UV1N/2
:			:
UVM1	UVM3		UVMN/2

I.3 Example of Coding for DPCM

Photographic Data Header Unit

This will normally only be sent once at the start of a photovideotex session.

VPCE : US 3/4 2/0 (photographic data header unit)
VSCE : 2/1 3/1 (display composition = 2:1:1)
2/3 3/3 (structure = line orthogonal single field)
2/4 3/1 3/0 (coding technique = DPCM – one dimensional)
Defaults not transmitted – components = Y, U*, V*
bits /component = 8/8/8

Photographic Data Transfer Unit

This unit is sent for each photovideotex image.

VPCE : US 3/4 (photographic data transfer unit)
PDSU: 2/12 3/hX 3/tX 3/uX 3/11 3/hY 3/tY 3/uY 3/11
not 2/0 (origin pixel location = htuX htuY)
: 2/13 3/hW 3/tW 3/uW 3/11 3/hH 3/tH 3/uH 3/11
(area width and height = htuW × htuH)
US 3/15 N Y Y Y Y ... (transparent data for N bytes)
luminance data
:
:
:
US 3/15 N Y Y Y Y ...
US 3/15 N U* V* U* V* ...
chrominance data
:
:
:
US 3/15 N U* V* U* V* ...

Appendix II

(to Part 3 of Annex C to Recommendation T.101)

DCT Image Coding – Two Dimensional

II.1 Introduction

The main steps for image coding and decoding on a transform basis are shown in Figure II.1.

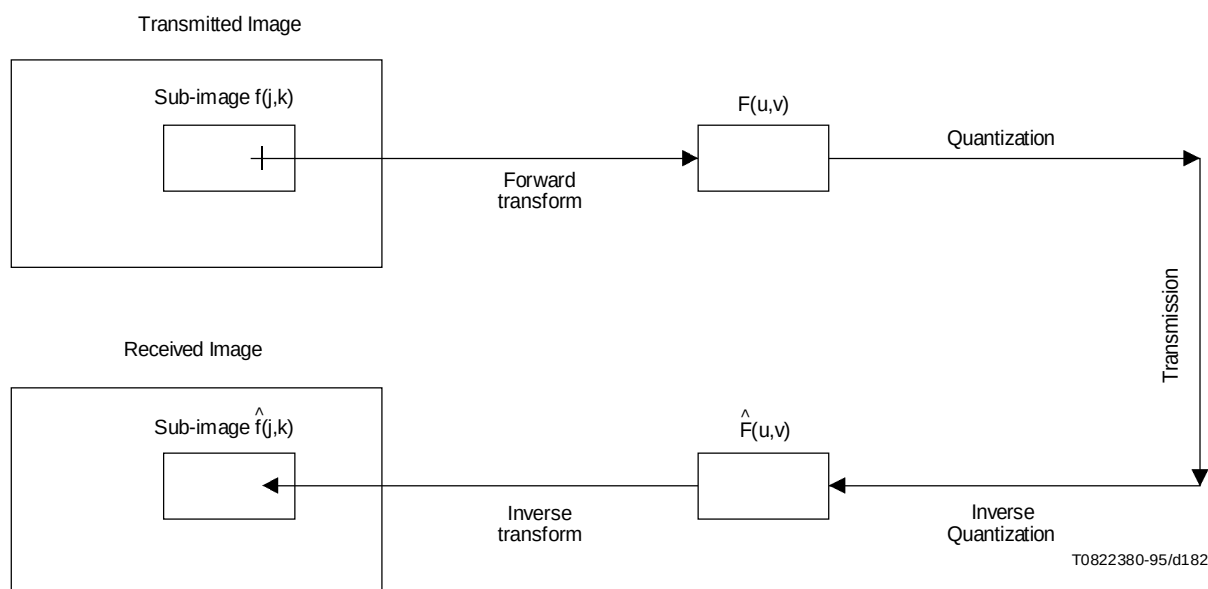


Figure II.1/T.101 – Transform Coding and Decoding

The principle characteristics of image coding by transform methods are:

- by using an orthogonal transform such as the Discrete Cosine Transform, high energy compaction is achieved;
- adaptivity due to sorting the transform sub-images of an image into classes by the level of image activity present;
- averaging of channel noise over the whole sub-image.

The Discrete Cosine Transform (DCT) is a coding method belonging to the general class of discrete orthogonal transforms.

The typical performance obtained with DCT is:

- 1) 0.5-1 bit per pixel for monochrome images;
- 2) 1-2 bits per pixel for colour images.

II.2 The Discrete Cosine Transform

II.2.1 General

The two-dimensional Cosine Transform of a discrete function

$f(j,k)$ $j,k = 0, 1, \dots, N-1$ is defined as:

$$F(u,v) = \frac{4C(u) \cdot C(v)}{N^2} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j,k) \cdot \cos\left[\frac{(2j+1)\pi \cdot u}{N}\right] \cdot \cos\left[\frac{(2k+1)\pi \cdot v}{N}\right]$$

where

$$u, v = 0, 1, \dots, N-1$$

The inverse transform is:

$$f(j,k) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u) \cdot C(v) F(u,v) \cdot \cos\left[\frac{(2j+1)\pi \cdot j}{N}\right] \cdot \cos\left[\frac{(2k+1)\pi \cdot k}{N}\right]$$

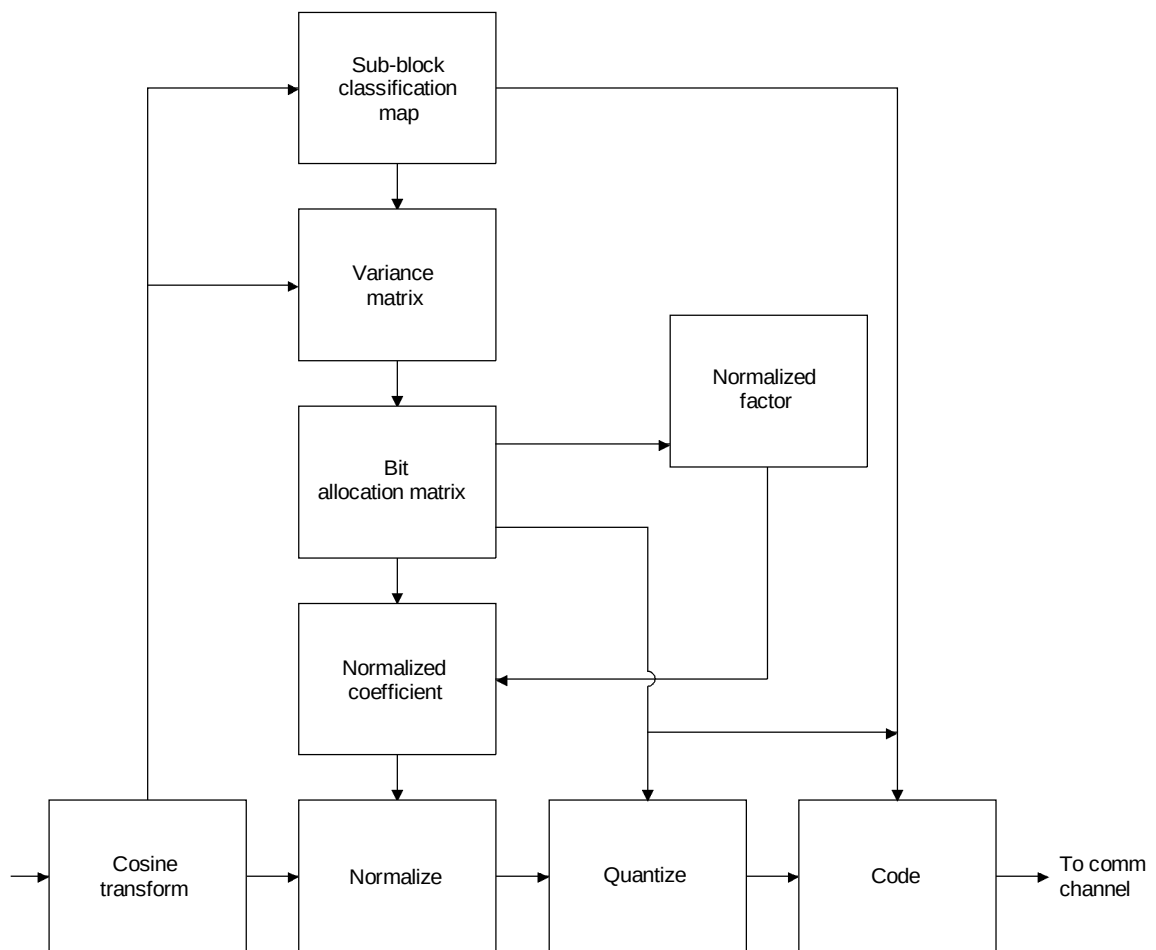
where

$$C(0) = \frac{1}{\sqrt{2}}$$

$$C(u) = C(v) = 1 \text{ for } u, v = 1, 2, \dots, N-1$$

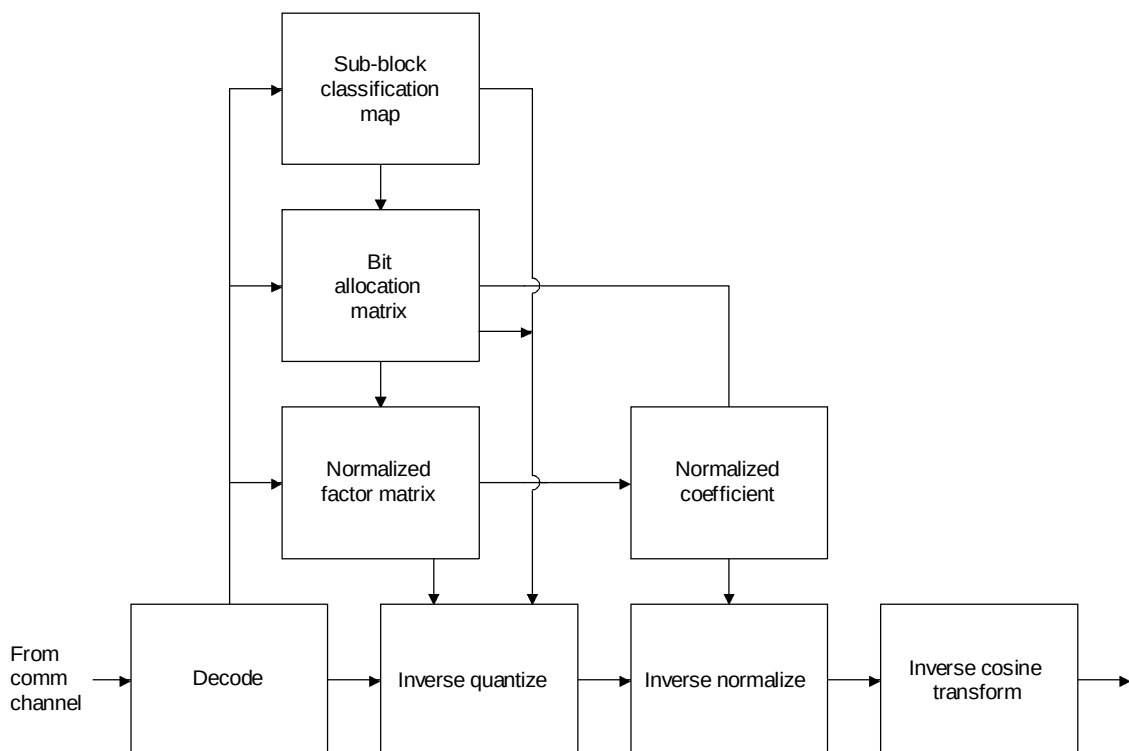
$$C(u) = C(v) = 0 \text{ elsewhere}$$

Block diagrams of a DCT adaptive coding system are shown in Figures II.2 and II.3.



T0822390-95/d183

Figure II.2/T.101 – Cosine Transform Adaptive Coding System



T0822400-95/d184

Figure II.3/T.101 – Cosine Transform Adaptive Coding System – Decode

II.2.2 Transform Sub-block Classification

The transform sub-block classification attempts to sort the transform blocks of an image according to criteria which may be functions of image activity, directionality, fineness etc present within each transform sub-image.

After the calculation is performed, the whole range of values of the chosen criterium is divided amongst the number of classes which serve the principle of adaptivity. Within limits, the greater the number of classes the better the adaptivity.

Finally, a classification map is generated within which each sub-image is identified by its class identifier. This identifier acts as an index to the bit allocation table associated with that class.

II.2.3 Bit Allocation Table

This step allocates a number of coding bits to individual elements according to their class reference and to a fixed data rate for an average distortion at or below an acceptable level (rate-distortion theory). Bits are then distributed between “busy” and “quiet” image areas to provide the desired adaptivity – more bits being assigned to areas of high image activity and less to those of low activity.

The bit allocation strategy is at the designer’s convenience. The following scheme provides for a maximum of 16 different classes.

II.2.4 Normalization of Coefficients

This calculation is performed to

- avoid clipping of the transform samples prior to quantization;
- use normalized quantizers associated with normalized probability laws.

In order to use normalized probability densities, e.g.

$$p(x) = \frac{l}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$$

for the definition of quantizers it is appropriate to specify normalized values for the transform coefficients on which the quantification process is applied.

The normalization of the transform coefficients is performed through the following scheme:

for $u, v = 0$ the normalized value to be quantized is:

$$x = \frac{Fm, \ell(0,0) - 2m}{2(2^{N_d} - 1) / X_{128}}$$

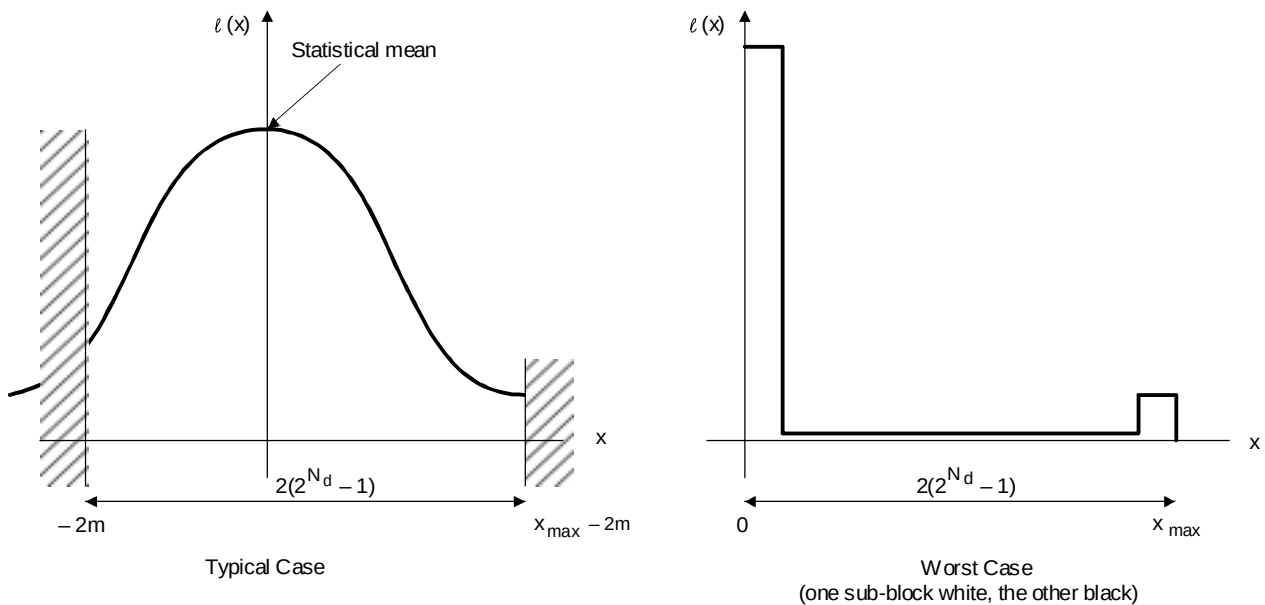
where

$$m = E f(j,k),$$

N_d Number of bits per pixel in the original image $f(j,k)$,

X_{128} The highest 8 bit decision level.

This guarantees that no dc values are clipped.



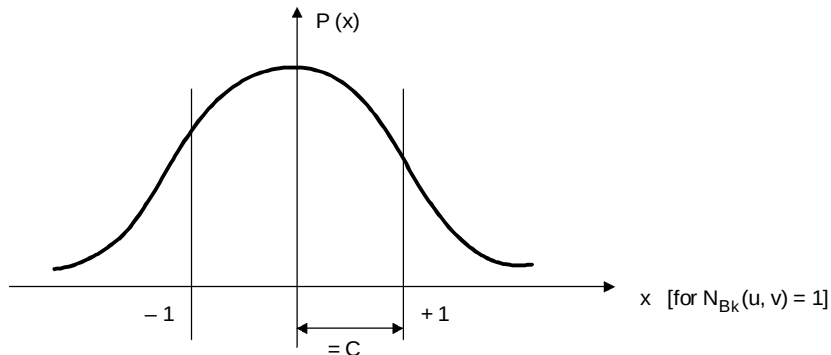
for $u, v \neq 0$

$$x = \frac{Fm, \ell(u, v)}{C \cdot 2^{N_{Bk}(u,v)-1}}$$

Where

$$C = \sup_{u, v, k} \sigma_x(u, v) \text{ for } N_{Bk}(u, v) = 1$$

$$E Fm, \ell(u, v) = 0$$



T0822410-95/d185

At the receiving end after inverse quantization the value of the normalisation factor for $F_{m, \ell}(0, 0)$ and $F_{m, \ell}(u, v)$ is required in order to obtain the correct value of the coefficient before processing the inverse cosine transform.

- For $F_{m, \ell}(0, 0)$ the value of N_d is required (default value is 8 bits)
- For $F_{m, \ell}(u, v)$ the value of C is required (no default value).

II.2.5 Quantization

The normalized samples are optimally quantized with the number of quantization levels (bits) set according to the bit allocation tables. The quantization process attempts to define a relationship between a transform coefficient $F_{m, \ell}(u, v)$ and a binary number from $N_{Bk}(u, v)$ bits from the appropriate bit allocation table. This relationship is derived from the distribution law of the coefficients and some function of the error between the input and the output of the quantizer.

The Gaussian distribution law used:

$$p(x) = \frac{\ell}{\sqrt{2\pi} \sigma(u, v)} e^{-\frac{x^2}{2}}$$

with x in the general form

$$x = \frac{F_{m, \ell}(u, v) - 2m}{\sigma_k(u, v)}$$

The criterion used as a function of the error between the input and the output of the quantizer is the mean squared error:

$$D = \sum_{i=1}^{2^{N_{Bk}}} \int_{S_{in(i)}}^{S_{in(i+1)}} (S_{in} - S_{out})^2 p(x) \cdot dx$$

which gives, if

x_k are the end points of the $2^{N_{Bk}(u, v)}$ input ranges

y_k are the output levels of the corresponding input ranges

$$x_i = (y_i + y_{i-1} - 1)/2 \text{ for } i = 2, \dots, 2^{N_{Bk}(u, v)}$$

and

$$\int_{x_i}^{x_{i+1}} (x - y_i) \cdot o(x) dx = 0 \text{ for } i = 1, \dots, 2^{N_{Bk}(u, v)}$$

and

Table II.1/T.101 – Example for $N_{Bk} = 1, 2, 3$ bits

i	N = 2		N = 4		N = 8	
	x_i	y_i	x_i	y_i	x_i	y_i
1	0.0	0.7980	0.0	0.4528	0.0	0.2451

II.3 Application of the Discrete Cosine Transform

DCT image coding may be performed in various ways. This application uses 1 to 2 bits per pixel to code colour images.

This application is coded for with:

$$\langle \text{ICT} \rangle = 2/4 \ 3/2 \ 3/1 \ 3/0$$

(coding ident, transform, cosine, two dimensional).

II.3.1 Image Structure

The whole image (two fields) is used (see Figure II.4).

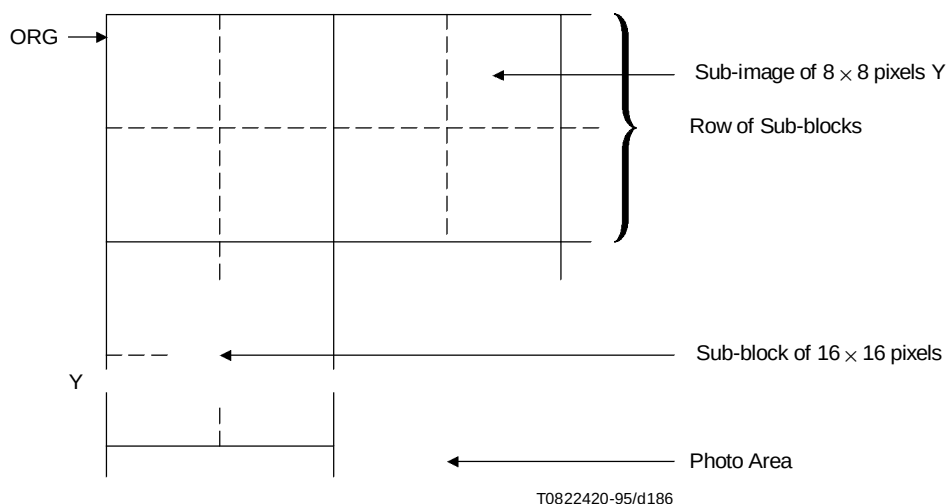


Figure II.4/T.101 – Image Structure

II.3.1.1 Spatial Structure

The sampled image has the following structure (see Figure II.).

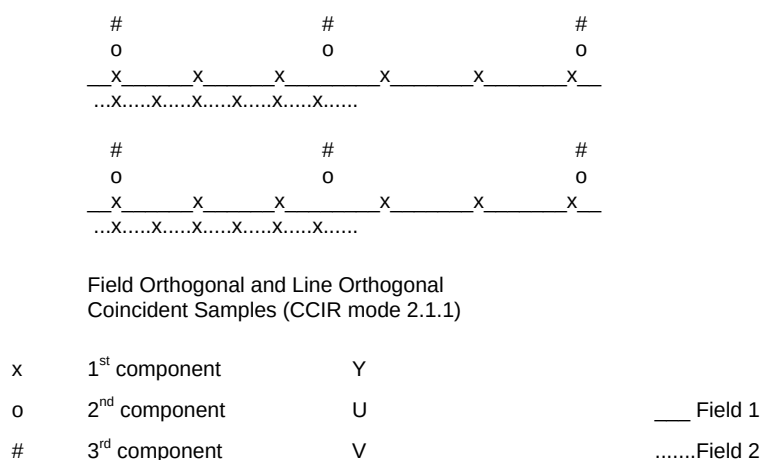


Figure II.5/T.101 – Sampling structure

II.3.1.2 Temporal Structure

The transmitted transform image has the following structure (see Figure II.6).

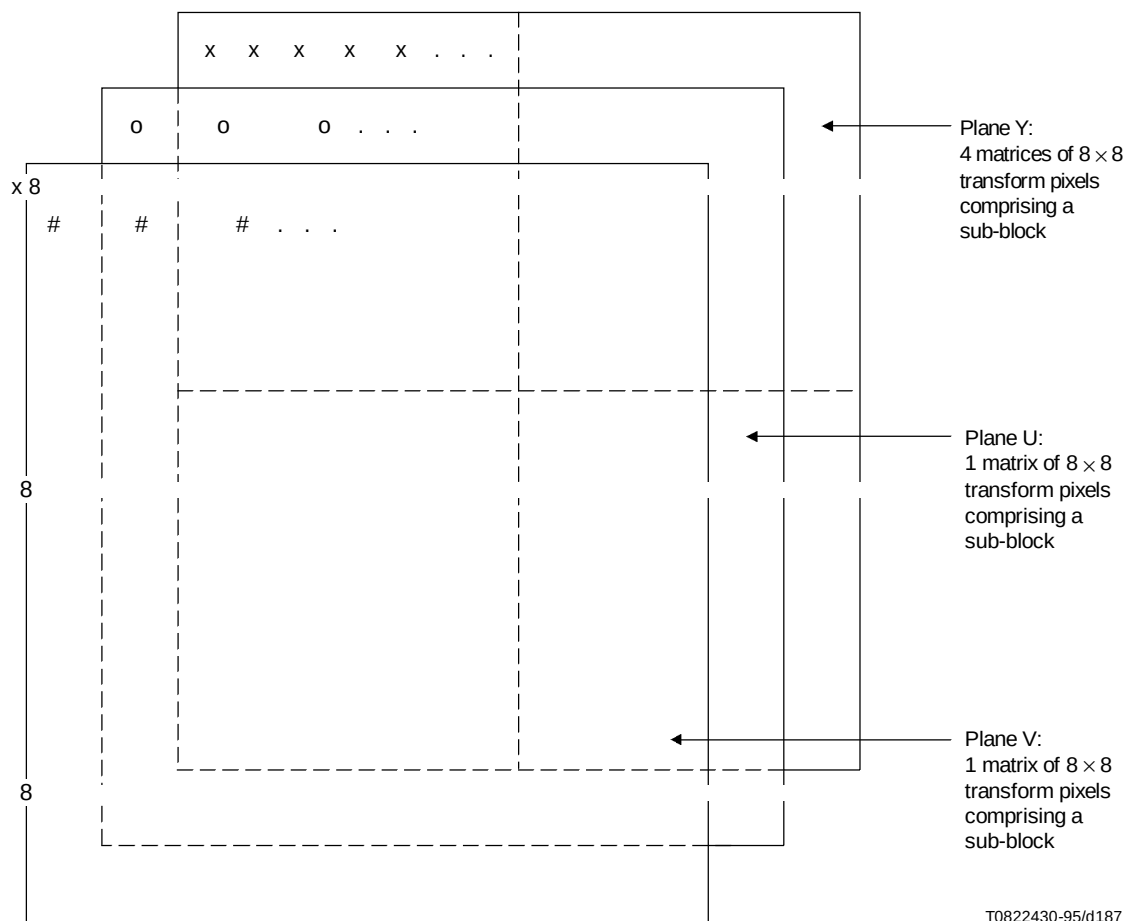


Figure II.6/T.101 – Transmitted Transform Image Structure

In mode 2.1.1 one sub-block U (8×8) and one sub-block V (8×8) are transmitted for each sub-block Y (16×16).

II.3.2 Coding Parameters

These parameters will be implicitly specified via <SCM> in the sub-sub-type byte with the coded value 3/0.

The parameters are:

- sub-block of 16×16 pixels (based on first component);
- sub-image of 8×8 pixels (based on first component);
- 1 to 16 activity classes;
- quantization law using Gaussian distribution of densities;
- source image quantized with 8 bits per component.

II.3.3 Table Types and Structures

The image is defined using three tables.

II.3.3.1 Table 1

This defines the following:

- $mY = E_y \{ f(j,k) \}$: mean of the Y component over the whole image on one unsigned byte.
- CY: normalization coefficient¹¹⁾ of Ym for $(u, v) = 0$ coded as $M \times 2$ with the mantissa M on two bytes and the exponent n on one signed byte.
- $mU = E_u \{ f(j,k) \}$: mean of the U component over the whole image on one unsigned byte.
- CU: normalization coefficient¹¹⁾ of Y for $(u, v) = 0$ coded as $M \times 2$ with the mantissa M on two bytes and the exponent n on one signed byte.
- $mV = E_v \{ f(j,k) \}$: mean of the V component over the whole image on one unsigned byte.
- CV: normalization coefficient¹¹⁾ of V for $(u, v) = 0$ coded as $M \times 2$ with the mantissa M on two bytes and the exponent n on one signed byte.

II.3.3.1.1 Data Structure

The data will be sent in the following sequence (mY first):

mY CY mU CU mV CV

(bytes are sent most significant bit first.)

II.3.3.2 Table 2

This defines the following:

- N Luminance bits allocation tables
- N U component bits allocation tables
- N V component bits allocation tables

II.3.3.2.1 Data Structure

Four bits are provided for allocation of fifteen bits per coefficient.

For each table data is transmitted row by row, commencing with byte (most significant bit first). Tables are transmitted in order Y, U and then V (see Figure II.7).

If required the data related to each component may be transmitted in separate Transparent data VPDEs. (See Part 7.)

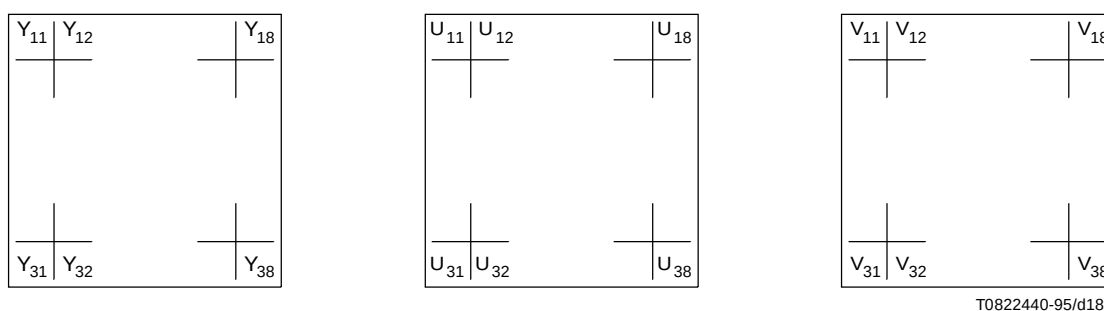


Figure II.7/T.101 – Bit Allocation Tables

¹¹⁾ The Normalization coefficient for Fm, $\ell (0,0)$ is not transmitted as the most likely value of N_d is 8 bits and therefore this is the default value.

II.3.3.3 Table 3

This is a table of variable length which describes the activity over the whole image. Each sub-image is given a 4-bit reference to one of the classes.

II.3.3.3.1 Data Structure

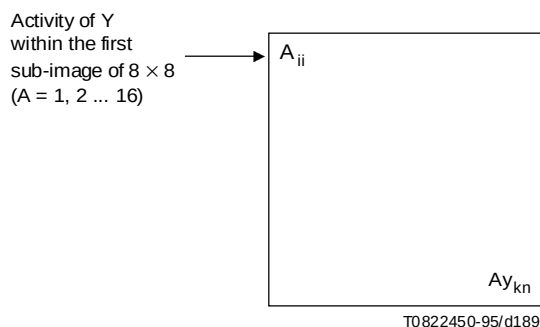


Figure II.8/T.101 – Activity Table for Y

Similar tables are transmitted for U and V with appropriate numbers of coefficients. (Four times fewer U and V coefficients than Y when in 2.1.1 CCIR mode.)

For each table data is transmitted row by row, commencing with byte (most significant bit first). Tables are transmitted in order Y, U and then V.

If required the data related to each component may be transmitted in separate Transparent data VPDEs. (See Part 7.)

II.3.4 Photographic Pixel Transfer Unit

The header is not described here as it depends on the particular image being transmitted. Photographic pixel data is transmitted in Transparent data mode (See Part 7).

In order to avoid propagation of errors through consecutive sub-blocks, each sub-block description is resynchronized by a new Transparent VPDE with an appropriate length indicator.

Each sub-block is coded with transformed and quantized coefficients sent in sequence as follows:

```

US 3/4  <ORG>  <ARE>

US 3/15 length Y1 Y2 Y3 Y4 U V

US 3/15 length Y'1 Y'2 Y'3 Y'4 U' V'

US 3/15 length Y"1 Y"2...

:

:

:
```

II.4 Example of Coding for DCT

The VPDEs will take the form:

PIXEL HEADER UNIT

VPCE	US	3/4	2/0			
SDC						Colour YUV (default)
Display composition	2/1	3/4 3/6	3/1 3/2	3/6 3/5	3/11 3/11	416Hy × 625Vy
		3/2 3/3	3/0 3/1	3/8 3/2	3/11 3/11	208Huv × 312Vuv
	2/3	3/0	3/1			Fields and lines alternate samples
SMC Coding method	2/4	3/2	3/0	3/0		Discrete cosine transform – two dimensional

TABLE TRANSFER UNIT

	Table 1 Header					
VPCE	US	3/5	2/0			
Table Set	2/1	3/1	3/11	3/3	3/11	Table 1; 3 sub-tables
Table Size	2/2	3/8	3/11	3/9	3/11	Z = 8 bits, Y = 9
432	Table Transfer Unit T.101 (11/94)					
VPCE	US	3/5				
Address	2/1	2/1	2/11			Table 1 address 0

VPCE	US	3/4					
<ORG>							Top Left DDA (default)
<ARE>	2/13	3/1	3/6	3/0	3/11		X = 160 pixels
		3/3	3/2	3/0	3/11		Y = 320 pixels
Data	US	3/15	length		transparent data ...		

Part 4 – Define Dynamically Redefinable Character Sets (DRCS)

CONTENTS

	<i>Page</i>
C.1 Introduction	434
C.2 DRCS Header Units.....	434
C.2.1 <ICS>: Identification of Character Set	435
C.2.2 <SDC>: Select Dot Composition.....	435
C.2.3 <SCM>: Select Coding Method.....	436
C.2.4 <SSA>: Select Set Attributes	437
C.3 Pattern Transfer Units.....	437
C.4 Coding Methods.....	437
C.4.1 'Direct' Coding Method.....	437
C.4.2 'Runlength' Coding Method.....	440
C.5 Designation and Invocation of DRCS	442
C.6 Applicability of Attributes.....	442

C.1 Introduction

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 ESCAPE sequences as G0, G1, G2 or G3 sets.

Two types of DRCS have been identified. The first type is the basic DRCS. Only the shapes of the characters are down-loaded. Characters are displayed on the screen in the prevailing foreground colour on the prevailing background colour. In the second type of DRCS the down-loaded characters are completely defined in foreground colours, i.e. all the dots of a character cell have a defined foreground colour, chosen from a number of colours.

The protocol defined below for down-loading of DRCS allows down-loading of both types of DRCS. The protocol is open ended to allow for future extensions.

The down-loading of DRCS is accomplished using units of two types:

- DRCS header units;
- DRCS pattern transfer units.

A DRCS header unit describes the general properties of the DRCS to be loaded. The actual pattern transfer takes place using DRCS pattern transfer units. Both units are coded as Videotex Presentation Data Elements (VPDE) in accordance with the Presentation Level Data Syntax (PLDS) as:

US 2/3 Y <data>

Y = 2/0 indicates DRCS header units

Y_2/0 indicates DRCS pattern transfer units.

C.2 DRCS Header Units

A DRCS header unit applies for all following pattern transfer units until the header is redefined or until the end of a session. A DRCS header unit is coded as:

US 2/3 2/0 <ICS> <SDC> <SCM> <SSA>

The various fields of the DRCS header unit are coded with bytes from different columns of the code table. If the default conditions apply these fields are omitted. The default conditions are specified in the description of the fields. They are independent of previously loaded header units.

C.2.1 <ICS>: Identification of Character Set

The <ICS> field identifies the DRCS to be loaded, by a number which will consequently be used in the designation sequence for this set. With the exception of the last byte, all the bytes of the <ICS> field are taken from column 2 of the code table.

<ICS> : 2/k F

k: Indicates whether the set belongs to the first or second repertory in the library (see clause C.5 for the designation sequence). It also indicates whether a possibly existing DRCS identified by the same <ICS> field should be deleted or merely be overwritten by the following pattern transfer units:

- 0: first repertory, do not delete existing DRCS;
- 1: second repertory, do not delete existing DRCS;
- 8: first repertory, delete existing DRCS;
- 9: second repertory, delete existing DRCS.

F: If the DRCS is registered in ISO 2375, then F is the character allocated by ISO.

If the DRCS is a non-registered set, then the sequence 2/0 Fx is transmitted. Fx can be taken from columns 4 to 7 of the code table.

The default for <ICS> is: 2/0 2/0 4/0, which will identify a non-registered set of the first repertory to be loaded in the library with final character for designation 4/0. An existing DRCS in this library position will not be deleted.

C.2.2 <SDC>: Select Dot Composition

The <SDC> field describes the structure of the cells of the DRCS to be loaded. The <SDC> field also discriminates between the two types of DRCS.

There are two alternative types of <SDC>.

C.2.2.1 SDC Type 1

The first type is coded with bytes from column 3 of the code table. This is the extended type of <SDC>. Its coding will allow for future extension of the DRCS architecture.

<SDC> type 1: <character cell> <blocking factor> <pixel characteristics>

<character cell> : 3/th 3/uh 3/11 3/tv 3/uv 3/11

th: (0,1 ... 9): tens of horizontal pixels, leading zeros may be omitted¹²;

uh: (0,1 ... 9): units of horizontal pixels, no default for the number of horizontal pixels;

tv: (0,1 ... 9): tens of vertical pixels, leading zeros may be omitted¹²;

uv: (0,1 ... 9): units of vertical pixels, default number of vertical pixels = 10.

<blocking factor>: indicates the grouping of character cells (horizontal × vertical) for a rectangular character block. This character block is considered as a single character cell during the character description. When down-loaded in the terminal a character block occupies $h \times v$ consecutive character positions in the DRCS. The coding of the <blocking factor> is: 3/th 3/uh 3/11 3/tv 3/uv 3/11.

th: (0,1 ... 9): tens of horizontal character cells, leading zeros may be omitted¹²;

uh: (0,1 ... 9): units of horizontal character cells;

tv: (0,1 ... 9): tens of vertical character cells, leading zeros may be omitted¹²;

uv: (0,1 ... 9): units of vertical character cells.

¹²) The coding scheme allows more significant digits to be added if needed.

The default coding for the <blocking factor> is 3/1 3/11 3/1 3/11, indicating a character block of 1 × 1 character cell.

<pixel characteristics>: for further study. The default condition is 1 bit/dot/pixel basic DRCS.

C.2.2.2 SDC Type 2

The second type of <SDC> is coded with bytes from columns 4 and 3 of the code table. This is the shorthand type of <SDC>.

<SDC> type 2: 4/p <blocking factor> 4/q

p: indicates envisaged recommended dot matrix sizes (horizontal × vertical). There is no default for p.

0: 16 × 24 8: 8 × 12

1: 16 × 20 9: 8 × 10

2: 16 × 12 10: 6 × 12

3: 16 × 10 11: 6 × 10

4: 12 × 24 12: 6 × 5

5: 12 × 20 13: 4 × 10

6: 12 × 12 14: 4 × 5

7: 12 × 10 15: 6 × 6

<blocking factor>: as above

q: indicates the number of bits per dot used to code the DRCS. The default for q is 1, indicating 1 bit per dot basic DRCS. Colour DRCS is coded with q _ 1 :

1: 1 bit/dot basic DRCS

2: 2 bit/dot colour DRCS, 4 colours

default DCLUT: black, red, green, yellow

3: 3 bit/dot colour DRCS, 8 colours

default DCLUT: 1st colour palette

4: 4 bit/dot colour DRCS, 16 colours

default DCLUT: 1st and 2nd colour palettes (or 3rd and 4th colour palettes if the 1st and 2nd colour palettes are not redefinable).

The colours as mentioned above can be modified by loading the DRCS Colour Look Up Table (DCLUT). See Part 5, redefinable colours.

Since there is no default for some bytes in the <SDC> a DRCS header unit must always contain at least part of an <SDC>. If <SDC type 1> is used, at least 3/uh and 3/11 must be contained in the header; if <SDC type 2> is used, 4/p must be included in the header.

C.2.3 <SCM>: Select Coding Method

The <SCM> field determines the way in which the DRCS patterns are coded as they are down-loaded. Details of the coding are also determined by the <SDC> field (e.g. matrix size, bits per dot, blocking factor, pixel characteristics). The bytes of the <SCM> field are taken from column 5 of the code table (see C.4.1.3).

<SCM>: <type> [<sub-type>]

<type> : 5/t (default t = 0)

<sub-type>: 5/st

The <type> field identifies the coding method of the DRCS to be down-loaded. Some coding methods require a <sub-type> field to identify options within the coding method.

- t = 0: 'direct' coding, described in C.4.1
no <sub-type> needed
- t = 1: 'Runlength' coding, described in C.4.2
the default coding for st = 0

Other coding methods are for further study.

C.2.4 <SSA>: Select Set Attributes

The <SSA> field describes the actions which certain attributes will have on the DRCS characters once they are displayed on the screen. The bytes of the <SSA> field are taken from column 6 of the code table (see C.4.1.3).

The details of the <SSA> are for further study.

The default for <SSA> is such that the LINED attribute causes an underline, as for alphanumeric characters, but has no effect on colour DRCS.

C.3 Pattern Transfer Units

Pattern transfer units are coded as:

US 2/3 Y <pattern data>

Y: the code of the first character (or character block) described in the unit; it has a value in the range 2/1 to 7/14 inclusive.

The <pattern data> field of a pattern transfer unit describes the patterns for the characters of the down-loaded DRCS, in accordance with the last received DRCS header unit.

The value of the Y parameter defines the code of the first defined character. If the pattern transfer unit contains more character definitions, they will be assigned subsequent codes. Data contained in a pattern transfer unit for a character subsequent to a character with code 7/14 will be discarded.

The coding methods to be used in the <pattern data> are described in C.4 below.

C.4 Coding Methods

In the following subclauses the recommended coding methods, as indicated in the <SCM> field of the DRCS header units, are defined.

C.4.1 'Direct' Coding Method

The 'direct' coding method is identified by t = 0 (default value) in the <SCM> field of the DRCS header unit. No <sub-type> is needed for this coding method. The method can be used to load basic DRCS as well as colour DRCS.

C.4.1.1 Basic DRCS

A DRCS character cell consists of m dots horizontally and n dots vertically (in total $m \times n$ dots). The values of m and n are determined by the <SDC> field of the DRCS header unit. The direct coding method can be used for all possible values of m and n.

The dots of a character are coded using bytes from columns 4 to 7 of the code table, these bytes are called D bytes. The dots are loaded six dots at a time, row by row, starting from the top left hand corner, using the six least significant bits. Dots defined as '1' are displayed in foregroundcolour.

To improve the efficiency of this code a number of special commands have been added. They are coded as bytes from column 2 of the code table (see C.4.1.3) and are called S-bytes. The coding of these bytes are:

<i>Code</i>	<i>Name</i>	<i>Description</i>
2/0	Sf	Fill rest of character with '0's
2/1	R1	Repeat last complete row once
2/2	R2	Repeat last complete row twice
2/3	R3	Repeat last complete row 3 times
2/4	R4	Repeat last complete row 4 times
2/5	R5	Repeat last complete row 5 times
2/6	R6	Repeat last complete row 6 times
2/7	R7	Repeat last complete row 7 times
2/8	R8	Repeat last complete row 8 times
2/9	R9	Repeat last complete row 9 times
2/10	R10	Repeat last complete row 10 times
2/12	S1	Defines a complete row containing '1's
2/14	Sr	Fill rest of character with last complete row
2/15	Ss	Fill rest of character with '1's
3/0	B1	Start of pattern block for new character

The pattern block for each DRCS character is preceded by the command B1 (3/0).

The insertion of an S-byte may leave a number of remaining bits in the previous D-byte, which will not define a complete row. The use of these bits are explained below.

The actions of the Sf (2/0) command are as follows. The remaining bits of the last D-byte are used as the first bits of the next row; the rest of this row and the possibly remaining rows of the character are filled with 0's. The action of the Ss (2/15) command is equivalent, but with the character filled with '1's.

The Sr (2/14) command causes the last complete row to be copied in the remaining rows of the character. Remaining bits in the last D-byte are discarded.

For the remaining commands S0 (2/12), S1 (2/13) and R1 (2/1) to R10 (2/10) the processing of the remaining bits in the last D-byte is postponed until the action indicated by the command is executed. Together with the next D-byte (or Sf or Ss) these bits are used for the definition of the remaining part of the character. If the rest of the character is completely defined by the commands mentioned in this paragraph, the remaining bits are discarded.

The extent of the repeat command cannot cross the border of a character block. If a repeat command is used as the first byte of a character definition (i.e. the first byte after a B1 command), the action is as if the last complete row consisted of all '0's.

If a B1 command is received before a character is completely defined, the remaining part is defined as all 0's. Excess bytes before a B1 command are ignored.

C.4.1.2 Colour DRCS

In the pattern transfer units for colour DRCS, a number of bits per dot are downloaded to identify the colour of each dot. In the 'direct' coding method the pattern information for the DRCS is transmitted as one or more pattern blocks for each DRCS character. A pattern block defines one bit of each of the dots of the DRCS character as shown in Figure C.1. The pattern blocks are separated by separation bytes (B-bytes) coded from column 3 of the code table (see Table C.1).

<i>Code</i>	<i>Name</i>	<i>Description</i>
3/0	B1	Start of the 1st pattern block of a DRCS character Defining the least significant bit of the dot
3/1	B2	Start of the 2nd pattern block
3/2	B3	Start of the 3rd pattern block
3/3	B4	Start of the 4th pattern block

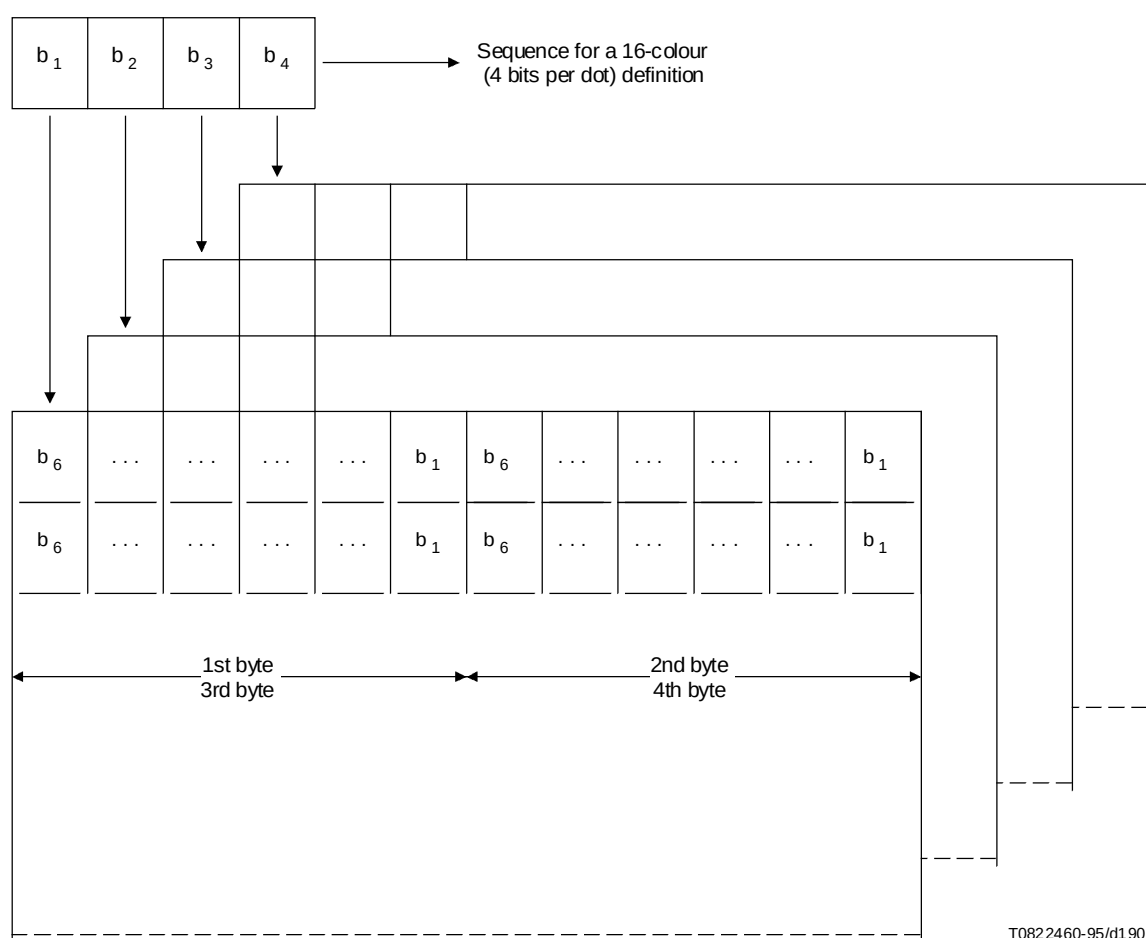


Figure C.1/T.101 – DRCS Downloading Format Definition for a 12 × 10 Dot Matrix

Equal pattern blocks only have to be transmitted once. In that case the pattern block is preceded by the two (or more) separation bytes to which the pattern block applies.

Examples of pattern transfer units for colour DRCS are given below:

Sixteen-colour DRCS (4 bits per dot)

```

US  2/3  Y  3/0 <1st pattern block> 3/1 <2nd pattern block>
      3/2 <3rd pattern block> 3/3 <4th pattern block> 3/0
...
US  2/3  Y  3/0 3/1 3/2 <pattern block> 3/3 2/0 3/0 ...

```

In the second example the pattern blocks for the first three bits of the dots are equal, while the fourth bit is '0'.

Four-colour DRCS (2 bits per dot):

US 2/3 Y 3/0 <1st pattern block> 3/1 <2nd pattern block> 3/0

...

C.4.1.3 Direct Coding Code Table

See Table C.1.

Table C.1/T.101 – Direct Coding Code Table

	0	1	2	3	4	5	6	7
0			Sf	B1	0	16	32	48
1			R1	B2	1	17	33	49
2			R2	B3	2	18	34	50
3			R3	B4	3	19	35	51
4			R4		4	20	36	52
5			R5		5	21	37	53
6			R6		6	22	38	54
7			R7		7	23	39	55
8			R8		8	24	40	56
9			R9		9	25	41	57
10			R10		10	26	42	58
11					11	27	43	59
12			S0		12	28	44	60
13			S1		13	29	45	61
14			Sr		14	30	46	62
15			Ss		15	31	47	63

S- B- <-----D-bytes ----->

C.4.2 'Runlength' Coding Method

The runlength coding method is identified by $t = 1$ in the <SCM> field of the DRCS header unit. In some cases a <sub-type> field is needed for this coding method. The default value for the <sub-type> is 5/0 ($st = 0$).

The runlength coding method can be used for basic DRCS as well as colour DRCS, although the method may be best used for advanced types of colour DRCS using a non-default <blocking factor>.

The following general rules apply for runlength coding.

Runlength coding uses the six least significant bits from bytes of columns 4 to 7 of the code table. These bits are identified as b_6, b_5, b_4, b_3, b_2 and b_1 (b_1 is the least significant).

Runlength coding is applied on character blocks as defined by the <SDC> field of the header unit (default 1×1), row by row, starting from the top left hand corner of the block.

If a runlength exceeds the right hand border of the character block, the remaining part of the runlength is continued on the next row. If it exceeds the right hand border of the last row of the character block the remaining part is ignored.

C.4.2.1 Basic DRCS

Two types of runlength coding for basic DRCS are specified. The first type is identified by $st = 0$ (default). In this case the runlength is coded with three bits:

b_6, b_5, b_4 : runlength for the background colour

b_3, b_2, b_1 : runlength for the foreground colour

The coding for each runlength is:

<i>Code</i>	<i>Length</i>
001	0
010	1
011	2
100	3
101	4
110	5
111	6
000	escape

If the escape code is used, the six bits of the following byte are completely used to code the runlength (1 to 63). If both runlengths in a byte are coded as escape, the second byte will contain the runlength of the background colour and the third byte the runlength of the foreground colour.

The second type of runlength coding for basic DRCS is defined by $st = 1$. In this case the coding is:

b_6 : 0 runlength for background colour

1 runlength for foreground colour

b_5, b_4, b_3, b_2, b_1 : runlength (1 to 31)

C.4.2.2 Colour DRCS

For colour DRCS the runlength is coded per colour.

In the case of sixteen-colour DRCS the runlength coding will be:

b_6, b_5, b_4, b_3 : colour definition

b_2, b_1 : runlength

01 length 1

10 length 2

11 length 3

00 escape

If the escape code is used, the six bits of the next byte define the runlength (1 to 63).

For eight-colour DRCS the runlength coding will be:

b ₆ , b ₅ , b ₄ :	colour definition
b ₃ , b ₂ , b ₁ :	runlength (1 to 7)
000	escape

For four-colour DRCS the runlength coding will be:

b ₆ , b ₅ :	colour definition
b ₄ , b ₃ , b ₂ , b ₁ :	runlength (1 to 15)
0000	escape

C.5 Designation and Invocation of DRCS

Once a DRCS (or part of it) is downloaded, the set is considered part of the library. The set can then be designated by the ESC-sequence.

ESC I F

$I = 2/k + i$

k: 8 or 12 indicating the first or second repertory. The value for k should be in accordance with the value for k in the <ICS> field of the header unit of the required DRCS.

i: 0, 1, 2 or 3 depending on whether the set is designated as a G0, G1, G2 or G3 set respectively.

F: If the DRCS is registered in ISO 2375, then F is the character allocated by ISO.

If the DRCS is a non-registered set then the sequence 2/0 Fx is transmitted, where Fx is equal to Fx in the <ICS> field of the header unit of the required DRCS.

Once the set is designated, it can be invoked in the normal manner.

If a character block is to be displayed on the screen, the top left hand character cell will be positioned at the active position. After the display of the block, the active position will be in the next position following the top right hand character cell of the block.

C.6 Applicability of Attributes

Unless the <SSA> field of the DRCS header unit defines otherwise, all attributes shall apply in the normal way to DRCS characters, the only exception being that the LINED attribute is not applicable to colour DRCS.

Although in colour DRCS a character is completely defined in foreground colour, it should be remembered that at the position where a colour DRCS character is displayed there is a defined background colour, which should be applied in the case of, for example, the INVERT or the FLASH attribute.

Part 5 – Define COLOUR

CONTENTS

	<i>Page</i>
C.1 Introduction	443
C.2 Colour System Extension.....	443
C.3 Coding of Redefinable Colours	444
C.3.1 COLOUR Header Unit	444
C.3.2 COLOUR Reset Unit	445
C.3.3 COLOUR Transfer Units	445

C.1 Introduction

The alphamosaic C1 sets provide for the selection of eight colours. In this part the method used to extend this colour system and to redefine colours will be described.

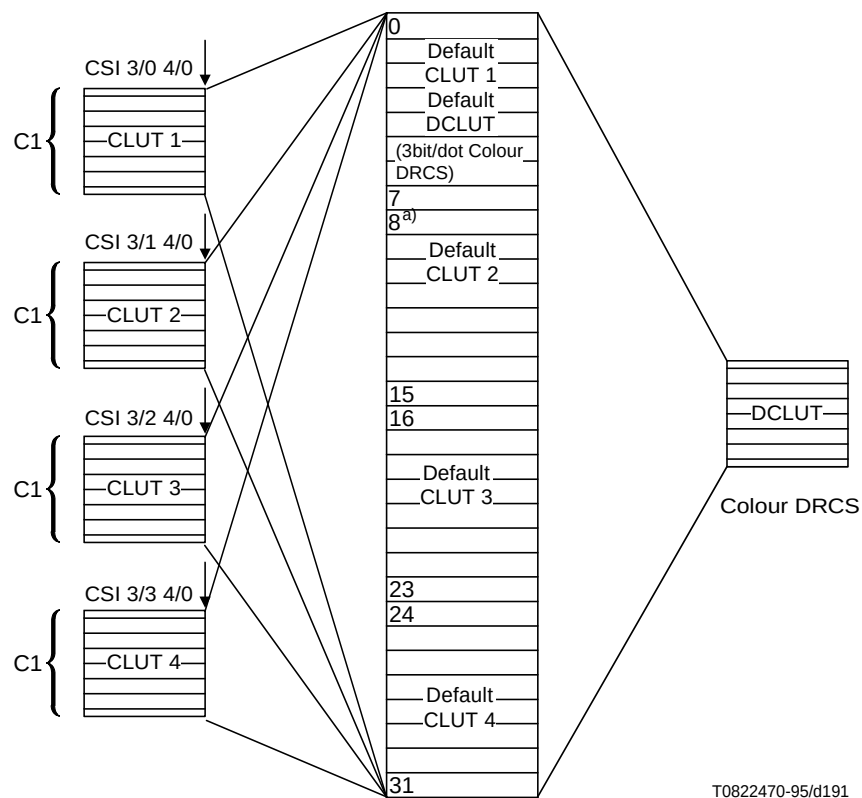
C.2 Colour System Extension

The extension of the colour system is accomplished by providing a number of colour tables of eight colours each. At a given instant only one table can be in use. This table is selected using a CSI sequence (see Part 1, C.2.3.13). Each table is implemented as a Colour Look Up Table (CLUT) with eight entries. The entry in the 'in use' CLUT is selected using the C1 controls according to the table below.

<i>C1 control colour</i>	<i>Entry No. in CLUT</i>
black	0
red	1
green	2
yellow	3
blue	4
magenta	5
cyan	6
white	7

The entry in the CLUT contains an ordinal number in the colour map. The contents of this colour map entry define the colour. In the alphamosaic mode four CLUTs are used. They are named CLUT1, CLUT2, CLUT3 and CLUT4. The size of the colour map is 32 entries, divided into 4 parts of 8 entries each.

For colour DRCS (see Part 4) separate look up tables called DCLUTs may be provided. A DCLUT contains a number of entries which are used to define the colours used in colour DRCS. The colour extension scheme is shown schematically in Figure C.1.



T0822470-95/d191

^a) If this entry (No. 8) is defined as BLACK (as it is by default), it will be interpreted as TRANSPARENT.

Figure C.1/T.101 – Colour Extension Scheme

C.3 Coding of Redefinable Colours

The Define COLOUR VPDE is used to redefine the contents of the colour map, or to redefine the contents of the CLUTs or the DCLUTs. The coding is:

US 2/6 Y <data>

Y: determines the function of the Define COLOUR VPDE

2/0: define COLOUR header unit

2/1: define COLOUR reset unit

3/x: COLOUR transfer unit

C.3.1 COLOUR Header Unit

A COLOUR header unit applies for all following colour transfer units until the header is redefined or until the end of a session. The header unit is coded as:

US 2/6 2/0 <ICT> <SUR> <SCM>

<ICT>: Identification of Colour Table, is coded as: 2/a I

a: indicates the type of colour table

0: colour map

1: CLUT

2: DCLUT

- I: indicates the number of the unit indicated in 2/a.
I is in the range 2/0 to 7/15.
The default coding for <ICT> is 2/0 2/0, identifying the colour map No. 1.
 - <SUR>: Select Unit Resolution, is coded as: 3/c
 - c: (1, 2...9) indicating the number of bits used to define each unit of the identified table.
The default value for <SUR> is 3/4.
 - <SCM>: Select Coding Method, is coded as: 4/d
 - d: indicates the coding method
 - 0: entries in colour map
 - 1: load colour map using R,G,B
- The default value for <SCM> = 4/1.
If necessary an extra byte (coded 4/e) may be added.

C.3.2 COLOUR Reset Unit

The COLOUR Reset Unit is used to reset all the colour tables (CLUTs, DCLUTs and colour map) to their default values. The reset unit is coded as:

US 2/6 2/1

C.3.3 COLOUR Transfer Units

COLOUR transfer units are used to load colour tables. The colour table to be loaded and the loading method used are defined by the 'Define COLOUR' header unit. The COLOUR transfer units are coded as:

US 2/6 Y <colour data>

Y: will indicate the first table entry to be loaded, and is coded as: 3/t 3/u

t: (0,1...9) tens of address, leading zeros may be omitted

u: (0,1...9) units of address¹³⁾

<colour data>: bytes in the range of 4/0 to 7/15.

The meaning of the <colour data> depends on the preceding 'Define COLOUR' header unit and is defined in the subclauses below.

C.3.3.1 Loading a CLUT or DCLUT

This function is identified by the last received 'Define COLOUR' header unit with <ICT> = 2/1 (CLUT) or 2/2 (DCLUT). The least significant <SUR> bits are taken from each byte of the colour data and stored in consecutive locations of the identified CLUT or DCLUT, starting at the address indicated by Y.

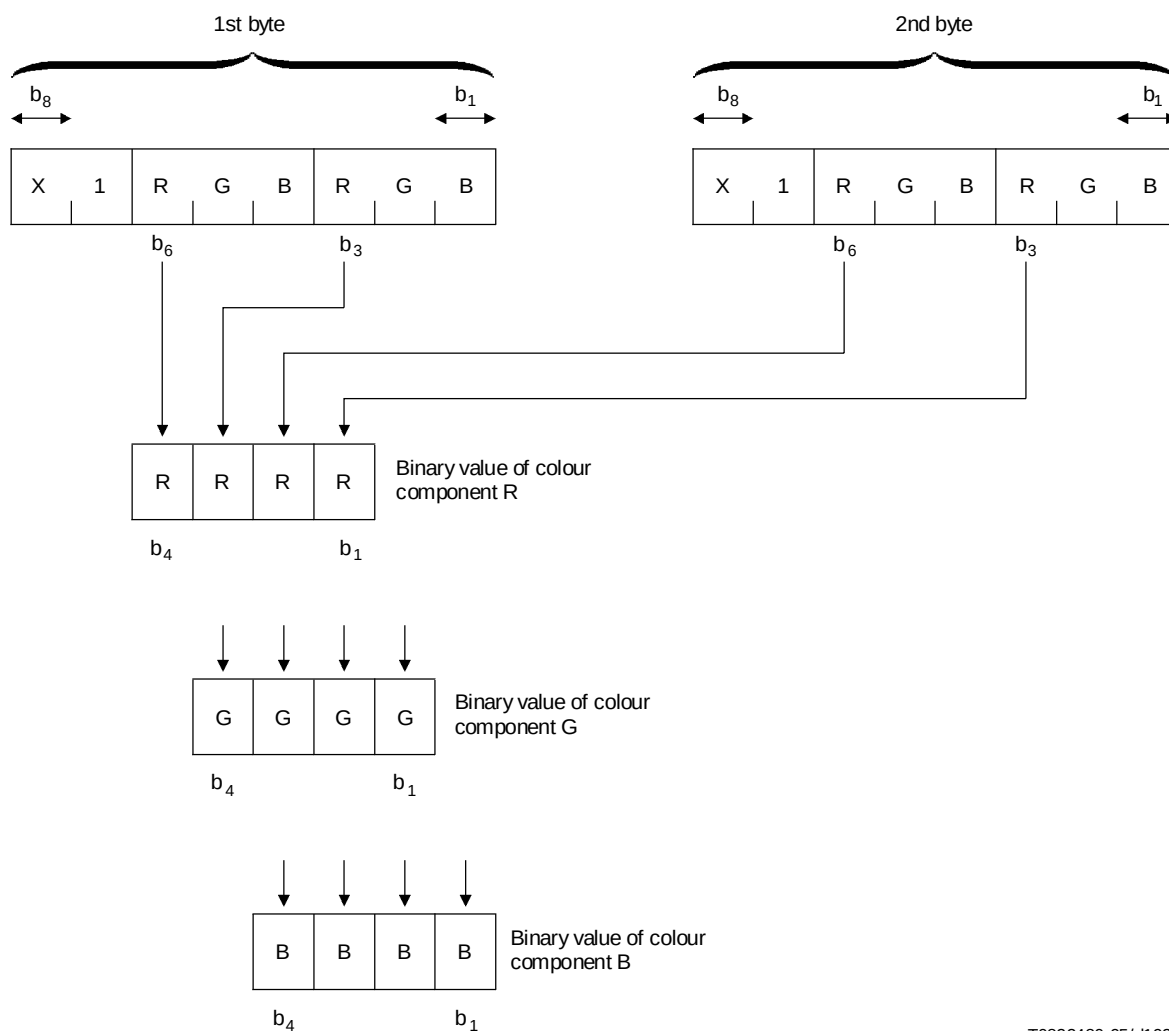
Data received for addresses outside the identified CLUT or DCLUT will be discarded.

C.3.3.2 Loading the Colour Map using R, G, B

This function will be identified by the last received 'Define COLOUR' header unit with <ICT> = 2/0 and <SCM> = 4/1, or the default header unit. The colour map is loaded starting at the address indicated by Y. The colours are defined in terms of their Red, Green and Blue components, each of which is defined by <SUR> bits.

Each <colour data> byte contains two bits for each of the primary colours, except for the last byte, which may contain only one relevant bit for each primary colour. The coding of the six least significant bits of the <colour data> bytes is: R G B R G B, the most significant bits defining the more significant bits of the colour components. A value of '0' for a colour component indicates zero intensity. All bits '1' indicate full intensity. Intermediate values are interpreted in equal brightness steps (eye corrected).

¹³⁾ The coding scheme allows more significant digits to be added if necessary.



T0822480-95/d192

Figure C.2/T.101 – Example of COLOUR Transfer in the case of Downloading Values of RGB with 4 bits each

Part 6 – Define FORMAT

CONTENTS

	<i>Page</i>
C.1 Introduction	447
C.2 Coding	447
C.3 Defaults.....	447

C.1 Introduction

The 'Define FORMAT' VPDE is used to define the number of rows and columns displayed within the defined display area for the alphamosaic display.

The default alphamosaic display format is 24 rows of 40 characters.

The possibility of changing the aspect ratio of the defined display area using the 'Define FORMAT' VPDE is for further study.

When the 'Define FORMAT' VPDE is executed no assumptions can be made about the subsequent contents of the screen.

C.2 Coding

The coding of the 'Define FORMAT' VPDE is as follows:

US 2/13 Y <CH> <CT> <CU> 3/11 <RH> <RT> <RU> 3/11 <WC>

If Y is 4/1 to 4/14 one of the following formats is defined:

4/1: 40 columns by 24 rows
4/2: 40 columns by 20 rows
4/3: 80 columns by 24 rows
4/4: 80 columns by 20 rows
4/5: 48 columns by 20 rows
4/6: 40 columns by 25 rows (numbered from 0 to 24)
4/7: 20 columns by 12 rows (for Chinese characters)
4/8: 20 columns by 8 rows (for Chinese characters)
4/9 to 4/14: are for further study

If Y is 4/15 the number of columns and rows is defined by the following data, where:

<CH> <CT> <CU> is the number of columns in hundreds, tens and units, coded from column 3 (leading zeros may be omitted);¹⁴⁾

<RH> <RT> <RU> is the number of rows in hundreds, tens and units, coded from column 3 (leading zeros may be omitted).¹⁴⁾

<WC> is used to define the wraparound controls. <WC> takes the following values:

7/0: wraparound ON
7/1: wraparound OFF

C.3 Defaults

The default 'Define FORMAT' VPDE is:

US 2/13 4/1 7/0

¹⁴⁾ The coding scheme allows more significant digits to be added if necessary.

Part 7 – TRANSPARENT Data

CONTENTS

	<i>Page</i>
C.1 Introduction	448
C.2 Protocol.....	448

C.1 Introduction

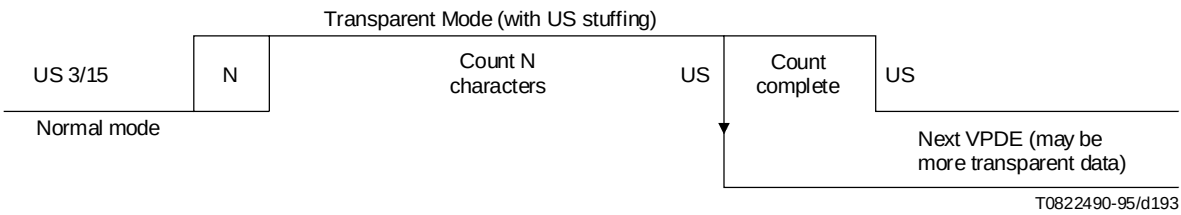
Certain Videotex applications such as geometric and photographic displays contain a relatively large amount of data. Consequently, it is desirable for increased efficiency to use all the presentation level code bits for actual data (7 or 8 bits per byte). In such a mode all codes pass uninterrupted by the normal presentation level control codes and the mode is thus, termed transparent.

C.2 Protocol

The “TRANSPARENT data” VPCE is used to enter transparent mode. There are two methods of leaving the transparent mode, either following a byte count (where a byte is 8 bits), or when a new VPCE is detected. Immediately exit by a new VPCE ensures “RESET to service break to row X” will operate.

When a US (01 15) code (which is used to signify the start of a new VPCE) appears naturally in the data, it should be transmitted twice (this technique is known as byte stuffing). A new VPCE is detected by a single US in the data stream.

The value of the first “N” bytes in the transparent mode data indicates the normal method of exit from the transparent mode. When “N” is zero, then no byte count is defined and transparent mode is only exited at the start of a new VPCE. If “N” has a value of between 1 and 254 inclusive, then this value specifies the number of bytes that is to be received before a return is automatically made to the previous VPCE. The transparent mode byte count is performed on received bytes after stuffing bytes have been removed.



T0822490-95/d193

N = 0 exit by new VPCE.
N = 1 to 254 inclusive exit following a byte count or by new VPCE.
N = 255 is not defined.

Part 8 – RESET

CONTENTS

	<i>Page</i>
C.1 Introduction	449
C.2 Coding Structure	449
C.3 Functional Description and Coding	449
C.3.1 General Display Reset.....	449
C.3.2 Reset Control and Graphic Sets	450
C.3.3 Service Break to Row X	450

C.1 Introduction

This function is used to set predefined states in the terminal and thereby synchronize the Videotex service and the terminal at the presentation layer.

C.2 Coding Structure

The coding structure for the Reset function is as follows:

US 2/15 <operation> <parameter>

<operation>: This character indicates the display mode being reset, and the operation required. This character is coded from columns 2 to 4 of the code table.

<parameter>: This character is coded from columns 4 to 7 of the code table. Its meaning depends upon the reset operation.

C.3 Functional Description and Coding

C.3.1 General Display Reset

By means of the General Display Reset the terminal can be brought into a defined condition, regardless of the terminal mode. It's function is to combine the actions of the Reset operation for the Alpha-mosaic display, the Geometric display, the Photographic display, etc. It's actions are summarized below:

- The actions as described in C.3.2 (Reset Control and Graphic Sets) are executed.
- The format is set to default to 24 rows of 40 characters.
- The Alpha-mosaic DDA shall be filled with spaces.
- The active position is set to the first character position of the first row.
- All attributes of the Alpha-mosaic display are set to their default values as described in Part 1, C.1.5.2.
- The actions on the geometric display (refer to T/C-06-02) can be described as follows:

General Display Reset is equivalent to the execution of the following sequence of primitives:

- DEACTIVATE WORKSTATION (ws_id) for all workstations.
- CLOSE WORKSTATION (ws_id) for all workstations.
- OPEN WORKSTATION (ws_id) for ws_id = 0.
- ACTIVATE WORKSTATION (ws_id) for ws_id = 0.
- SET DEFAULTS.
- SET WORKSTATION WINDOW (ws_id, P1, P2).
ws_id = 0, P1 = (0.0, 0.0), P2 = (1.0, 0.75).

- SET WORKSTATION VIEWPORT (ws_id, XMIN, XMAX, YMIN, YMAX).
ws_id = 0, XMIN = 0.0, XMAX = 1.0, YMIN = 0.0, YMAX = 0.75.
NOTE 1 – OPEN WORKSTATION (ws_id) ensures that the display surface is cleared.
- For the Photographic display all the Photographic tables are cleared and the Photographic display is set to transparent. (This is for further study.)
- The action of a Timing Control Wait VPDE is terminated and the data received since the start of the Wait is deleted.
- The data following the General Display Reset function is to be interpreted as Alpha-mosaic data.

The coding of the General Display Reset is as follows:

- a) The serial C1 set is invoked
US 2/15 4/1
- b) The parallel C1 set is invoked
US 2/15 4/2

NOTE 2 – This command will be extended to reset other terminal display functions yet to be defined.

C.3.2 Reset Control and Graphic Sets

The actions of this function are as follows:

- the default graphic sets, as described in Part 1, C.1.5.6, are designated;
- in the 7-bit environment the G0 set is invoked into columns 2 to 7 of the code table;
- in the 8-bit environment the G0 set is invoked into columns 2 to 7 of the code table and the G2 set is invoked columns 10 to 15 of the code table;
- data following the Reset Control and Graphic Sets is to be interpreted as Alpha-mosaic data,

and

- a) The serial C1 set is invoked
US 2/15 4/3
or
- b) The parallel C1 set is invoked
US 2/15 4/4

C.3.3 Service Break to Row X

This function affects the terminal from the time it is received until the next US command is received. The terminal resets to the previous state before the next US command is executed. The action of this function is as follows:

Previous display states, including character sets, colours attribute controls and the active position will be stored in the terminal but no longer active.

Downloading processes to the terminal will be terminated.

The primary set of graphic characters is designated the G0 set and the supplementary set of graphic characters is designated the G2 set. Other character sets are not affected.

In the 7-bit environment the G0 set is invoked into columns 2 to 7 of the code table.

In the 8-bit environment the G0 set is invoked into columns 2 to 7 of the code table and the G2 set is invoked into columns 10 to 15 of the code table.

The format is unaffected but wrap-around is inactive.

The active position is set to the first character position of the designated row.

Only the following controls of the primary control function set are valid:

- in the 7-bit environment:
APB, APF, APR, CAN, SS2, ESC (in combination with a character from columns 4 or 5 of the code table), US.
- in the 8-bit environment:
APB, APF, APR, CAN, US.

If the serial C1 set is invoked the following controls are invalid:

- in the 7-bit environment:
5/0 to 5/7, 5/11, 5/14, 5/15.
- in the 8-bit environment:
9/0 to 9/7, 9/11, 9/14, 9/15.

The protected area attribute is inactive, all other attributes are unchanged.

Colour look up table is active.

Data following is interpreted as Alpha-mosaic.

If a Service Break to Row X is received by a terminal while it is executing a Timing Control Wait VPDE, the Service Break is executed immediately, after which the original wait will be continued.

Coding: US 2/15 <CS> (RN)

<CS> The designated C1 set is coded as follows:

4/0 = serial C1-set

4/5 = parallel C1-set

(RN) 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 significant bits. If the row X specified is outside the D.D.A, following data is displayed on the bottom row of the D.D.A.

To reset to the previous state following a Service Break the following coding is defined:

US 2/15 4/15

This function should follow a Service Break to Row X. It is only valid used in this way.

The protected area attribute is inactive, all other attributes are unchanged.

Previous display states, including character sets, colours, attribute controls and the active position will be restored by the function Reset to the Previous States.

Part 9 – Telesoftware

CONTENTS

	<i>Page</i>
SECTION 1 – INTRODUCTION	453
C.1 Foreword.....	453
C.2 Scope	453
C.3 Related Recommendations and standards.....	453
SECTION 2 – SERVICE, APPLICATIONS, PROTOCOLS, CODING	454
C.1 Service definition.....	455
C.1.1 Scope and field of application.....	455
C.1.2 Model of the T-SERVICE.....	455
C.1.3 Association regime control	461
C.1.4 Access regime control.....	466
C.1.5 Transfer regime control.....	475
C.1.6 Exception	479
C.1.7 Collisions	480
C.2 Telesoftware and auxiliary device applications	481
C.2.1 Preliminaries	481
C.2.2 The telesoftware application organisation	482
C.2.3 Printer application organisation	483
C.2.4 Files.....	483
C.2.5 Description of the transfer file structure	485
C.2.6 Use of the T-SERVICE for telesoftware and printer device application	488
C.2.7 The designation field in a directory request.....	492
C.3 T-Protocol specification.....	494
C.3.1 Definitions.....	494
C.3.2 Description and use of TDU	495
C.3.3 Exceptions and timers	507
C.3.4 Use of DDU layer	508
C.3.5 Use of Syntax-based-videotex	508
C.4 Coding of TDUs	508
C.4.1 Coding of TDUs.....	508
C.4.2 Coding of provider and user refusals	521
C.4.3 File coding	522
C.4.4 Coding of parameters for the telesoftware and printer device applications	532
C.5 D-Protocol specification	534
C.5.1 Definitions.....	534
C.5.2 General overview of the protocol.....	534
C.5.3 Repertoire and use of Dialogue Data Units.....	536
C.5.4 Error detection and recovery mechanism.....	539
C.5.5 Coding.....	542
Appendix I – A compression algorithm	552

SECTION 1 – INTRODUCTION

C.1 Foreword

This annex describes the videotex processable data enabling symmetrical file transfer. This annex includes as a subset a basic kernel which provides only for file transfer from host to terminal.

Section 1, Section 2 and all the notes contained in these two sections form an integral part of this annex.

C.2 Scope

The videotex processable data facility specified in this annex is intended to be used for data file transfer. These files may contain computer software or other file types.

This protocol may be used to download files from the host to the terminal. It may also be used for transferring files between two end systems in both directions, all the operations being then performed under the control of one or the other system depending on a preliminary negotiation.

It has been defined to work in a videotex environment but it may also work without specific application environment.

C.3 Related Recommendations and standards

- CCITT T.101 (Rev. 1994), *International Interworking For Videotex Services*.
- ETS 300 072, *Terminal Equipment (TE) Videotex presentation layer protocol, Videotex presentation layer data syntax*.
- ETS 300 079 (1991), *Integrated Services Digital Network (ISDN) – Syntax – based videotex, end-to-end protocols*.
- ETS 300 223, *Terminal Equipment (TE) Syntax – based videotex, end-to-end protocols*.

SECTION 2 – SERVICE, APPLICATIONS, PROTOCOLS, CODING

General introduction

The Videotex processable data facility specified in this annex provides particularly for the transfer of files of data; these files may contain computer software, but other file types are not precluded. This facility also provides for data to be reliably transferred to devices associated with a videotex terminal under control of the received data. In addition to transparent transfer, specific standardized provisions are made for passing data to an associated printer, but the protocol is not limited to this device, and other devices may be standardized later.

Two service classes are currently defined in this annex:

- A “basic kernel” provides for file transfer from a host to a terminal, all the operations being controlled by the host. The access to a file or to a specific set of files is carried out by performing a videotex dialogue which takes place outside the downloading phase and which is not part of the current specification. Moreover, this basic kernel provides for low level recovery mechanisms.
- An enhanced service, called “symmetrical service”, provides for transferring files between two systems in both directions, all the operations being able to be performed under the control of one or the other systems according to a preliminary negotiation. The access to a file or to a specific set of files is carried out with a dialogue phase which can be completely automated and which is part of the downloading protocol. In this service class enhanced facilities are available: recovery, user identification, window mechanism.

These two service classes allow to support a wide range of applications which require file transfer. Two applications are currently defined in this annex: a telesoftware application and a file transfer application intended for printing. These applications are specified as rules for the use of the T-service and as structure and coding of the exchanged files.

Section 1 of this annex describes the functions which are offered to an application using the processable data facility. These functions are described in terms of service elements.

In order to allow for the transfer of files in already existing videotex systems as well as allowing more advanced facilities for future systems, the transfer is described as consisting of two layers.

Processable data, including telesoftware files, data for printing, file parameters and control data related to the downloading procedure are transmitted by means of Telesoftware Data Units (TDU's). These TDU's are exchanged between cooperating entities according to the T-protocol. This protocol, as well as the specific rules for telesoftware and printer device applications are specified in C.3.

In addition, Dialogue Data Units (DDU's) are used for adaptation to the different Videotex systems. Optional 8-bit transparency capabilities and optional error detection and recovery facilities are provided. Handling of these DDU's are described in C.5.

In the case of syntax based videotex, the DDU protocol is not used.

Figure C.1 describes the theoretical model for handling processable data.

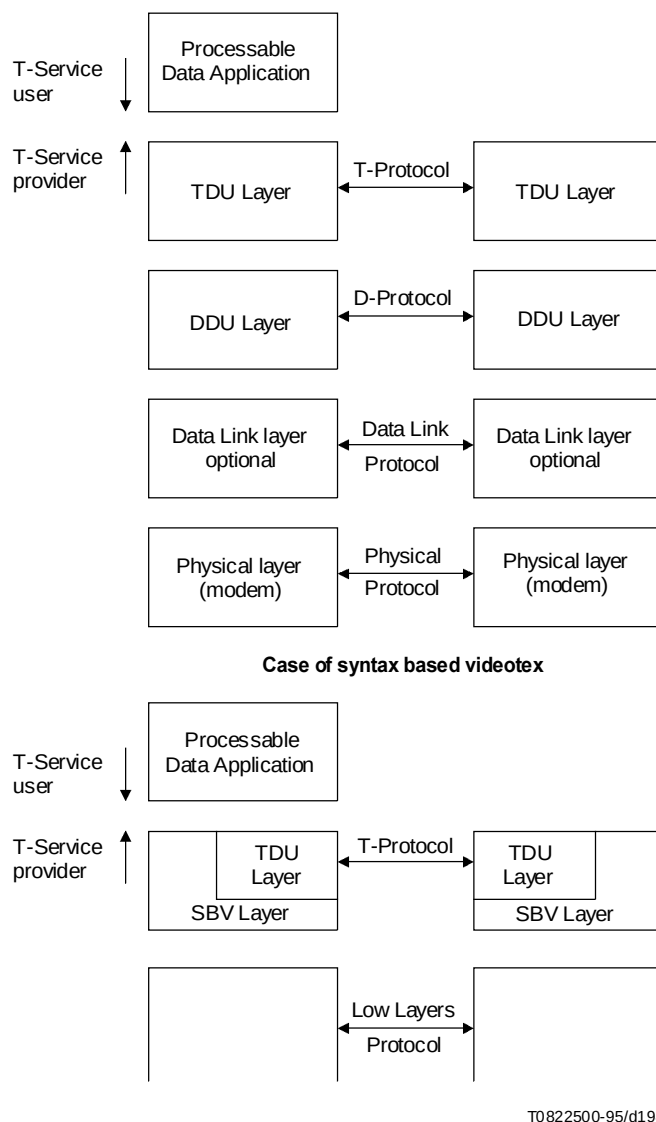


Figure C.1/T.101 – Theoretical model of processable data handling

C.1 Service definition

C.1.1 Scope and field of application

This subclause defines in an abstract way the externally visible service provided by the TDU layer (T-service) in terms of:

- a) the primitive actions and events of the service;
- b) the parameter data associated with each primitive action and event;
- c) the relationship between, and the valid sequence of these actions and events.

This subclause also describes the processable data applications which make use of the above mentioned service.

C.1.2 Model of the T-SERVICE

C.1.2.1 Services references

This annex is based on the concepts which were developed in CCITT and ISO for the description of the OSI Reference model (CCITT Rec. X.200 | ISO 7498).

The conventions used to describe this service are based on CCITT Recommendation X.210 (OSI layer service definition conventions).

C.1.2.2 Services definitions

C.1.2.2.1 General terms

This subclause makes use of the following terms as they are defined in Recommendation X.210:

- a) service user;
- b) service provider;
- c) primitive;
- d) request;
- e) indication;
- f) response;
- g) confirmation;
- h) confirmed, non-confirmed, provider initiated service.

The following definitions also apply:

optionally confirmed service: confirmed or non-confirmed service according to the user's choice specified in the request primitive.

protocol phase: period of time during which the exchanges are dedicated to a specific function (connection, disconnection, mass transfer, etc.).

regime: set of protocol phases; a Regime is a continuous period of time. A Regime is established by using a confirmed or optionally confirmed service and it is orderly terminated by using a confirmed service, it may also be interrupted in an abnormal manner.

A regime is fully defined by specifying the service(s) used to establish it and the service(s) used to terminate it. A regime is used in this description to limit the range of some services which may only be available during a particular regime.

transfer unit: data transferred by using one mass transfer primitive.

initiator: the entity which initiates a service request.

acceptor: the entity which accepts (or refuses) a service indication.

C.1.2.2.2 Regimes

Three regimes are defined: the Association regime, the Access regime and the Transfer regime. In the basic kernel only the Association and the Transfer regimes may be established.

These regimes are defined in C.1.2.3.

In the following, the functions of the regimes and their relationship with each other are specified.

An Association regime determines a period during which two applications remain associated.

An Access regime is used to allow functions to be negotiated and it determines a period during which these functions are available. The Access regime is never established in "basic kernel".

A Transfer regime determines a period during which a mass transfer is performed.

The mass transfer phase is related to the data transfer itself and takes place during a Transfer regime.

An Access regime is established within an Association regime, provided no other Access regime is already established. A Transfer regime is established within an Access regime (symmetrical service) or within an Association regime (basic kernel), provided no other Transfer regime is already established.

Figures C.2 and C.3 show examples of establishing regimes in the basic kernel and in the symmetrical service.

C.1.2.2.3 Roles

At a given time each entity is assigned a unique role. Within a given regime, a role determines the set of services for which the entity may be initiator or acceptor.

The following roles are defined:

- The **Master** is the entity which controls the dialogue.

- The **Slave** is the entity which performs the operations requested by the Master.
- The **Sender** is the entity which sends the data during a mass transfer.
- The **Receiver** is the entity which receives the data during a mass transfer.

Mass transfer phases	***** ** ** *** *****
<i>Regimes</i>	
Transfer	----- --- ---- -----
Association	-----

Figure C.2/T.101 – Example of regimes establishment in the basic kernel

Mass transfer phases	***** ** *** *****
<i>Regimes</i>	
Transfer	----- --- ---- -----
Access	----- -----
Association	-----

Figure C.3/T.101 – Example of regimes establishment in the symmetrical service

During a whole Transfer regime a given entity keeps the same sender or receiver role.

After having established an Association regime the Master role is assigned to the initiating entity of the association establishment service, the Slave role is assigned to the accepting entity of the association establishment service.

At the Access regime establishment (in symmetrical service) the Master and Slave roles may be modified and will then remain unchanged during the whole Access regime. After the end of the Access regime the Master role is assigned to the initiating entity of the Access regime release service, the Slave role is assigned to the accepting entity of the Access regime release service.

At the Transfer regime establishment the Sender and Receiver roles are assigned according to the mass transfer direction. The mass transfer direction is determined by the service which has been used to establish the Transfer regime. At the end of the Transfer regime, each entity takes its own Master or Slave role which was assigned before the Transfer regime establishment.

In the service class “basic kernel”, the mass transfer is always performed in the same direction, therefore the Sender role is always assigned to the Master, the Receiver role is always assigned to the Slave.

C.1.2.2.4 Local concepts

- The **Server** is the system which contains a database (which stores and retrieves information without processing it). In the basic kernel the **Server** may be called the **Host**.
- The **Executor** is the system which processes the downloaded files.

A given system may contain both a Server application and an Executor application. The Server and Executor concepts are local concepts and are not related with file transmission, however this may impact implementation subsets definition.

In the service class “basic kernel”, during a whole association, the Server will play the Master and Sender roles, while the other entity is the Executor and will play the Slave and Receiver roles.

C.1.2.3 Service elements

C.1.2.3.1 General organisation

Table C.1 gives the list of the service elements:

Table C.1/T.101 – List of the service primitives

Service		Initiated by	Function	S	B
T-ASSOCIATE	OC	Both	Association establishment	M	M
T-RELEASE	C	Both (Note 1)	Association release	M	M
T-U-ABORT	NC	Both (Note 2)	Association user abort	M	M
T-P-ABORT	P	Both	Association provider abort	M	–
T-ACCESS	C	Master	Access regime establishment	M	–
T-END-ACCESS	C	Both	End of Access regime	M	–
T-DIRECTORY	C	Master	File Directory request	O	–
T-LOAD	C	Master	Init. Slave to Master mass transfer	O	–
T-SAVE	C	Master	Init. Master to Slave mass transfer	O	–
T-RENAME	C	Master	Rename file	O	–
T-DELETE	C	Master	Delete file	O	–
T-TYPED-DATA	NC	Both	Typed data transfer	O	–
T-WRITE	OC	Sender	Data transfer	M	M
T-WRITE-END	C	Sender	End of data transfer	M	M
T-U-EXCEPTION	NC	Both (Note 3)	User exception report	M	M
T-P-EXCEPTION	P	Both	Provider exception report	M	–
C Confirmed service OC Optionally confirmed service NC Non-confirmed service P Provider initiated service B Basic kernel S Symmetrical service O Optional M Mandatory – Irrelevant NOTES 1 May only be sent by the Master in the basic kernel. 2 May only be sent by the Slave in the basic kernel. 3 May only be sent by the Receiver in the basic kernel.					

Figure C.4 gives the relationship between services and regimes, indicating which services are used to establish and terminate a regime and which services are available in a given regime. The definition of the regimes is given in the following subclauses.

C.1.2.3.2 Association regime

The Association regime is established by using the association establishment service. It may be terminated by using the association release service or the association abort services.

In the idle state the Master may invoke the association abort or release services, the Access (symmetrical service) or Transfer (basic kernel) regime establishment services.

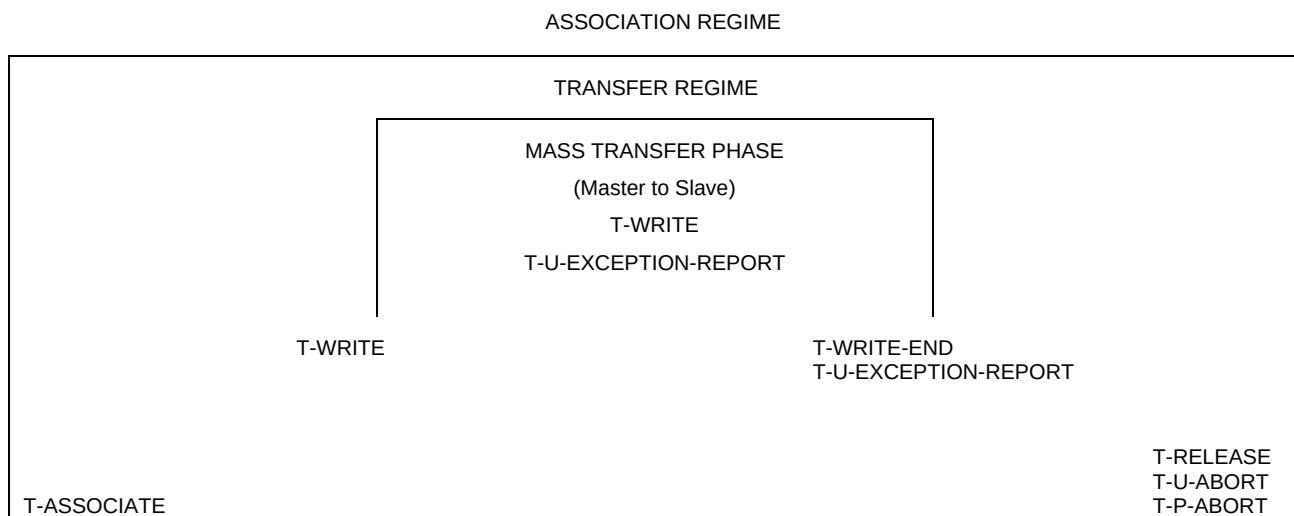
In the idle state the Slave may invoke the association abort or release services.

C.1.2.3.3 Access regime

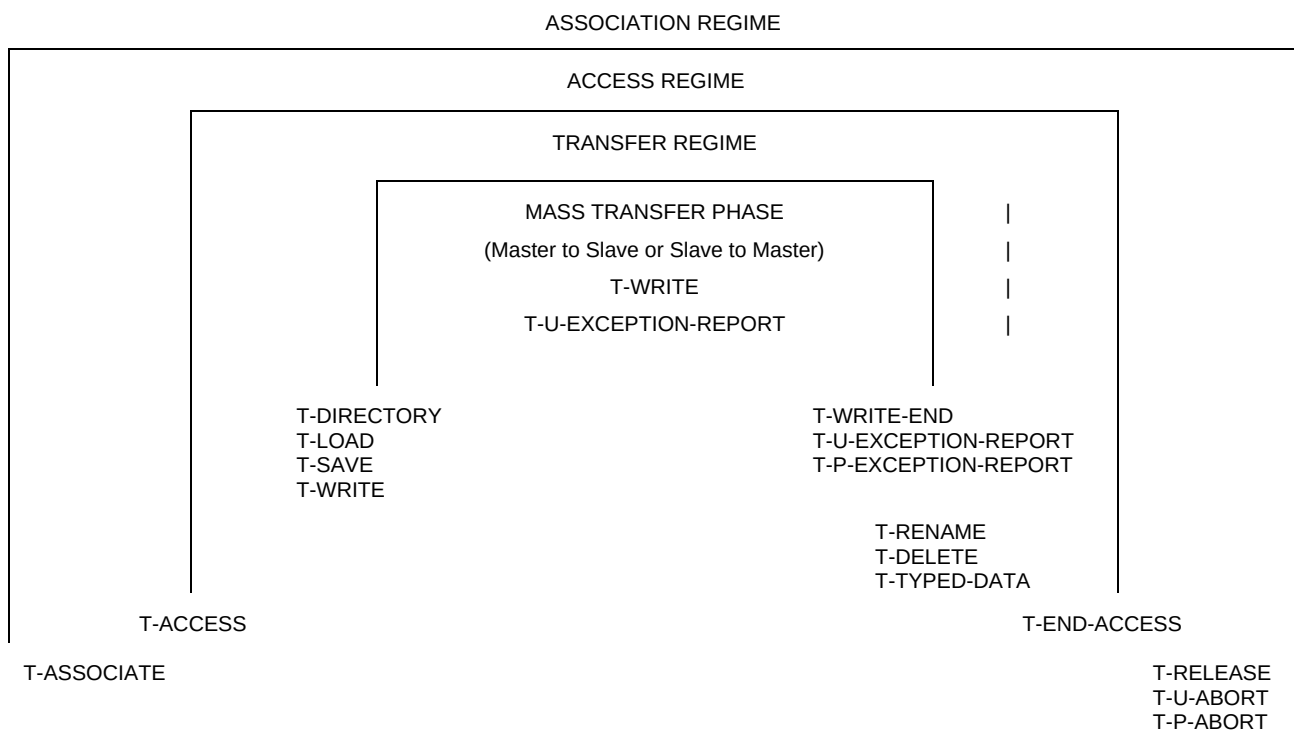
The Access regime is established (in symmetrical service) by using the Access regime establishment service. It may be terminated by using the end of Access regime service or the association abort service (in this latter case the association is also terminated).

When the Access regime is established, the Master and the Slave may invoke the end of Access regime service, the association abort service or the exception report services. The Master may invoke the data transfer, load, save, rename, delete, file directory services, the Master and the Slave may also invoke the typed data transfer service provided that the use of those services had been negotiated during the Access regime establishment.

a) Basic Kernel



b) Symmetrical service



NOTE – T-U-ABORT and T-P-ABORT may be used at any time in every regime, they terminate all the currently established regimes.

Figure C.4/T.101 – Regimes

C.1.2.3.4 Transfer regime

The Transfer regime is used to transmit a large amount of information (e.g. files) from the Sender to the Receiver. The mass transfer phase, which consists in performing the data transfer and end of data transfer services, takes place during the transfer regime.

In the symmetrical service class, a Transfer regime may be established within the Access regime by using the load, save or file directory services. When the Access regime is established, the Transfer regime may also be established implicitly by the Master by starting the mass transfer phase. When a Transfer regime consists only in a mass transfer phase (i.e. it is directly established by issuing a data transfer service primitive) the transfer is called Basic Transfer Mode. In the symmetrical service class, the use of the Basic transfer mode is negotiated at the Access regime establishment and is exclusive of the use of file directory, load or save services.

In the basic kernel, a Transfer regime is reduced to the mass transfer phase (Basic transfer mode). Therefore, the Transfer regime is established when the Association regime is established by starting the mass transfer phase.

A Transfer regime may be terminated by using the end of data transfer service, the exception report service or the association abort services.

During the Transfer regime, the Sender may invoke the data transfer service, the end of data transfer service, the exception report services or the association abort services. The Receiver may invoke the exception report services and the association abort services. In the symmetrical service only the Receiver may invoke the user exception report service.

C.1.2.3.5 Restrictions on the use of services

Subclauses C.1.2.3.2, C.1.2.3.3, C.1.2.3.4 specify which services may be used in each regime.

Moreover, between sending a confirmed service request and receiving the corresponding confirmation (at the service initiator side) or between receiving a confirmed service indication and sending the corresponding response (at the service acceptor side), no other service may be initiated except the exception report, or association abort services. This restriction is not applicable to the data transfer services when the window size is greater than 1 and in the conditions described in C.1.2.4.3.

C.1.2.4 Concepts related to mass transfer

In order to facilitate the transfer of a large amount of information various mechanisms are provided.

The main notion is the recovery point. The recovery points are located at the beginning of the mass transfer phase and at each confirmed data transfer primitive.

In the symmetrical service, the size of transfer units (conveyed within a data transfer primitive) is negotiated during the Access regime establishment.

C.1.2.4.1 Recovery during a mass transfer

Several facilities are provided to recover during a mass transfer phase:

- The Receiver may request to restart the transmission at the beginning of the mass transfer phase. This facility is always available in the basic kernel and it is negotiated during the Access regime establishment in the symmetrical service.
- During the mass transfer, the transmission may be resumed from the last confirmed data transfer primitive, this may be done by sending a negative response to a data transfer primitive. The transmission is resumed from the data which immediately followed the last data for which a positive confirmation had been received by the Sender or, if not, at the beginning of the transfer.

C.1.2.4.2 Recovery outside a mass transfer

The recovery outside a mass transfer phase is carried out when the mass transfer phase has been interrupted. This recovery may take place during the same association or during another association than the original transfer. This mechanism is only available in the symmetrical service when using Load or Save services and if it is negotiated during the Access regime establishment.

When the recovery outside a mass transfer has been negotiated, the Sender must associate a recovery point number to each transfer unit and each data transfer primitive must be confirmed.

NOTE 1 – The service does not ensure that the information for which the transmission is resumed are consistent with the previously transmitted information. It is up to the service user to provide means for a correct recovery (e.g. storage of the interrupted transfer context, file version number, etc.).

The recovery request is performed by indicating a recovery point number in a parameter of the T-SAVE request or T-LOAD request primitives. The transmission is resumed starting with the data which immediately followed the indicated recovery point in the original transfer. The indicated recovery point corresponds to the last data transfer primitive for which a positive confirmation had been received or sent.

NOTE 2 – After a transfer interruption, it may happen that the recovery point number, from the Sender point of view, will be lower than the recovery point number from the Receiver point of view. It is up to the Receiver to verify that no data duplication occurs in case of recovery.

C.1.2.4.3 Anticipation window

The anticipation mechanism gives the ability to send several data transfer primitives with explicit confirmation requested, without waiting for having received the confirmation of the previous ones. The aim of the window mechanism is to improve the transmission efficiency by avoiding idle periods due to the waiting of confirmations.

The anticipation window is the number of recovery point (i.e. the number of data transfer primitives with explicit confirmation requested) which the Sender may send without having received any confirmation. When there is no anticipation the window size is equal to 1.

When the anticipation window size is greater than 1, the Sender must associate a recovery point number to each transfer unit and each data transfer primitive must be confirmed.

In the basic kernel the window size is always equal to 1.

In symmetrical service, the window size is negotiated during the Access regime establishment. Each entity indicates the window size it may accept when the entity plays the role of Receiver. However, the Sender may use, for sending, a window of a lower value than the value indicated by the Receiver (i.e. the Sender is not bound to fill in the sending window). The Receiver must confirm the recovery points “as soon as possible” and should not wait until the window maximum value is reached to confirm them (this value may never be reached if the Sender uses a lower window size for sending).

When the value of the maximum window size is reached the Sender is no longer permitted to send data until it receives an explicit confirmation to a previous recovery point. The recovery points are set by incrementing the recovery point number by one for each new recovery point and they are explicitly confirmed (one by one) in the increasing order of their reception.

C.1.3 Association regime control

If the association establishment service requires confirmation, the association regime is established as soon as the acceptor has sent a T-ASSOCIATE Response (positive) and, at the initiator end, a T-ASSOCIATE Confirmation (positive) has been received.

If the association establishment does not require confirmation, the association regime is established as soon as the initiator has sent a T-ASSOCIATE Request and, at the acceptor’s end, a T-ASSOCIATE Indication has been received.

The association regime is terminated upon transmission of T-RELEASE Response or T-U-ABORT Request and upon reception of T-RELEASE Confirmation, T-U-ABORT Indication or T-P-ABORT Indication.

C.1.3.1 Association establishment

C.1.3.1.1 Function

T-ASSOCIATE is used by a service user to associate with a specified processable data application (identified by the “application name” parameter).

The T-ASSOCIATE Request primitive is used to establish an Association regime, this primitive may not be used when the association is being established or is already established.

In the basic kernel, a T-ASSOCIATE request may only be initiated by the host.

T-ASSOCIATE is an optionally confirmed service, the parameter “explicit confirmation” indicates whether or not an explicit confirmation is requested.

C.1.3.1.2 Parameters

The different primitives and parameters required by the association establishment service are described in the following table.

Parameter	T-ASSOCIATE Request	T-ASSOCIATE Indication	T-ASSOCIATE Response	T-ASSOCIATE Confirm	S
Service class	Mandatory	Mandatory (=)	Mandatory	Mandatory (=)	
Called address	Optional	Optional (=)	Optional	Optional (=)	S
Calling address.	Optional	Optional (=)			S
Appl. name	Mandatory	Mandatory (=)			
Explicit confirmation	Mandatory	Mandatory (=)			
Timeouts	Optional	Optional (=)	Optional	Optional (=)	
Request ident.	Optional	Optional (=)			S
Identification	Optional	Optional (=)	Optional	Optional (=)	S
User data	Optional	Optional (=)	Optional	Optional (=)	
Result			Mandatory	Mandatory	
(=) The value of the parameter is identical to the value of the corresponding parameter in the preceding request or response primitive. S Symmetrical service only.					

C.1.3.1.2.1 Service class

This parameter is used to indicate the service class in use. It may take the following values:

- a) basic kernel;
- b) symmetrical service.

In the Request/Indication it indicates the service class(es) proposed.

In the Response/Confirmation it announces the service class selected if the association is accepted.

C.1.3.1.2.2 Called address

This parameter indicates the address of the entity who has sent a response or a confirmation. This address may differ from the parameter provided in the request or the indication (this can be the case after rerouting).

This parameter is not used if the service class parameter indicates only the value “basic kernel”.

C.1.3.1.2.3 Calling address

This parameter provides the address of the entity who originated the association.

This parameter is not used if the service class parameter indicates only the value “basic kernel”.

C.1.3.1.2.4 Application name

The parameter “application name” announces the use of a specified processable data application. This parameter is a variable length string.

The following standardized names are defined:

- !A: Any application (including telesoftware and printer).
- !T: Telesoftware.
- !P: Printer device.

Names starting with “!” are reserved for standardized applications.

C.1.3.1.2.5 Explicit confirmation

This parameter is used in the Basic Kernel to indicate whether or not explicit confirmation is requested.

If the service class parameter indicates “Symmetrical Service”, this parameter must be set to indicate that confirmation is requested.

C.1.3.1.2.6 Timeouts

In the basic kernel, this parameter indicates the master’s maximum response time and contains the maximum value of the delay between a response primitive issued by the slave and the next indication primitive. This delay is controlled by the service provider. Absence of this parameter means that this delay must not be controlled by the service provider.

In the symmetrical service, this parameter specifies the maximum response time required to answer a request. Each entity is responsible for defining their own response time. This response time is controlled by the service provider at the other end.

Absence of this parameter means in both cases, that the response time should not be checked by the service provider.

C.1.3.1.2.7 Request identification

This parameter enables the initiator to require identification in the Response/Confirmation. It may take the two following values:

- a) identification required,
- b) identification not required.

This parameter is not used if the service class parameter indicates only the value “basic kernel”.

C.1.3.1.2.8 Identification

This parameter is a variable length string, of no more than 254 bytes. It provides identification data on the user of this service.

This parameter is not used if the service class parameter indicates only the value “basic kernel”.

C.1.3.1.2.9 User data

This parameter is used to convey string type information of no more than 254 bytes.

In the basic kernel this parameter cannot be used in the T-ASSOCIATE response and T-ASSOCIATE confirmation.

C.1.3.1.2.10 Result

This parameter indicates whether the association is accepted or rejected.

If the selected service class is the symmetrical service, possible values of this parameter are the following:

- a) Association request accepted.
- b) Association request rejected. Reason not specified.
- c) Reject from the service provider, reason:
 - called address incorrect;
 - calling address incorrect;
 - service class refused.

- d) Reject from the service user, reason:
- called address incorrect;
 - calling address incorrect;
 - service class refused;
 - application’s name unknown;
 - wrong identification;
 - erroneous user data;
 - other reason.

If the selected service class is the basic kernel, the value of this parameter can be either:

- a) association request accepted; or
b) association request rejected.

C.1.3.1.3 Association establishment operation

In the case when an explicit confirmation is requested by the parameter “explicit confirmation”, the corresponding response primitive should be sent by the receiver before any other protocol exchange.



If explicit confirmation is not requested and if the application specified in a T-ASSOCIATE is not available at the acceptor’s side or in the case of any user error, then the acceptor should use the T-U-ABORT service.

C.1.3.2 Association release

C.1.3.2.1 Function

An Association regime may be terminated by the exchange of T-RELEASE primitives. This service may be used only once the Association regime is established and provided no other regime is established.

The T-RELEASE service is thus used to request the orderly termination of the processable data application.

If the selected service class is the basic kernel only the Master may issue a T-RELEASE request.

C.1.3.2.2 Parameters

The different primitives and parameters required by the association release service are described in the following table.

Parameter	T-RELEASE Request	T-RELEASE Indication	T-RELEASE Response	T-RELEASE Confirmation
User data	Optional	Optional (=)	Optional	Optional (=)
Result			Mandatory	Mandatory (=)
(=) The value of the parameter is identical to the value of the corresponding parameter in the preceding request or response primitive.				

C.1.3.2.2.1 User data

This parameter is used to convey string type information of no more than 254 bytes.

In the basic kernel, this parameter cannot be used in the T-RELEASE response and T-RELEASE confirmation.

C.1.3.2.2.2 Result

This parameter always indicates that the T-Release service has been accepted.

C.1.3.2.3 Association release operation



C.1.3.3 Association abort

C.1.3.3.1 Function

This service is performed by the primitives T-U-ABORT Request when it is user initiated, by T-P-ABORT when it is provider initiated.

In the basic kernel a T-U-ABORT request may only be initiated by the slave. T-P-ABORT is not used in the basic kernel.

A T-U-ABORT request may be sent by the user of the T-Service at any time after reception of a T-ASSOCIATE Indication or after transmission of a T-ASSOCIATE Request. The provider of the T-Service may issue a T-P-ABORT Indication at any time after reception of a T-ASSOCIATE Request or after transmission of T-ASSOCIATE Indication.

It is recommended to avoid the use of the T-U-ABORT service to reject a T-ASSOCIATE for which explicit confirmation was requested.

These primitives terminate the association regime abnormally. Data may be lost.

C.1.3.3.2 Parameters

The different primitives and parameters required by the abort service are described in the following table.

Parameter	T-U-ABORT request	T-U-ABORT indication
Reason		Mandatory

Reason (user):

- reason not specified;
- incorrect identification;
- incorrect role/function;
- other reason.

If the service class is the basic kernel this parameter is always equal to “reason not specified”.

Parameter	T-P-ABORT indication
Reason	Mandatory

Reason (provider):

- reason not specified;
- repeated negative acknowledgements/repeated errors;
- delay expired;
- unknown message;
- protocol conflict;
- lower layer error;
- syntax error/parameter’s absence;
- other reason.

C.1.3.3.3 Association abort operation



C.1.4 Access regime control

The Access regime is established upon transmission of the primitive T-ACCESS Response (positive) on the acceptor’s side and upon reception of the primitive T-ACCESS Confirmation (positive) on the initiator’s side.

The Access regime is terminated upon transmission of the primitive T-END-ACCESS Response or T-U-ABORT Request (on the acceptor’s side) and upon reception of the primitive T-END-ACCESS Confirmation, T-U-ABORT Indication or T-P-ABORT Indication (on the initiator’s side).

C.1.4.1 Access establishment

C.1.4.1.1 Function

The T-ACCESS service enables the establishment of the Access regime provided the service class selected is the symmetrical service.

The establishment of this regime is initiated by the primitive T-ACCESS Request. This primitive may only be issued by the Master in the Association regime and provided no other Access regime is established.

The establishment of the Access regime can be rejected by the acceptor by use of the primitive T-ACCESS Response. In such case, after completing the T-ACCESS service, both the entities are idle in the Association regime.

In case the initiator of the T-ACCESS primitive is unsatisfied with the answer, it may issue a T-END-ACCESS Request primitive thus terminating the Access regime and supplies the reason of the termination.

C.1.4.1.2 Parameters

The different primitives and parameters required by the Access service are described in the following table.

Parameters	T-ACCESS Request	T-ACCESS Indication	T-ACCESS Response	T-ACCESS Confirmation
Role	Mandatory	Mandatory (=)		
Functions	Mandatory	Mandatory (=)	Mandatory	Mandatory (=)
Transfer unit size	Mandatory	Mandatory (=)	Mandatory	Mandatory (=)
Anticipation window	Mandatory	Mandatory (=)	Mandatory	Mandatory (=)
Recovery	Mandatory	Mandatory (=)	Mandatory	Mandatory (=)
Transfer Mode	Mandatory	Mandatory (=)	Mandatory	Mandatory (=)
User data	Optional	Optional (=)	Optional	Optional (=)
Result			Mandatory	Mandatory (=)
(=) The value of the parameter is identical to the value of the corresponding parameter in the preceding request or response primitive.				

C.1.4.1.2.1 Role

This parameter can take either of the two following values:

- a) Master;
- b) Slave.

This parameter indicates the role adopted by the initiator of the T-ACCESS Request primitive. This role will replace the role previously assigned as soon as the Access regime is established.

C.1.4.1.2.2 Function

This parameter indicates whether or not the user handles the following Indication primitives:

- *Slave*: T-DIRECTORY, T-LOAD, T-SAVE, T-RENAME, T-DELETE, T-TYPED DATA, T-U-EXCEPTION (Read Restart).
- *Master*: T-TYPED DATA, T-U-EXCEPTION (Read Restart).

If the Basic Transfer mode is selected (see C.1.4.1.2.6), then the T-DIRECTORY, T-LOAD and T-SAVE services must not be used.

C.1.4.1.2.3 Size of transfer units

This parameter indicates the maximum size of the application data contained in a T-WRITE or T-WRITE-END Request primitive.

The following values are permitted:

512, 1024, 2048, 4096, 8192, 16384, 32768, 65528 bytes

In the basic kernel this parameter is not used and the value is 1024 bytes.

C.1.4.1.2.4 Anticipation window

This parameter indicates the maximum number of consecutive T-WRITE or T-WRITE-END (confirmation required) primitives the user is allowed to receive before issuing an answer.

It may take a value ranging from 1 to 8.

This parameter is not used in the basic kernel, the value is 1.

C.1.4.1.2.5 Recovery

This parameter indicates whether or not the slave is able to handle the recovery mechanism upon transmission and reception of files.

C.1.4.1.2.6 Transfer mode

This parameter may take one of the two following values:

- a) Basic Transfer Mode supported (Slave) or Basic Transfer Mode required (Master);
- b) Basic Transfer Mode not supported (Slave) or Basic Transfer Mode not required (Master).

If the Master indicates that it requires the use of the Basic Transfer Mode and the Slave indicates that it supports this mode, then the Basic Transfer Mode is selected which means that only this mode may be used to transfer files and that the Load, Save or Directory functions must not be used during the Access regime whatever the Slave indicated in the Function parameter. Otherwise the Basic Transfer Mode must not be used (only the other functions indicated in the Function parameter may be used).

C.1.4.1.2.7 User data

This parameter is used to convey string type information of no more than 254 bytes.

C.1.4.1.2.8 Result

This parameter indicates whether or not the service has been accepted or rejected.

This parameter may take one of the following values:

- a) Access request accepted.
- b) Access request rejected. Reason not specified.
- c) Refusal from the service provider, reason:
role refused.
- d) Refusal from the service user, reason:
 - role refused;
 - insufficient primitives handled;
 - erroneous user data;
 - other reason.

C.1.4.2 End of Access service

C.1.4.2.1 Function

The Access regime may be terminated using a T-END-ACCESS primitive. The T-END-ACCESS Request primitive may only be issued within the idle state of the Access regime.

It may be initiated by both the Master and the Slave.

C.1.4.2.2 Parameters

The different primitives and parameters required by the End Access service are described in the following table.

Parameters	T-END-ACCESS Request	T-END-ACCESS Indication	T-END-ACCESS Response	T-END-ACCESS Confirmation
User data	Optional	Optional (=)	Optional	Optional (=)
Result			Mandatory	Mandatory (=)
(=) The value of the parameter is identical to the value of the corresponding parameter in the preceding request or response primitive.				

C.1.4.2.2.1 User data

This parameter is used to convey string type information of no more than 254 bytes.

C.1.4.2.2.2 Result

This parameter indicates the reason of the Access regime termination.

- reason not specified;
- insufficient primitives handled;
- other reason.

C.1.4.3 File directory service

C.1.4.3.1 Function

The File directory service enables the Master to request the transfer of one file directory from the Slave to the Master and the establishment of the Transfer regime. This service is not available in the Basic Kernel.

Only the Master may issue a T-DIRECTORY Request, provided the Access regime is established and the use of T-DIRECTORY has been accepted by the Slave at the time the Access regime was established.

The information is transferred from the Slave to the Master after the Slave has answered by T-DIRECTORY Response (accept). A positive response establishes the Transfer regime.

In the Transfer regime the Slave becomes Sender and the Master Receiver.

C.1.4.3.2 Parameters

The different primitives and parameters required by the File directory service are described in the following table.

Parameters	T-DIRECTORY Request	T-DIRECTORY Indication	T-DIRECTORY Response	T-DIRECTORY Confirmation
Designation	Mandatory	Mandatory (=)	Mandatory	Mandatory (=)
User data	Optional	Optional (=)		
Result				
(=) The value of the parameter is identical to the value of the corresponding parameter in the preceding request or response primitive.				

C.1.4.3.2.1 Designation

This parameter enables the Master to designate the directory it requires. It is a byte sequence of no more than 254 bytes.

C.1.4.3.2.2 User data

This parameter is used to convey string type information of no more than 254 bytes.

C.1.4.3.2.3 Result

This parameter enables the Slave to indicate acceptance or refusal of the directory request.

This parameter may take one of the following values:

- a) acceptance;
- b) user refusal, reason:
 - reason not specified;
 - erroneous designation;
 - no answer to the request;
 - erroneous user data;
 - other reason.

C.1.4.4 Load service

C.1.4.4.1 Function

The Load service enables the Master to request the transfer of one file from the Slave to the Master and establishes the Transfer regime.

The Load service is not used if the Basic Transfer Mode is used (the Basic Transfer Mode is used in the Basic Kernel or as an option of the Symmetrical Service).

Only the Master may issue a T-LOAD Request, provided the Access regime is established and the use of T-LOAD has been accepted by the Slave at the time the Access regime was established.

The information is transferred from the Slave to the Master after the Slave has answered by T-LOAD Response (accept). A positive response establishes the Transfer regime.

In the Transfer regime the Slave becomes Sender and the Master Receiver.

C.1.4.4.2 Parameters

The different primitives and parameters required by the Load service are described in the following table.

Parameters	T-LOAD Request	T-LOAD Indication	T-LOAD Response	T-LOAD Confirmation
Recovery point	Optional	Optional (=)		
Designation	Mandatory	Mandatory (=)		
User data	Optional	Optional (=)		
Result			Mandatory	Mandatory (=)
(=) The value of the parameter is identical to the value of the corresponding parameter in the preceding request or response primitive.				

C.1.4.4.2.1 Recovery point

This parameter is used to enable the Master to indicate the recovery point which corresponds to the last data transfer primitive for which a positive confirmation had been sent. The Transfer will be resumed from the data transfer service immediately following the indicated recovery point. The value of the recovery point is in the range 0 to 65535.

This parameter cannot be used if the recovery function has not been accepted at the time the Access regime was established.

C.1.4.4.2.2 Designation

This parameter enables the Master to designate the information it requires. It is a byte sequence of no more than 254 bytes.

C.1.4.4.2.3 User data

This parameter is used to convey string type information of no more than 254 bytes.

C.1.4.4.2.4 Result

This parameter enables the Slave to indicate acceptance or refusal of the load request.

This parameter may take one of the following values:

- a) acceptance;
- b) user refusal, reason:
 - reason not specified;
 - erroneous designation;
 - unknown file;
 - erroneous recovery point;
 - erroneous user data;
 - other reason.

C.1.4.5 Save service

C.1.4.5.1 Function

The Save service enables the Master to request the transfer of one file from the Master to the Slave and establishes the Transfer regime. This service is not available in the Basic Kernel.

Only the Master may issue a T-SAVE Request, provided the Access regime is established and the use of T-SAVE has been accepted by the Slave at the time the Access regime was established.

The information is transferred from the Master to the Slave after the Slave has answered by T-SAVE Response (accept). A positive response establishes the Transfer regime.

In the Transfer regime the Slave becomes Receiver and the Master Sender.

C.1.4.5.2 Parameters

The different primitives and parameters required by the Save service are described in the following table.

C.1.4.5.2.1 Recovery point

This parameter is used to enable the Master to indicate the recovery point which corresponds to the last data transfer primitive for which a positive confirmation had been received. The Transfer will be resumed from the data transfer service immediately following the indicated recovery point. The value of the recovery point is in the range 0 to 65535.

This parameter cannot be used if the recovery function has not been accepted at the time the Access regime was established.

Parameters	T-SAVE Request	T-SAVE Indication	T-SAVE Response	T-SAVE Confirmation
Recovery point	Optional	Optional (=)		
Designation	Mandatory	Mandatory (=)		
User data	Optional	Optional (=)		
Result			Mandatory	Mandatory (=)
(=) The value of the parameter is identical to the value of the corresponding parameter in the preceding request or response primitive.				

C.1.4.5.2.2 Designation

This parameter enables the Master to designate the information it wants to save. It is a byte sequence of no more than 254 bytes.

C.1.4.5.2.3 User data

This parameter is used to convey string type information of no more than 254 bytes.

C.1.4.5.2.4 Result

This parameter enables the Slave to indicate acceptance or refusal of the save request.

This parameter may take one of the following values:

- a) acceptance;
- b) user refusal, reason:
 - reason not specified;
 - erroneous designation;
 - file already exist;
 - erroneous recovery point;
 - erroneous user data;
 - other reason.

C.1.4.6 Rename service

C.1.4.6.1 Function

The Rename service enables the Master to rename the designation of a file within the Slave. This service is not available in the Basic Kernel.

Only the Master may issue a T-RENAME Request, provided the Access regime is established and the use of T-RENAME has been accepted by the Slave at the time the Access regime was established.

C.1.4.6.2 Parameters

The different primitives and parameters required by the Rename service are described in the following table.

C.1.4.6.2.1 New name

This parameter provides the new designation for the required information. It is a byte sequence of no more than 254 bytes.

Parameters	T-RENAME Request	T-RENAME Indication	T-RENAME Response	T-RENAME Confirmation
New name	Mandatory	Mandatory (=)		
Designation	Mandatory	Mandatory (=)		
User data	Optional	Optional (=)		
Result			Mandatory	Mandatory (=)
(=) The value of the parameter is identical to the value of the corresponding parameter in the preceding request or response primitive.				

C.1.4.6.2.2 Designation

This parameter enables the Master to designate the information it requires to be renamed. It is a byte sequence of no more than 254 bytes.

C.1.4.6.2.3 User data

This parameter is used to convey string type information of no more than 254 bytes.

C.1.4.6.2.4 Result

This parameter enables the Slave to indicate acceptance or refusal of the rename request.

This parameter may take one of the following values:

- a) acceptance;
- b) user refusal, reason:
 - reason not specified;
 - erroneous designation;
 - erroneous new name;
 - unknown file;
 - new name already in use;
 - erroneous user data;
 - other reason.

C.1.4.7 Delete service

C.1.4.7.1 Function

The Delete service enables the Master to delete a file within the Slave. This service is not available in the Basic Kernel.

Only the Master may issue a T-DELETE Request, provided the Access regime is established and the use of T-DELETE has been accepted by the Slave at the time the Access regime was established.

C.1.4.7.2 Parameters

The different primitives and parameters required by the Delete service are described in the following table.

Parameters	T-DELETE Request	T-DELETE Indication	T-DELETE Response	T-DELETE Confirmation
Designation	Mandatory	Mandatory (=)		
User data	Optional	Optional (=)		
Result			Mandatory	Mandatory (=)
(=) The value of the parameter is identical to the value of the corresponding parameter in the preceding request or response primitive.				

C.1.4.7.2.1 Designation

This parameter enables the Master to designate the information it requires to be deleted. It is a byte sequence of no more than 254 bytes.

C.1.4.7.2.2 User data

This parameter is used to convey string type information of no more than 254 bytes.

C.1.4.7.2.3 Result

This parameter enables the Slave to indicate acceptance or refusal of the delete request.

This parameter may take one of the following values:

- a) acceptance;
- b) user refusal, reason:
 - reason not specified;
 - erroneous designation;
 - unknown file;
 - erroneous user data;
 - other reason.

C.1.4.8 Typed data service

C.1.4.8.1 Function

The Typed data service enables the transfer of information either from the Slave or from the Master to the other entity, provided the Access regime is established and the use of T-TYPED DATA has been accepted at the time the Access regime was established. This service is not available in the Basic Kernel.

C.1.4.8.2 Parameters

The different primitives and parameters required by the Typed data service are described in the following table.

C.1.4.8.2.1 User data

This parameter is used to transmit information from the user of application transfer service and is a sequence of no more than 254 bytes.

Parameter	T-TYPED-DATA Request	T-TYPED-DATA Indication
User data	Mandatory	Mandatory (=)
(=) The value of the parameter is identical to the value of the corresponding parameter in the preceding request or response primitive.		

C.1.5 Transfer regime control

If the selected service class is the symmetrical class and if the Basic Transfer Mode has not been selected at the Access regime establishment and provided no other Transfer regime is established, the Transfer regime is established by the Master within the Access regime:

- a) on the acceptor's side upon transmission of one of the following primitives T-DIRECTORY Response (positive), T-LOAD Response (positive), or T-SAVE Response (positive);
- b) on the initiator's side upon reception of one of the following primitives T-DIRECTORY Confirmation (positive), T-LOAD Confirmation (positive), or T-SAVE Confirmation (positive).

If the selected service is the symmetrical class and if the Basic Transfer Mode has been selected at the Access regime establishment, and no other Transfer regime is established, the Transfer regime is established by the Master within the Access regime:

- a) on the initiator's side upon transmission of a T-WRITE (T-WRITE END in the case of a "short" file) Request primitive;
- b) on the acceptor's side upon reception of a T-WRITE (T-WRITE END in the case of a "short" file) Indication primitive.

If the selected service class is the basic kernel and no other Transfer regime is established, the Transfer regime is established by the Master within the Association regime:

- a) on the initiator's side upon transmission of a T-WRITE (T-WRITE END in the case of a "short" file) Request primitive;
- b) on the acceptor's side upon reception of a T-WRITE (T-WRITE END in the case of a "short" file) Indication primitive.

The Transfer regime is terminated upon transmission of a T-WRITE-END Response or a T-U-EXCEPTION Request (reject) primitive and upon reception of T-WRITE-END Confirmation, T-U-EXCEPTION Indication (reject) or T-P-EXCEPTION Indication.

C.1.5.1 Mass transfer

This service provides for confirmed and reliable transfer of data to virtual memory space in the Receiver.

The memory space is regarded as a sequence of bytes or as a sequence of records. The relation of this memory space to files, foreground memory, or other devices in the actual receiving equipment is defined in each case by the file attributes included in the file header.

C.1.5.1.1 Function

The T-WRITE service carries data from the required file. The data associated with each of these primitives is intended to immediately follow that of the previous T-WRITE in the virtual memory space.

The T-WRITE-END service conveys the last set of data of the required file. It is a confirmed service. A positive confirmation of this service terminates the Transfer regime.

Only the Sender may issue a T-WRITE Request or a T-WRITE-END Request primitive.

If the selected service class is the basic kernel only the Master may become the Sender.

T-WRITE is an optionally confirmed service. A recovery point is provided with each T-WRITE primitive if an explicit confirmation is requested.

C.1.5.1.2 Parameters for T-WRITE

Parameter	T-WRITE Request	T-WRITE Indication	T-WRITE Response	T-WRITE Confirmation	S
Explicit confirmation	Mandatory	Mandatory (=)			
First block	Mandatory	Mandatory (=)			
Block number	Optional	Optional (=)	Optional	Optional (=)	S
Data field	Mandatory	Mandatory (=)			
Result			Mandatory	Mandatory (=)	
(=) The value of the parameter is identical to the value of the corresponding parameter in the preceding request or response primitive. S Symmetrical service only.					

C.1.5.1.2.1 Explicit confirmation

This parameter is used to indicate whether or not explicit confirmation is requested.

This parameter must always be set to “explicit confirmation requested” as soon as any given entity suggest an anticipation window other than 1 or the recovery function.

C.1.5.1.2.2 First block

This parameter is used to indicate whether or not the data contained in the data field parameter consist in the beginning of the file.

C.1.5.1.2.3 Block number

In the basic kernel this parameter is absent.

In the symmetrical service, the block number is mandatory if the recovery mechanism is selected or the anticipation window is not 1. In this case every block is numbered and the block number is increased by 1 at each transmission of a new block. The block number is in the range 0 to 65535. The first block is 0.

If a block number is present in a request, it must be sent back in the response.

C.1.5.1.2.4 Data field

It is used to convey data from the file, it is a variable length string of no more than 1024 bytes in the basic kernel, and the maximum length is negotiated in the symmetrical service during the establishment of the Access regime.

C.1.5.1.2.5 Result

This parameter indicates, if the explicit confirmation has been requested, that the service has been accepted or rejected. The result parameter set to “reject” has the effect of restarting the transmission from the T-WRITE following the last confirmed T-WRITE (or from the first T-WRITE of the file if no T-WRITE was previously confirmed).

C.1.5.1.3 Parameters for T-WRITE-END

The T-WRITE-END service is a confirmed service, so the corresponding response code should be sent by the receiver before any other protocol exchange.

Parameter	T-WRITE-END Request	T-WRITE-END Indication	T-WRITE-END Response	T-WRITE-END Confirmation	S
First block	Mandatory	Mandatory (=)	Optional	Optional (=)	S
Block number	Optional	Optional (=)			
Data field	Optional	Optional (=)			
Result			Mandatory	Mandatory (=)	
(=) The value of the parameter is identical to the value of the corresponding parameter in the preceding request or response primitive.					

C.1.5.1.3.1 First block

This parameter is used to indicate whether or not the data contained in the data field parameter consist in the beginning of the file.

C.1.5.1.3.2 Block number

In the basic kernel this parameter is absent.

In the symmetrical service, the block number is mandatory if the recovery mechanism is selected or the anticipation window is not 1. The block number is in the range 0 to 65535. The first block is 0.

If a block number is present in a request, it must be sent back in the response.

C.1.5.1.3.3 Data field

It is used to convey data from the file, it is a variable length string of no more than 1024 bytes in the basic kernel, and the maximum length is negotiated in the symmetrical service during the establishment of the Access regime.

C.1.5.1.3.4 Result

This parameter indicates:

- in the basic kernel, file acceptance or refusal;
- in the symmetrical service, this parameter indicates file acceptance or refusal, or rejection of the service(s) since the last confirmed recovery point.

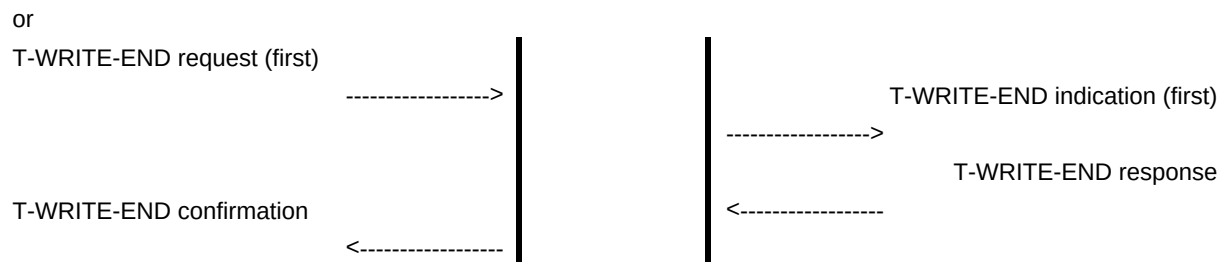
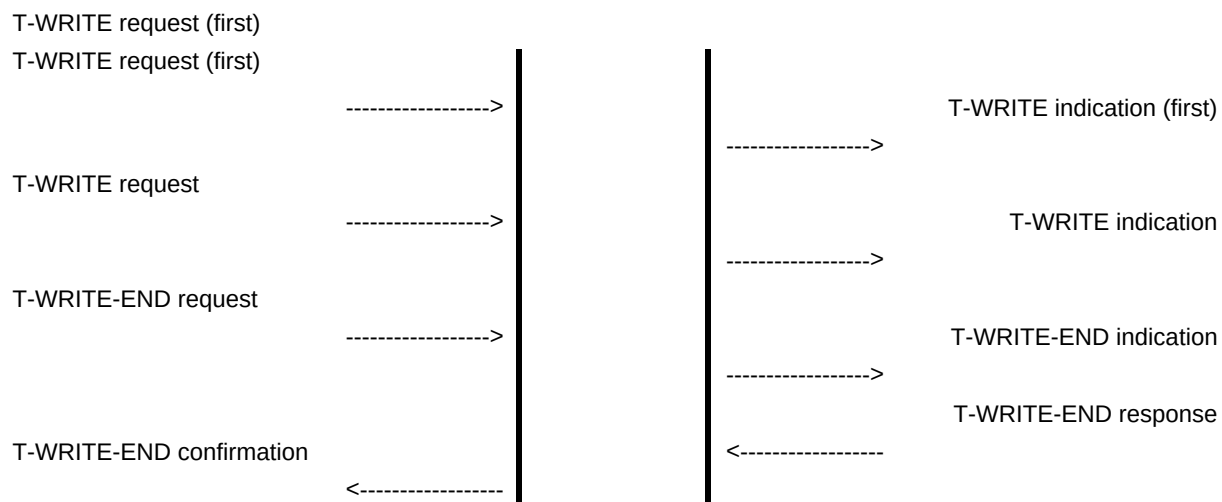
C.1.5.1.4 Operation of the mass transfer procedure

It is the purpose and the responsibility of the mass transfer phase to ensure the correct transfer of the identified mass data.

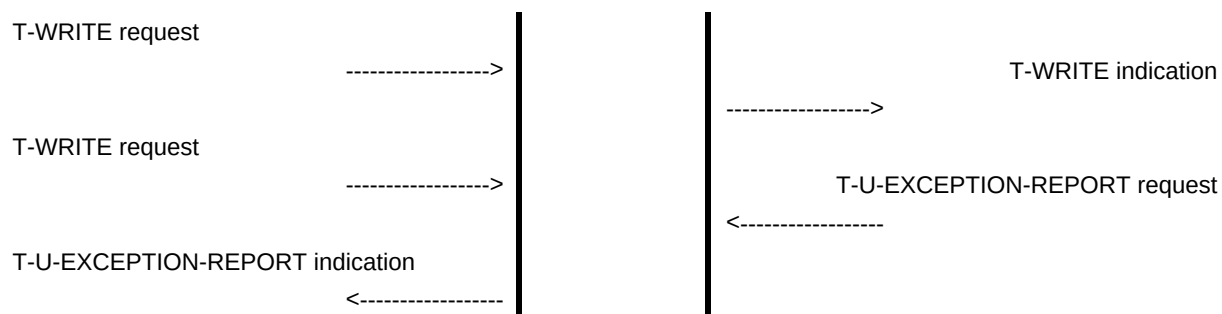
A mass transfer phase is initiated by T-WRITE request and indication primitives with the first block of data containing the file header or a part of it, and during a mass transfer phase, the reception of a valid T-WRITE indication should cause the data to be added or overwritten at the appropriate place in the defined memory space. The data fields accompanying T-WRITE or T-WRITE-END are assembled into a file according to the parameters specified in the file header, and a T-WRITE-END response and confirmation should only be transmitted when the receiver has successfully stored (as required) the transferred file. Positive confirmation of T-WRITE-END indicates that the Receiver accepts the responsibility of the file; negative confirmation indicates that the Receiver does not accept the responsibility of the file.

In the printer device application, the data fields are stored or passed to the printer device. The data must be transferred sequentially and contiguously. The T-WRITE-END response and confirmation should be issued only when the receiver has successfully received and forwarded the transferred data or has reliably stored it for later transfer.

In case the transfer is correctly completed, the sequence of events may be represented as follows:



In case of a user error, the sequence of events is as follows:



The resulting state of the two entities depends on the value of the parameter “Reason” in the T-U-EXCEPTION-REPORT as defined in C.1.5.2.2.1.

C.1.5.2 Exception report service

C.1.5.2.1 Function

This service is used in the Transfer regime:

- by the Receiver to force the transfer abort or to restart the mass transfer at the beginning of the file;
- by the Sender or the Receiver in the symmetrical service to force the transfer abort.

This service is performed by the T-U-EXCEPTION REPORT Request primitive.

C.1.5.2.2 Parameters

The different primitives and parameters required by the Exception report service are described in the following table.

Parameter	T-U-EXCEPTION-REPORT Request	T-U-EXCEPTION-REPORT Indication
Reason	Mandatory	Mandatory (=)
(=) The value of the parameter is identical to the value of the corresponding parameter in the preceding request primitive.		

C.1.5.2.2.1 Reason

This parameter may take the following values:

- Read restart* – If used during the mass transfer phase it will cause the mass transfer to be reset and restarted from the beginning.
- Transfer reject* – Used if the Receiver cannot support elements required by the file attributes transmitted in the first T-WRITE primitive, or in the case of any user error, when receiving a T-WRITE, to discontinue the transfer.

If this parameter is equal to “Transfer reject”, the Transfer regime is terminated.

If this parameter is equal to “Read Restart” the Transfer regime is not interrupted and the transfer phase is resumed from its beginning.

C.1.5.2.3 Error recovery operation



C.1.6 Exception

C.1.6.1 Exception reporting

C.1.6.1.1 Function

The T-P-EXCEPTION-REPORT Indication primitive is used by the service provider to signal exceptions during the service to the user on the other end. This primitive may only be used within the Access regime.

The exception is an error the service provider considers as recoverable.

After reception of T-P-EXCEPTION-REPORT the two entities are back in the idle state in the Access regime.

C.1.6.1.2 Parameters

The different primitives and parameters required by the exception reporting service are described in the following table.

Parameter	T-P-EXCEPTION-REPORT Indication
Reason	Mandatory

C.1.6.1.2.1 Reason

- Reason:
 - reason not specified;
 - repeated negative acknowledgements;
 - protocol conflict;
 - delay expired;
 - syntax error/missing parameter;
 - primitive not handled;
 - other reason.

C.1.6.1.3 Error recovery operation



C.1.7 Collisions

In a given regime a user can normally issue a request only in the idle state, meaning no other request or indication is in the process.

In certain cases, both the entities are allowed to initiate services. Collisions between two services requested can thus arise.

C.1.7.1 Collision in the Association phase

In case of collision in the Association phase, the initiator of the physical connection ignores the T-ASSOCIATE indication, and the acceptor of the physical connection must answer to the T-ASSOCIATE indication so that only the association requested by the initiator is considered.

C.1.7.2 Collision in the Association regime

The only possible collisions in the Association regime are the ones between a release request and an access or abort request.

The abort service has the highest priority, in this case the collision results always in the service abort. If an abort request collides with a service indication, the service indication is not taken into account and the abort request is carried out. If an abort indication collides with a service request, the service request is ignored and the abort indication is carried out.

The release service has priority on an access request; in case a collision occurs it will result in this case in an association termination.

If a release request collides with an access service indication, the service indication is not taken into account and the release request is carried out. If a release indication collides with an access service request, the access service request is ignored and the release indication is carried out.

If a release request collides with a release indication, the release indication of the Master is carried out and the other one is ignored.

C.1.7.3 Collision in the Access regime

In the Access regime a certain number of collisions between service requests can occur, these collisions are solved in the following way:

The following table specifies the result of collisions in the Access regime.

Master	ABORT	END-ACCESS	DLSRD	TYPED-DATA	P-EXCEPTION-REPORT
Slave					
ABORT	Abort	Abort	Abort	Abort	Abort
END-ACCESS	Abort	End-Access (Note 1)	End-Access	End-Access	End-Access
TYPED-DATA	Abort	End-Access	(Note 2)	(Note 3)	(Note 3)
P-EXC-REPORT	Abort	End-Access	(Note 2)	(Note 3)	(Note 4)
ABORT	T-U-ABORT, T-P-ABORT				
END-ACCESS	T-END-ACCESS				
DLSRD	T-DIRECTORY, T-LOAD, T-SAVE, T-RENAME, T-DELETE				
TYPED-DATA	T-TYPED-DATA				
P-EXCEPTION-REPORT	T-P-EXCEPTION-REPORT				
Collision result					
Abort:	The abort request is fulfilled. The association is terminated, data may be lost.				
End-Access:	The end-access request is fulfilled. The Access regime is terminated. The other service is ignored or given up.				
NOTES					
1 The Masters end-access request is fulfilled. The Slavs request is ignored or given up.					
2 The Slaves request is fulfilled. The Masters request is confirmed normally by the Slave. The T-TYPED-DATA primitive has no consequence on the service.					
3 The two requests are fulfilled.					
4 No collision, back to the idle state in the Access regime.					

C.1.7.4 Collision in the Transfer regime

During the transfer phase, the priority order is the following: first the abort service, then, the user's exception report service and the provider's exception report service.

Whenever the service provider issues a T-P-ABORT Indication or a T-P-EXCEPTION REPORT Indication he gives up the current service and goes back to the idle state, as before the Association regime establishment or in the idle state in the Access regime.

Whenever the service provider receives a T-U-ABORT Request or a T-U-ABORT Indication, he gives up the current service and fulfils the user's abort service, and goes back to the idle state, as before the association regime establishment.

Whenever a transfer request and a abort transfer request collide, the abort transfer request is fulfilled, and all the transfer primitives in the process are given up.

C.2 Telesoftware and auxiliary device applications

C.2.1 Preliminaries

Two applications are hereby defined: telesoftware and auxiliary device.

Through a network, a telesoftware application allows interconnection of two machines and thus allows file exchange.

The telesoftware and auxiliary device application are defined by rules specifying the use of the service described in Section 1 and by a file structure which contains the characteristics of each virtual file and the relationship of these characteristics to the actions performed by the applications which manipulate these files.

A transfer file structured is composed of a set of information containing:

- a unique file name, that allows it to be referenced without ambiguity;
- descriptive file attributes which express characteristics of the file (date, etc.);
- attributes describing the logical structure and the data stored in the file;
- data, forming the content of the file.

The three first kinds of information constitute the header of the file, and the data constitute the content of the file.

To be able to use the file transfer service, a practical implementation must state the relationship between the elements in the transferred file and the real storage system available.

In this application the files have been divided up in three different groups.

In the symmetrical service, the application does not make any assumption on the two machines and their functioning. This application may be used between two Servers, or two Executors, or between a Server and an Executor. Either entity may take the part of the Master depending on its requirements by redefining an Access regime within a given Association regime.

C.2.2 The telesoftware application organisation

The Telesoftware application uses files described in C.2.5 and these files are transferred according to the rules described in C.2.6.

The Telesoftware organisation allows file exchange from the three different groups:

- group of transferable files (group A);
- group of application presentation files (group B);
- group of service support files (group C).

NOTE – When using the service class “basic kernel” only group A files are considered. In the symmetrical service class, the default group is group A and other groups may not be implemented in some Data Bases.

C.2.2.1 Transferable files – Group A

This group contains transferable files consisting of description, text, software, data and command files which together make up the application (see C.2.4).

These files can be obtained by their transfer name. The list of all transfer names make up the file directory. The sublist of all transfer names of structure files make up the application’s directory.

C.2.2.2 Application presentation file – Group B

This group contains files describing the applications. If these files exist, they share the same transfer name as the corresponding description file (name of the application). The list of all the transfer names of application presentation files make up the application presentation file’s directory.

These files are text files. They carry technical, user and managerial information on the application sharing the same name. The aim of these files is to provide the maximum of information on the application in order to better guide the user.

C.2.2.3 Service support – Group C

This group contains files for general information. This information is organised by use of files and is accessible using key words. The list of all the transfer names of the service support files make up the service support file directory.

These files are text files. They carry general information on the service.

C.2.2.4 Working area

Several working areas are possible. It is up to the higher level application to define the different working areas. These applications should be able to offer access to a public and a private working area. The access rights to these working areas may depend on different criteria (example: password).

However, access to one or the other working areas should be left transparent for the operator.

C.2.3 Printer application organisation

The printer device application is a specific case of the auxiliary device application. It uses files as described in C.2.5. These files are transferred according to the rules described in C.2.6. Not all the parameters of the file are used for the printer device application (see C.2.5.1.17).

C.2.4 Files

C.2.4.1 File identification

C.2.4.1.1 Preliminaries

In telesoftware, there are five types of files:

- description files (DeF);
- software files (SF);
- data files (DF);
- command files (CF);
- text files (TF).

The following subclauses indicate how these different files are regrouped in the three groups previously defined.

C.2.4.1.2 Description files

These files belong to the group A. They carry data required to transfer the application, and also the names of all the files which are part of this application.

C.2.4.1.3 Software file

These files belong to the group A. They carry software to be executed immediately or after processing.

C.2.4.1.4 Data files

These files belong to the group A. They carry information required by software files at execution time.

C.2.4.1.5 Command files

These files belong to the group A. They carry processing indications meaningful for the entity who required transfer of these files. These indications are to be applied on files part of the application. The processing is done before or during execution of these files.

C.2.4.1.6 Text files

These files may belong to the group A, B or C. These files may be created upon a directory request on group A, B or C.

C.2.4.1.6.1 Text files – Group A

These files can be displayed and represent a user's manual or some data displayed during the execution of the application.

C.2.4.1.6.2 Text files – Group B

These files carry displayable data. This data can be useful, before requiring transfer of the corresponding application.

C.2.4.1.6.3 Text files – Group C

These files carry displayable data of the telesoftware support service.

This information is not necessarily related to the transferable applications.

C.2.4.1.6.4 Text files from a file directory request

These files list all the file names belonging to the set of files where the request took place. The length of this list depends on the parameter designation in the file directory request.

C.2.4.2 Transferable applications

A transferable application is a set of files. These files are part of the group A and their number is not limited.

C.2.4.2.1 Structure of a transferable application

A transferable application is made up of a description file and at least one software, command, data or text file.

C.2.4.2.1.1 Purpose of the description file

The purpose of the description file is to provide coded information regrouping the characteristics of a transferable application and the list of all files which are part of this application.

C.2.4.2.1.2 Organisation of the description file

Like all other files, the description file is made up of a header and of a main body.

The header is described in C.2.5.1.

The main body regroupes all the headers of all the files which are part of this application.

The data in these headers are regrouped in each file.

This data provides useful elements needed to perform correctly the transfer of an application, for instance: type, transfer name, size, resources and coding. Other data fields can be found.

C.2.4.3 File classification

The files are classified into three groups.

Inside each group, the files have a unique transfer name. This name is kept in a directory. It points to an application whenever it is the name of a description file, and to a file in all other cases.

C.2.4.3.1 Structure of a transfer name

A transfer name is made up of one or more keywords. The maximum number of keywords in a name is eight (8).

These keywords are called subnames of the transfer name.

C.2.4.3.1.1 Keywords

A keyword is a byte sequence of no more than 12 bytes. Each byte can take different values.

C.2.4.3.1.2 Transfer name

If a transfer name is made up of several keywords, then they are separated by a byte "/". The maximum length of a transfer name, counting all separators, is 70.

C.2.5 Description of the transfer file structure

The transfer file is structured as follows:

- the file's attributes: the header;
- the data in the file: the file's content.

This organisation is valid for an application:

- a) telesoftware;
- b) printing;
- c) telesoftware and printing.

C.2.5.1 Header

The header provides information on the characteristics of the file. It is always transmitted in the first T-WRITE (or T-WRITE-END for short files) and possibly in the following ones if necessary, in order to link the header information to the data. The file content may be transmitted in the first T-WRITE (if its data field is not filled up with the file header), and subsequently in other T-WRITE or T-WRITE-END.

The following attributes may be used:

C.2.5.1.1 File type

There are five different types of file:

- the description file which identifies the characteristics of the files which compound the application;
- the software files to be executed;
- the data files;
- the command files;
- the text files to be displayed or printed.

The default value is text file.

C.2.5.1.2 Execution order

This attribute indicates whether the application should be executed immediately or after the association to the telesoftware application has been released. It also indicates the starting file of the application (default value: don't care).

C.2.5.1.3 Transfer name

This attribute represents the designation of the file on the sender side (C.2.4.3.1).

C.2.5.1.4 Filename

This attribute provides a "name" which may be associated with the transferred file in the filestore, (the filename is not necessarily related to any transfer name).

NOTE – Some systems may require that the first 6 characters of the filename should be alphabetic or numeric. Drive, device or directory designations should not be included in this attribute, but file-type suffices may be included.

C.2.5.1.5 Date/time of last modification

This attribute indicates the date and time of the last modification of the file.

C.2.5.1.6 File length

This attribute indicates the size in bytes of the file content in uncompressed form.

C.2.5.1.7 Destination code

This attribute may take the values "don't care" (default), "foreground memory", "background memory random access required" (e.g. disk) or "cassette tape required".

NOTE – The latter option may be required for some software to overcome operating system or security constraints.

C.2.5.1.8 File coding

This attribute indicates the data syntax, the language, the machine type or the operating system.

According to the file type, this attribute can take the following values:

- a) *Software file:*
 - source code in a text form;
 - source code in a tokenized form;
 - intermediate code;
 - object code;
 - executable code (default value).
- b) *Data file:*
 - binary code (default value);
 - character code.
- c) *Command file:*
 - dependant machine code (default value);
 - standardized code.
- d) *Text file:*
 - character code (default value);
 - Videotex code (+ profile);
 - other code.

Additional information, if present, identify by means of a text string a language (such as “BASIC”, “P-CODE”) or a target processor (such as “6502”). This may, optionally, be followed by an identification of the language dialect (such as “MSDOS”).

C.2.5.1.9 Destination name

This attribute indicates a drive, device and/or directory name for the storage of the file. It may not be used in combination with a destination code. The length of the attribute is limited to 255 bytes.

C.2.5.1.10 Cost

This attribute represents the price of the software, it is a string of no more than 31 bytes. The content of this field is only for information.

C.2.5.1.11 User field/Application reference

This field is used to carry comments related to the file or to the relevant application and consisting of no more than 254 displayable characters.

C.2.5.1.12 Load address

This attribute may be used to indicate an address in foreground memory at which the data should be loaded. In the case of a transfer to foreground memory, this attribute provides a base address for the transferred data, in the case of transfer to background memory, this attribute may be recorded as a file attribute according to the terminal filing's system. The relationship of this address to any executor operating system virtual memory addressing scheme is not specified by this protocol.

C.2.5.1.13 Execute address (absolute)

This attribute may be used to indicate an absolute address at which programme execution should start when the file is loaded into foreground memory. The relationship of this address to any executor operating system virtual memory addressing scheme is not specified by this protocol.

C.2.5.1.14 Execute address (relative)

This attribute may be used to indicate an address relative to the beginning of the file after loading into foreground memory, at which programme execution should start.

C.2.5.1.15 Compression mode

This attribute is used to indicate which compression algorithm is used for the file's content.

This attribute can take the following values:

- Basic compression mode;
- High efficiency compression mode;
- “Application defined” compression mode.

NOTE – There is a proposal for a compression algorithm given in an informative Annex A in order to provide interoperability.

C.2.5.1.16 Device

This attribute is used to specify the characteristics of the device to which a file content is to be sent.

Note that in the basic kernel only the printer is taken into account.

C.2.5.1.17 File checksum

This attribute contains the 32-bit “Frame Check Sequence” (FCS) calculated from the uncompressed file content.

C.2.5.1.18 Author name

This attribute indicates the name of the creator of the file.

C.2.5.1.19 Future file length

This attribute indicates the nominal length to which the file may grow as a result of an extension.

C.2.5.1.20 Permitted actions

This attribute indicates the set of actions that can be performed on the file (read, insert, replace, erase).

C.2.5.1.21 Legal qualification

This attribute conveys information about the status and use of the file.

C.2.5.1.22 Creation

This attribute indicates the date and time of the creation of the file.

C.2.5.1.23 Last read access

This attribute indicates the date and time of the last reading of the file.

C.2.5.1.24 Identity of the last modifier

This attribute indicates the name of the last modifier of the file. It is altered by the receiver whenever the file has been opened for modification or extension and is closed.

C.2.5.1.25 Identity of the last reader

This attribute indicates the name of the last reader of the file. It is altered by the receiver whenever the file has been opened for reading and is closed.

C.2.5.1.26 Recipient

This attribute indicates the name of the receiver of the file.

C.2.5.1.27 Telematic file transfer version

This attribute indicates the telematic file transfer method version being used (ETS number and date).

C.2.5.1.28 Status of file attributes

C.2.5.2 File content

The file's content carries executable data or data needed for presentation, downloading or execution of other files.

C.2.6 Use of the T-SERVICE for telesoftware and printer device application

C.2.6.1 Preliminaries

The telesoftware and auxiliary device applications use all of the symmetrical class service primitives in the service.

Only one telesoftware application may be associated at a time, and only one file may be transmitted at a time.

All the attributes listed in C.2.5.1.17 may be used to characterise a file involved in a telesoftware application.

Only one printer device application may be associated at a time, and only one mass transfer may be processed at a time.

The attributes listed in C.2.5.1.17 may be used to characterise a file involved in a printer device application.

For telesoftware applications consisting in several files, a description file may be transmitted first. This file consists of its own header and contains all the headers of the files to be downloaded. Each header is self-delimited by TLV encoding. When transmitting the subsequent files, the transmitted headers may not include information which were included in the description file.

This organisation allows for using the TDU service for other file transfer applications than telesoftware which would require a different file organisation (e.g. a specific application descriptor which would be different from the description file).

C.2.6.2 Association

The purpose of the association is to establish a link between two applications.

The association uses the T-ASSOCIATE Request and the T-ASSOCIATE Response services primitives.

File attributes	Application		Default value
	Telesoftware	Printer	
File type	O	O	Text file
Execution order	O	–	Don't care
Transfer name	O	O	No
Filename	O	O	No
Date of last modification	O	O	No
File length	O	O	No
Destination code	O	–	Don't care
File coding	O	O	Depends on file type
Destination name	O	–	No
Cost	O	O	No
User field/Application reference	O	O	No
Load address (abs.)	O	–	No
Execute address (abs.)	O	–	No
Execute address (rel.)	O	–	No
Compression mode	O	O	No compression
Device	O	O	Don't care
File checksum	O	O	No
Author name	O	O	No
Future file length	O	O	No
Permitted actions	O	O	No
Legal qualification	O	O	No
Creation	O	O	No
Last read access	O	O	No
Identity of the last modifier	O	O	No
Identity of the last reader	O	O	No
Recipient	O	O	No
Telematic file transfer version	O	O	No
O Optional – Irrelevant			

The application name will carry:

- !T telesoftware;
- !P printer's application;
- !A telesoftware and printer's application.

If !T is used as an application name, only telesoftware files may be transferred during the association.

If !P is used as an application name, only printer device files may be transferred during the association.

If !A is used, printer device and telesoftware files may be exchanged during the association.

If the selected service class is “basic kernel”, the restriction on the T-service apply and the access, end access, file directory, load, save help, rename, delete and typed data services are not available.

C.2.6.3 Release

The release of the application is performed by the T-RELEASE Request and T-RELEASE Response service primitives.

C.2.6.4 Abort

The abort service is used to leave an application in an abnormal way.

The T-U-ABORT Request primitive may be used by the application.

The parameter Reason in the Request/Indication indicates the reason for the abort.

C.2.6.5 Access

The transfer conditions are established by use of the T-ACCESS Request and T-ACCESS Response service primitives.

If the role is Slave, the parameter User data indicates the scope of services like T-DIRECTORY, T-LOAD, T-SAVE, T-RENAME, T-DELETE on the groups A, B and C.

If the role is Master, the parameter User data is not used.

C.2.6.6 End of access

Terminating an Access regime is performed by the T-END-ACCESS Request and T-END-ACCESS Response service primitives.

C.2.6.7 File directory

The T-DIRECTORY Request primitive is used by the Master to request transfer of files, collection of file names and application name, from the Slave to the Master. In case the request is accepted, the file content will depend on the designation parameter in the request.

A directory request is performed by the T-DIRECTORY Request primitive.

The parameter Designation is a byte sequence indicating the criteria for selecting the file names.

The parameter User data indicates the group or subgroup on which applies the directory request.

The response/confirmation is provided by the parameter Result in the T-DIRECTORY Response primitive, this parameter indicates acceptance or the reason for the refusal.

Acceptance means that file or application names meet with the criteria in the request.

The Slave will then carry on with the transfer of the file listing all the file or application names fulfilling the request.

C.2.6.7.1 Byte sequence in a directory request

The designation parameter in a directory request is made of one or more elementary words separated by a specific operator. Other operators as OR, AND, or specific codes as parenthesis or star can be used. A detailed description of the syntax is given in C.2.7.

The maximum number of elementary words is 8.

C.2.6.8 Load

The load request is performed by the T-LOAD Request service primitive. This primitive is used by the Master to request file transfer from the Slave to the Master.

The designation parameter contains the file transfer name requested (see C.2.4.3.1).

The recovery parameter if it is present (provided it was previously allowed) indicates the number of the last T-WRITE correctly received. Recovery will start with the following number.

The user data parameter indicates the group required in the LOAD service. If this parameter is absent the group is meant to be the A group.

The response/confirmation is provided by the parameter Result in the T-LOAD Response primitive. This parameter indicates acceptance or the reason for the refusal.

Acceptance means that a file meets the request.

The Slave will then carry on with the transfer of the file.

C.2.6.9 Help

Request of a help file can be performed by the T-LOAD Request service primitive on the group A, the value of the designation parameter is then 2/0.

The recovery parameter if it is present (provided it was previously allowed) indicates the number of the last T-WRITE correctly received. Recovery will start with the following number.

The response/confirmation is provided by the parameter Result in the T-LOAD Response primitive. This parameter indicates acceptance or the reason for the refusal.

Acceptance means that a file meets the request.

The Slave will then carry on with the transfer of the file.

The purpose of this file is to provide information on the organisation of the Slave's data base: structure, key words, file name, etc.

This file is a text file coded characters.

C.2.6.10 Save

In the basic kernel, the Master may request transfer from Master to Slave by initiating a mass transfer using T-WRITE.

In the symmetrical service, the T-SAVE Request primitive is used by the Master to request file transfer from the Master to the Slave.

In the telesoftware service, one can transfer files from any of the A, B or C groups (see C.2.2).

The designation parameter contains the file transfer name requested (see C.2.4.3.1).

The recovery parameter if it is present (provided it was previously allowed) indicates the number of the last T-WRITE correctly received. Recovery will start with the following number.

The user data parameter indicates the group required in the SAVE service. If this parameter is absent the group is meant to be the A group.

The response/confirmation is provided by the parameter Result in the T-SAVE Response primitive. This parameter indicates acceptance or the reason for the refusal.

Acceptance means that a file meets the request.

The Master will then carry on with the transfer of the file.

C.2.6.11 Rename

The T-RENAME Request primitive is used by the Master to request renaming of a file or application transfer name in the Slave's data base.

The designation and new name parameters contain the old and new file (or application) transfer name.

The user data parameter indicates the group required in the RENAME service. If this parameter is absent, the group is meant to be the A group.

The response/confirmation is provided by the parameter Result in the T-RENAME Response primitive. This parameter indicates acceptance or the reason for the refusal.

Acceptance means that a file was renamed.

Refusal indicates that the renaming was not performed.

C.2.6.12 Suppression

The T-DELETE Request primitive is used by the Master to request the suppression of a file or application transfer name in the Slave's data base.

The designation parameter indicates the file to be deleted.

The user data parameter indicates the group required in the DELETE service. If this parameter is absent the group is meant to be the A group.

The response/confirmation is provided by the parameter Result in the T-DELETE Response primitive. This parameter indicates acceptance or the reason for the refusal.

Acceptance means that a file was deleted.

Refusal indicates that the suppression was not performed.

C.2.6.13 Transfer abort

The T-U-EXCEPTION REPORT Request primitive is used by either entity to suspend or restart a file transfer.

The parameter indicates the reason for this.

C.2.7 The designation field in a directory request

The designation field in a directory request enables one to define a research criteria in the current data base.

A joker "*" can be used in this field to designate a character string. Also logical operators AND, displayed /, and OR, displayed +, can be used to combine various character strings. Parenthesis (and) indicate how these operators act.

Examples:

- 1) Any file in the data base meets criteria * or criteria * + *, */*, (*+*).
- 2) Files SOFT and SMALL.BAT meet criteria S*T.
- 3) Files SOFT, SMALL.BAT, SUN meet criteria S*/(*N + *T).

The aim of this annex is to define the syntax of the designation field in a directory request and the associated criteria research.

The sequences (AB), (A) + B*, (((*))), */* are all directory request syntactically correct, this is not the case of the following sequences: A++B, (((A*)B*)C*), A(/)B.

Definitions:

- a) A **file name** is any sequence of no more than 8 key words separated by /, a **key word** is any sequence of no more than 12 bytes (any value between 2/1 and 7/14 is permitted with the exception of 2/8, 2/9, 2/10, 2/11, 2/15 displayed (,), *, + and /). Each one of these keywords is called **subname** of the file name.
- b) An **elementary word** is any sequence of no more than 12 bytes (any value between 2/1 and 7/14 is permitted with the exception of (,), + and /). Moreover no more than one byte is equal to *.

Using the above definitions, a correct syntax for a directory request can be specified by also using the following 4 grammatical rules (\$ designates here an abstract symbol but not a character):

(0) \$ -----> choose your elementary word

(1) \$ -----> (\$)

(2) \$ -----> \$ + \$

(3) \$ -----> \$/\$

One reads -----> changes into

Any sequence generated using those four rules is said to be a correct sequence for a directory request.

Example: $S^{*}/(*N + *T)$

For this example the first rules (1), (2) and (3) are used:

$\begin{array}{ccccc} \$ & \text{-----}> & \$/\$ & \text{-----}> & \$/(\$) & \text{-----}> & \$/(\$ + \$) \\ (3) & & (1) & & (2) \end{array}$

and then rule (0) is used to replace each symbol \$ by an elementary word:

$\$/(\$ + \$) \text{-----}> S^{*}/(\$ + \$) \text{-----}> S^{*}/(*N + *T)$

and finally

$\text{-----}> S^{*}/(*N + *T)$

To any byte sequence forming a correct directory request is associated a research criteria:

A file name is said to meet the criteria “a” if and only:

- a) If “a” is an elementary word and if substitution of the * (if any) by 0, 1 or more characters gives a subname of this file name.
 - Examples: SOFT, RON/SOFT, SUN/RON/SOFT meet criteria SOFT.
 - SOFT, SUN, SAND/SUN/ALL meet criteria S^{*} .
 - SUN, SAND/SUN/ALL meet criteria $S^{*}N$.
- b) Or if “a” can be written (b) and the file name meets criteria b.
 - Examples: any file meets criteria (*).
 - SOFT, SAND/SUN/ALL meet criteria (S^{*}).
- c) Or if “a” can be written b + c and the file name meets the criteria b OR criteria c.
 - Examples: SOFT and SAND/SUN/ALL meet criteria $*N + *T$.
 - SOFT and RON meet criteria ($S^{*}T$) + ($*N$).
- d) Or if “a” can be written b/c and the file name meets criteria b AND criteria c.
 - Examples: SAND/SUN/ALL meets criteria $S^{*}D/A^{*}$.
 - SOFT meets criteria $S^{*}/*T$.
 - RON/SOFT meets criteria $*N/(SAND + R^{*})$.

Important: If “a” is not an elementary word, decomposing “a” in simpler criteria is made in this order. It is equivalent to say the operator “/” has higher priority than operator “+”.

Example:

$AB/T + CA$ is $(AB/T) + CA$.

C.3 T-Protocol specification

C.3.1 Definitions

block: A block of user information sent in a T-Write TDU.

mass transfer phase: The mass transfer phase is started by sending (or receiving) the first T-Write TDU and terminated when receiving (or sending) a T-Abort , a T-Transfer-reject, a T-Read-restart or the confirmation to the last T-Write TDU.

telesoftware data unit (TDU): These units are protocol elements which are used to handle the T-Protocol.

idle state in the association regime: The state which is reached when the association regime is established, when no other regime is established and when no other service is being initiated.

idle state in the access regime: The state which is reached when the access regime is established, when no Transfer regime is established and when no other service is being initiated.

Table C-2 gives an overview of TDU's and their mapping on the service primitives, and indicates which TDU must be sent for a given service primitive and which TDUs may be received in response to that TDU. However, this table is not exhaustive and does not deal with collisions nor with the complete error handling mechanism.

Table C.2/T101 – List of the TDU's used

Issued service primitive	TDU	Possible response TDU	Received service primitive
T-ASSOCIATE request	T-Associate	T-Response-positive (S) T-Response-negative (S) T-Abort (A)	T-ASSOCIATE conf(+) T-ASSOCIATE conf(-) T-U/P-ABORT ind
T-RELEASE request	T-Release	T-Response-positive (S) T-Abort (A)	T-RELEASE conf(+) T-U/P-ABORT ind
T-U-ABORT request	T-Abort	No response is required	
T-ACCESS request T-END-ACCESS request	T-Access T-End-Access	T-Response-positive (S) T-Response-negative (S) T-Abort (A) T-Response-positive (S) T-Abort (A)	T-ACCESS conf(+) T-ACCESS conf(-) T-U/P-ABORT ind T-END-ACCESS conf(+) T-U/P-ABORT ind
T-DIRECTORY request	T-Directory	T-Response-positive (S) T-Response-negative (S) T-P-Exception (A) T-Abort (A)	T-DIRECTORY conf(+) T-DIRECTORY conf(-) T-P-EXCEPTION ind T-U/P-ABORT ind
T-LOAD request	T-Load	T-Response-positive (S) T-Response-negative (S) T-P-Exception (A) T-Abort (A)	T-LOAD conf(+) T-LOAD conf(-) T-P-EXCEPTION ind T-U/P-ABORT ind
T-SAVE request	T-Save	T-Response-positive (S) T-Response-negative (S) T-P-Exception (A) T-Abort (A)	T-SAVE conf(+) T-SAVE conf(-) T-P-EXCEPTION ind T-U/P-ABORT ind
T-RENAME request	T-Rename	T-Response-positive (S) T-Response-negative (S) T-P-Exception (A) T-Abort (A)	T-RENAME conf(+) T-RENAME conf(-) T-P-EXCEPTION ind T-U/P-ABORT ind
T-DELETE request	T-Delete	T-Response-positive (S) T-Response-negative (S) T-P-Exception (A) T-Abort (A)	T-DELETE conf(+) T-DELETE conf(-) T-P-EXCEPTION ind T-U/P-ABORT ind

Table C.2/T101 – List of the TDU's used (end)

Issued service primitive	TDU	Possible response TDU	Received service primitive
T-WRITE request	T-Write	T-Response-positive (S) T-Response-negative (S) T-Transfer-reject (A1) T-Read-restart (A1) T-P-Exception (A) T-Abort (A)	T-WRITE conf(+) T-WRITE conf(-) T-U-EXCEPTION ind T-U-EXCEPTION ind T-P-EXCEPTION ind T-U/P-ABORT ind
T-WRITE-END request	T-Write (end)	T-Response-positive (S) T-Response-negative (S) T-Transfer-reject. (A1) T-Read-restart (A1) T-P-Exception (A) T-Abort (A)	T-WRITE-END conf(+) T-WRITE-END conf(-) T-U-EXCEPTION ind T-U-EXCEPTION ind T-P-EXCEPTION ind T-U/P-ABORT ind
T-TYPED-DATA request	T-Typed-data	No response is required T-Abort (A)	T-U/P-ABORT ind
T-U-EXCEP request (Read-rest)	T-Rd-restart	T-Write (S) T-Abort (A)	T-WRITE ind T-U/P-ABORT ind
T-U-EXCEP request (Any other reason)	T-Tr-reject	No response is required T-P-Exception (A) T-Abort (A)	T-P-EXCEPTION ind T-U/P-ABORT ind
(A) Means that the TDU may be sent in an asynchronous manner (at any time after having received the TDU). (A1) Means that the TDU may be sent in an asynchronous manner (at any time after having received the TDU during the mass transfer phase). (S) Means that the TDU must be sent synchronously (in response to the other TDU). (+) The result parameter of the primitive is set to “accept”. (-) The result parameter of the primitive is set to “reject”.			

C.3.2 Description and use of TDU

C.3.2.1 T-Associate

T-Associate is used to establish the association with a specified processable data application.

The response to a T-Associate is conveyed using the TDU T-Response-positive or T-Response negative.

C.3.2.1.1 Content of the T-Associate TDU and the associated responses

T-Associate

Calling address *

Called address *

Application name

Service class

Explicit confirmation

Timeouts

Request identification *

Identification *

User data

T-Response-positive

Called address *

Time-outs *

Identification *

User data *

T-Response-negative

Result *

* This parameter may only be present if the symmetrical service class is proposed.

The use of the parameters is described in the service definition.

The format of the T-Response-positive indicates which service class is selected.

The Result parameter in T-Response-negative may indicate the following reasons:

- Reason not specified (default value)
- Rejection by the service provider:
 - Called address incorrect;
 - Calling address incorrect;
 - Service class refused.
- Rejection by the service user:
 - Called address incorrect;
 - Calling address incorrect;
 - Service class refused;
 - Application name unknown;
 - Wrong identification;
 - Erroneous user data;
 - Other reason (this last value may be followed by a string of no more than 62 displayable characters).

C.3.2.1.2 Sending T-Associate

A T-ASSOCIATE request primitive results in sending a T-Associate TDU.

If no explicit confirmation have been requested, the sender is in the state of association established.

If explicit confirmation had been requested, the sender waits for a T-Response-positive or negative. In this case no other action is permitted before reception of this response except aborting the association. The reception of this response results in this case in a T-ASSOCIATE confirmation with the appropriate result code. The association will be established as soon as a T-Response-positive is received. If a T-Response-negative or a T-Abort is received, the association is not established and a D-U-Abort must be sent by the initiator.

NOTES

- 1 It is recommended that T-Abort is not used to reject a T-Associate for which an explicit confirmation was requested.
- 2 If the symmetrical service class is proposed in T-Associate an explicit confirmation must be requested.

C.3.2.1.3 Receiving T-Associate

A valid incoming T-Associate TDU results in a T-ASSOCIATE indication primitive.

If no explicit confirmation is requested, the association is established (it may be rejected using T-Abort).

If explicit confirmation is requested, a T-ASSOCIATE response primitive results in sending a T-Response-positive or negative depending on the result parameter of the primitive. If a T-Response-positive is returned, the association is established, otherwise the association is not established.

C.3.2.2 T-Release

This TDU is used to request the orderly termination of the processable data application.

The response to a T-Release is conveyed using the TDU T-Response-positive.

C.3.2.2.1 Content of the T-Release TDU and the associated response

T-Release

User data

T-Response-positive

User data *

* This parameter may only be present if the symmetrical service class has been selected.

The use of the parameters is described in the service definition.

C.3.2.2.2 Sending T-Release

A T-RELEASE request primitive results in a T-Release TDU. If the selected service class is “basic kernel”, only the initiator of the association is permitted to send a T-Release TDU. A T-Release TDU may be sent at any time after the association has been established when no other regime is established.

No other action is permitted after having sent a T-Release TDU before having received a T-Response-positive TDU except aborting the association.

On reception of the T-Response-positive TDU a D-U-Abort must be sent. The reception of a T-Response-positive results in a T-RELEASE confirmation.

C.3.2.2.3 Receiving T-Release

A valid incoming T-Release TDU results in a T-RELEASE indication. A T-RELEASE confirmation primitive results in sending a T-Response-positive to acknowledge the T-Release, a negative response is not permitted.

C.3.2.3 T-Abort

This TDU is used to abruptly terminate the association.

C.3.2.3.1 Content of the T-Abort TDU

T-Abort

Reason *

* This parameter may only be present if the symmetrical service class has been selected.

The following values of the Reason parameter are related to the T-P-ABORT service:

- Repeated negative acknowledgements / repeated errors;
- Delay expired;
- Unknown message;
- Protocol conflict;
- Unrecoverable lower layer error;
- Syntax error;
- Other reason (This last value may be followed by a string of no more than 62 displayable characters).

The following values of the Reason parameter are related to the T-U-ABORT service:

- Reason not specified (default value);
- Wrong identification;
- Erroneous role/function;
- Other reason (This last value may be followed by a string of no more than 62 displayable characters).

C.3.2.3.2 Sending T-Abort

A T-U-ABORT request primitive or an exception condition (see C.3.3.2, C.3.3.3) results in sending a T-Abort TDU.

A T-Abort may be sent at any time after having received a T-Associate, to reject a non-confirmed T-Associate or to abruptly terminate an association.

If the selected service class is basic kernel, only the acceptor of the association is permitted to send a T-Abort.

C.3.2.3.3 Receiving T-Abort

A valid incoming T-Abort TDU results in issuing a T-U-ABORT or T-P-ABORT indication primitive, depending on the reason parameter. The association is terminated.

A D-U-Abort DDU must be sent on reception of a T-Abort TDU.

C.3.2.4 T-Access

T-Access is used to establish the Access regime. This TDU may only be used if the symmetrical service class has been selected.

The response to a T-Access is conveyed using the TDU T-Response-positive or T-Response negative.

C.3.2.4.1 Content of the T-Access TDU and the associated responses

T-Access

Role
Functions
Transfer unit size
Anticipation window
Recovery
Transfer Mode
User data

T-Response-positive

Functions
Transfer unit size
Anticipation window
Recovery
Transfer Mode
User data

T-Response-negative

Result

The use of the parameters is described in the service definition.

The Result parameter in T-Response-negative may indicate the following reasons:

- Reason not specified (default value)
- Rejection by the service provider:
 - Role refused.
- Rejection by the service user:
 - Role refused;
 - Insufficient primitives handled;
 - Erroneous user data;
 - Other reason (this last value may be followed by a string of no more than 62 displayable characters).

C.3.2.4.2 Sending T-Access

A T-ACCESS request primitive results in sending a T-Access TDU. Only the Master is permitted to send a T-Access TDU when the Association regime is established and provided that no other regime is established. Then the sender waits for a T-Response positive or negative; no other action is permitted before reception of this response except aborting the association. The reception of this response results in a T-ACCESS confirmation with the appropriate result code. The Access regime and the assigned role will be established as soon as a T-Response positive is received. If a T-Response-negative or a T-Abort is received, the Access regime is not established.

C.3.2.4.3 Receiving T-Access

A valid incoming T-Access TDU results in a T-ACCESS indication primitive.

A T-ACCESS response primitive results in sending a T-Response-positive or negative depending on the result parameter of the primitive. If a T-Response-positive is returned, the Access regime is established, otherwise the Access regime is not established.

C.3.2.5 T-End-Access

T-End-Access is used to request the termination of the Access regime.

The response to a T-End-Access is conveyed using the TDU T-Response-positive.

C.3.2.5.1 Content of the T-End-Access TDU and the associated response

T-End-Access

Reason
User data

T-Response-positive

User data

The use of the parameters is described in the service definition.

The Reason parameter in T-End-Access may take the following values:

- Termination of the access regime requested by the service user:
 - Reason not specified (default value);
 - User abort of the access regime;
 - Insufficient primitives handled;
 - Other reason (this last value may be followed by a string of no more than 62 displayable characters).

C.3.2.5.2 Sending T-End-Access

A T-END-ACCESS request primitive results in sending a T-End-Access TDU. A T-End-Access TDU may be sent at any time after the Access regime has been established when no other embedded regime is established.

No other action is permitted after having sent a T-End-Access TDU before having received a T-Response-positive TDU except aborting the association. Upon reception of a T-Response-positive, both entities are in an idle state with the association regime established.

C.3.2.5.3 Receiving T-End-Access

A valid incoming T-End-Access TDU results in a T-END-ACCESS indication. A T-END-ACCESS confirmation primitive results in sending a T-Response-positive to acknowledge the T-End-Access, a negative response is not permitted.

C.3.2.6 T-Directory

T-Directory is used to request the transmission of a file directory from the Slave to the Master and it establishes a Transfer regime.

The response to a T-Directory is conveyed using the TDU T-Response-positive or T-Response negative.

C.3.2.6.1 Content of the T-Directory TDU and the associated responses

T-Directory

Designation
User data

T-Response-positive

T-Response-negative

Result

The use of the parameters is described in the service definition.

The Result parameter in T-Response-negative may indicate the following reasons:

- Rejection by the service user:
 - Reason not specified (default value);
 - Erroneous designation;
 - No answer to the request;
 - Erroneous user data;
 - Other reason (this last value may be followed by a string of no more than 62 displayable characters).

C.3.2.6.2 Sending T-Directory

A T-DIRECTORY request primitive results in sending a T-Directory TDU. Only the Master is permitted to send a T-Directory TDU when the Access regime is established and provided that no Transfer regime is established. Then the sender waits for a T-Response-positive or negative; no other action is permitted before reception of this response except aborting the association or reporting an exception. The reception of this response results in a T-DIRECTORY confirmation with the appropriate result code. The Transfer regime will be established as soon as a T-Response-positive is received. If a T-Response-negative, a T-P-Exception or a T-Abort is received, the Transfer regime is not established.

C.3.2.6.3 Receiving T-Directory

A valid incoming T-Directory TDU results in a T-DIRECTORY indication primitive.

A T-DIRECTORY response primitive results in sending a T-Response-positive or negative depending on the result parameter of the primitive. If a T-Response-positive is returned, the Transfer regime is established, otherwise the Transfer regime is not established.

C.3.2.7 T-Load

T-Load is used to request the transmission of a file from the Slave to the Master and it establishes a Transfer regime.

The response to a T-Load is conveyed using the TDU T-Response-positive or T-Response negative.

C.3.2.7.1 Content of the T-Load TDU and the associated responses

T-Load

Recovery point
Designation
User data

T-Response-positive

T-Response-negative

Result

The use of the parameters is described in the service definition.

The Result parameter in T-Response-negative may indicate the following reasons:

- Rejection by the service user:
 - Reason not specified (default value);
 - Erroneous designation;
 - Unknown file;
 - Erroneous recovery point;
 - Erroneous user data;
 - Other reason (this last value may be followed by a string of no more than 62 displayable characters).

C.3.2.7.2 Sending T-Load

A T-LOAD request primitive results in sending a T-Load TDU. Only the Master is permitted to send a T-Load TDU when the Access regime is established and provided that no Transfer regime is established. Then the sender waits for a T-Response-positive or negative; no other action is permitted before reception of this response except aborting the association or reporting an exception. The reception of this response results in a T-LOAD confirmation with the appropriate result code. The Transfer regime will be established as soon as a T-Response-positive is received. If a T-Response-negative, a T-P-Exception or a T-Abort is received, the Transfer regime is not established.

C.3.2.7.3 Receiving T-Load

A valid incoming T-Load TDU results in a T-LOAD indication primitive.

A T-LOAD response primitive results in sending a T-Response-positive or negative depending on the result parameter of the primitive. If a T-Response-positive is returned, the Transfer regime is established, otherwise the Transfer regime is not established.

C.3.2.8 T-Save

T-Save is used to announce the transmission of a file from the Master to the Slave and it establishes a Transfer regime.

The response to a T-Save is conveyed using the TDU T-Response-positive or T-Response-negative.

C.3.2.8.1 Content of the T-Save TDU and the associated responses

T-Save

Recovery point
Designation
User data

T-Response-positive

T-Response-negative

Result

The use of the parameters is described in the service definition.

The Result parameter in T-Response-negative may indicate the following reasons:

- Rejection by the service user:
 - Reason not specified (default value);
 - Erroneous designation;
 - Already existing file;
 - Erroneous recovery point;
 - Erroneous user data;
 - Other reason (this last value may be followed by a string of no more than 62 displayable characters).

C.3.2.8.2 Sending T-Save

A T-SAVE request primitive results in sending a T-Save TDU. Only the Master is permitted to send a T-Save TDU when the Access regime is established and provided that no Transfer regime is established. Then the sender waits for a T-Response-positive or negative; no other action is permitted before reception of this response except aborting the association or reporting an exception. The reception of this response results in a T-SAVE confirmation with the appropriate result code. The Transfer regime will be established as soon as a T-Response-positive is received. If a T-Response-negative, a T-P-Exception or a T-Abort is received, the Transfer regime is not established.

C.3.2.8.3 Receiving T-Save

A valid incoming T-Save TDU results in a T-SAVE indication primitive.

A T-SAVE response primitive results in sending a T-Response-positive or negative depending on the result parameter of the primitive. If a T-Response-positive is returned, the Transfer regime is established, otherwise the Transfer regime is not established.

C.3.2.9 T-Rename

T-Rename is used to change the designation of a file in the Slave's file system.

The response to a T-Rename is conveyed using the TDU T-Response-positive or T-Response negative.

C.3.2.9.1 Content of the T-Rename TDU and the associated responses

T-Rename

New name
Designation
User data

T-Response-positive

T-Response-negative

Result

The use of the parameters is described in the service definition.

The Result parameter in T-Response-negative may indicate the following reasons:

- Rejection by the service user:
 - Reason not specified (Default value);
 - Erroneous designation;
 - Erroneous new name;
 - Unknown file;
 - New name already in use;
 - Erroneous user data;
 - Other reason (this last value may be followed by a string of no more than 62 displayable characters).

C.3.2.9.2 Sending T-Rename

A T-RENAME request primitive results in sending a T-Rename TDU. Only the Master is permitted to send a T-Rename TDU when the Access regime is established and provided that no Transfer regime is established. Then the sender waits for a T-Response-positive or negative; no other action is permitted before reception of this response except aborting the association or reporting an exception. The reception of this response results in a T-RENAME confirmation with the appropriate result code.

C.3.2.9.3 Receiving T-Rename

A valid incoming T-Rename TDU results in a T-RENAME indication primitive.

A T-RENAME response primitive results in sending a T-Response-positive or negative depending on the result parameter of the primitive.

C.3.2.10 T-Delete

T-Delete is used to request the deletion of a file in the Slave's file system.

The response to a T-Delete is conveyed using the TDU T-Response-positive or T-Response-negative.

C.3.2.10.1 Content of the T-Delete TDU and the associated responses

T-Delete

Designation
User data

T-Response-positive

T-Response-negative

Result

The use of the parameters is described in the service definition.

The Result parameter in T-Response-negative may indicate the following reasons:

- Rejection by the service user:
 - Reason not specified (default value);
 - Erroneous designation;
 - Unknown file;
 - Erroneous user data;
 - Other reason (this last value may be followed by a string of no more than 62 displayable characters).

C.3.2.10.2 Sending T-Delete

A T-DELETE request primitive results in sending a T-Delete TDU. Only the Master is permitted to send a T-Delete TDU when the Access regime is established and provided that no Transfer regime is established. Then the sender waits for a T-Response-positive or negative; no other action is permitted before reception of this response except aborting the association or reporting an exception. The reception of this response results in a T-DELETE confirmation with the appropriate result code.

C.3.2.10.3 Receiving T-Delete

A valid incoming T-Delete TDU results in a T-DELETE indication primitive.

A T-DELETE response primitive results in sending a T-Response-positive or negative depending on the result parameter of the primitive.

C.3.2.11 T-Typed-data

T-Typed-data is used to transfer data in either direction, independently from the currently assigned roles.

There is no response required by the TDU layer entity.

C.3.2.11.1 Content of the T-Typed-data TDU

T-Typed-data

User data

The use of the parameter is described in the service definition.

C.3.2.11.2 Sending T-Typed-data

A T-TYPED-DATA request primitive results in sending a T-Typed-data TDU. Both entities are permitted to send a T-Typed-data TDU when the Access regime is established and provided that no Transfer regime is established.

C.3.2.11.3 Receiving T-Typed-data

A valid incoming T-Typed-data TDU results in a T-TYPED-DATA indication primitive.

C.3.2.12 T-Write

T-Write is used to convey data during a mass transfer phase.

The optional response to a T-Write is conveyed using the TDU T-response-positive or T-response-negative.

C.3.2.12.1 Content of the T-Write TDU and associated responses

T-Write

Explicit confirmation

First block/last block

Block number *

Data

T-Response-positive

Block number *

T-Response-negative

Block number *

C.3.2.12.1.1 First/last

This parameter indicates if the data transmitted in the T-Write TDU consist in the first block of data or the last block or both or none.

This parameter is set to “first” when the parameter “first block” of the T-WRITE request or T-WRITE-END request primitive is set to “first”.

This parameter is set to “last” when the TDU is issued subsequently to a T-WRITE-END request primitive.

C.3.2.12.1.2 Explicit confirmation

This parameter indicates whether or not a positive or negative confirmation is expected to the T-Write TDU.

In the case of the last block of data or when the proposed anticipation window size is greater than 1 or when the recovery function is in use, the value must be set to “explicit confirmation requested”.

C.3.2.12.1.3 Block number

This parameter may only be present when the symmetrical service class has been selected. It must be present when the proposed anticipation window size is greater than 1 or when the recovery has been selected.

In T-Write this parameter indicates the number of the block, this number is increased by 1 at each transmission of a T-Write.

In T-Response-positive, this number indicates the block number of the acknowledged T-Write.

In T-Response-negative this number indicates the block number of the first T-Write to be retransmitted.

In T-Response-positive or negative the block number parameter is present if and only if it was present in the corresponding T-Write.

C.3.2.12.1.4 Data

The T-Write TDU may contain a block of no more than 1024 octets of data if the Basic Kernel is used. In the symmetrical service the maximum size of a block is negotiated in the T-Access service.

C.3.2.12.2 Sending T-Write

A T-WRITE request or T-WRITE-END request primitive results in a T-Write TDU. This TDU is sent in a D-Data DDU. Only the Sender is permitted to send a T-Write TDU in the following cases:

- in the idle state of the Association regime if the basic kernel service class has been selected;
- in the idle state of the Access regime if the symmetrical service class has been selected and the Basic Transfer mode has been selected;
- after having sent a T-Response-positive to a T-Directory or T-Load TDU;
- after having received a T-Response-positive to a T-Save TDU;
- during a mass transfer phase.

If the Basic Kernel is being used or a window size of 1 is being used in the symmetrical service then, if explicit confirmation has been requested, the Sender waits for a T-Response-positive or negative, in this case no other action is permitted before reception of this response except aborting the association or reporting an exception.

If the window size is not equal to 1, then the sender may transmit up to the window size number of T-WRITE request or up to T-WRITE-END request before waiting for a T-Response except aborting the association or reporting an exception.

The reception of T-Response results in this case in a T-WRITE confirmation or T-WRITE-END confirmation primitive with the appropriate result code.

In the case of T-Response-negative on a T-Write which was not indicating “last block”, the transfer must be resumed from the first T-Write sent after the last T-Write for which a T-Response-positive had been received (or from the T-Write (first) if no T-Write was acknowledged since the beginning of the mass transfer phase).

C.3.2.12.3 Receiving T-Write

The mass transfer phase is initiated by the reception of a T-Write TDU with the parameter first/last indicating the beginning of the file.

A valid incoming T-Write TDU results in a T-WRITE indication or T-WRITE-END indication primitive. If explicit confirmation was requested, a T-Response-positive or negative must be sent in return.

NOTE – T-Write may also be rejected by T-Transfer-reject, T-Read-restart or T-P-Exception.

C.3.2.13 T-Transfer-reject

This TDU is used by the terminal to request the termination of a mass transfer.

C.3.2.13.1 Content

T-Transfer-reject

Reason

The Reason parameter is absent in case of user rejection, otherwise it contains a string of no more than 62 displayable characters (Other reason).

C.3.2.13.2 Sending T-Transfer-reject

A T-U-EXCEPTION-REPORT request with the “transfer reject” reason code results in a T-Transfer-reject TDU.

A T-Transfer-reject may be sent at any time during the mass transfer phase after having transmitted a T-Write. If the selected service class is the basic kernel, only the Receiver may send a T-Transfer-reject TDU.

C.3.2.13.3 Receiving T-Transfer-reject

A valid incoming T-Transfer-reject TDU results in a T-U-EXCEPTION-REPORT indication. The mass transfer phase is terminated.

C.3.2.14 T-Read-restart

This TDU is used by the Receiver to request the termination of a mass transfer and causes the retransmission of data from the beginning of the file.

C.3.2.14.1 Content

This TDU contains no parameter.

C.3.2.14.2 Sending T-Read-restart

A T-U-EXCEPTION-REPORT request with the “read restart” reason code results in a T-Read-restart TDU.

A T-Read-restart may only be sent by the Receiver, at any time during the mass transfer phase after having received a T-Write.

C.3.2.14.3 Receiving T-Read-restart

A valid incoming T-Read-restart results in a T-U-EXCEPTION-REPORT indication. The mass transfer is terminated. The Sender should restart the mass transfer from the beginning.

C.3.2.15 T-P-Exception

This TDU is used to abort any service in the Access regime.

C.3.2.15.1 Content

T-P-Exception

Reason

The Reason parameter may take the following values:

- Repeated negative acknowledgements;
- Protocol conflict;
- Syntax error/missing parameter;
- primitive not handled;
- Other reason (this last value may be followed by a string of no more than 62 displayable characters).

C.3.2.15.2 Sending T-P-Exception

A T-P-Exception may only be sent when the Access regime is established after the detection of an error by the service provider (see C.3.3). T-P-Exception may not be sent in the idle state of the Access regime.

Sending a T-P-Exception results in a local T-P-EXCEPTION-REPORT indication. Then the sending entity enters the idle state of the Access regime.

C.3.2.15.3 Receiving T-P-Exception

A valid incoming T-P-Exception results in a T-P-EXCEPTION-REPORT indication. Then the receiving entity enters the idle state of the Access regime.

C.3.2.16 T-Response-positive

This TDU is used to acknowledge the TDUs for which a confirmation is required by this protocol specification (see C.3.2.1 to C.3.2.12).

C.3.2.16.1 Content

T-Response-positive

TDU code	*
Other parameters	*

* This parameter may only be present if the symmetrical service class is selected.

The TDU code parameter indicates which TDU is acknowledged.

The other parameters depend on the TDU which is acknowledged (see C.3.2.1.1 to C.3.2.12.1).

C.3.2.16.2 Sending T-Response-positive

The conditions for sending a T-Response-positive are described in C.3.2.1.3 to C.3.2.12.3.

C.3.2.16.3 Receiving T-Response-positive

The actions to be taken on the reception of T-Response-positive are described in C.3.2.1.2 to C.3.2.12.2.

C.3.2.17 T-Response-negative

This TDU is used to refuse the TDUs for which a confirmation is required by this protocol specification (see C.3.2.1 to C.3.2.12).

C.3.2.17.1 Content

T-Response-negative

TDU code	*
Result	*

* This parameter may only be present if the symmetrical service class is selected.

The TDU code indicates which TDU is refused.

The Result parameter contains a reason which depends on the TDU which is refused (see C.3.2.1.1 to C.3.2.12.1)

C.3.2.17.2 Sending T-Response-negative

The conditions for sending a T-Response-negative are described in C.3.2.1.3 to C.3.2.12.3.

C.3.2.17.3 Receiving T-Response-negative

The actions to be taken on the reception of T-Response-negative are described in C.3.2.1.2 to C.3.2.12.2.

C.3.3 Exceptions and timers

C.3.3.1 Application response timer

This timer is used to control the maximum delay between sending a TDU corresponding to a service request and receiving the TDU corresponding to the service confirmation (T-Response-positive or negative) or any other event intended to terminate the service request (exception report, collision, etc.).

The delay is fixed by the service user and controlled by the service provider. At the Association establishment each entity indicates the maximum delay it requires to answer to a request. This delay is controlled by the peer entity.

If the proposed service class is basic kernel, only the initiator of the association indicate the value of the application response timer.

If this timer expires recovery may be attempted using the procedures described in C.3.3.2 or C.3.3.3.

NOTE – It is the host responsibility to ensure that this timer does not expire when started on a T-response-positive or negative. (In particular, user dialogue using non processable VPDE's should not take place within an association.)

C.3.3.2 Abnormal termination of the mass transfer

The following error conditions detected by the Receiver during a mass transfer phase cause the transmission by the Receiver of a T-Read-restart or T-Transfer-reject.

- occurrence of a DDU exception condition;
- reception of any TDU other than T-Write;
- expiration of the application response timer.

If more than five successive occurrences of the same error are detected, the Receiver may either send a T-Abort or D-U-Abort.

A mass transfer phase may also be abnormally terminated by:

- transmission of a D-U-Abort DDU by either entity;
- transmission of a T-Abort, T-Transfer-reject, T-Read-restart, T-P-Exception by either entity.

C.3.3.3 Errors outside a mass transfer phase

The following error conditions may occur outside a mass transfer phase:

- occurrence of a DDU exception condition;
- reception of a T-Write TDU;
- reception of an unknown or unexpected TDU;
- expiration of the application response timer.

In this case the entity may send a T-Abort or a T-P-Exception. If more than five occurrences of the same error are detected then the entity sends a D-U-Abort.

C.3.4 Use of DDU layer

The T-Associate TDU must be conveyed by the D-Set-mode DDU.

Other TDU's are conveyed in D-Data DDUs. It is possible to concatenate several TDUs in the same D-Data provided that only the last one is a TDU which requires confirmation at the TDU level and that the total length does not exceed the DDU maximum size.

The T-Responses to a TDU must never be concatenated with any other TDU.

When DDU modes A, B, D and E are used, T-Abort, T-Read-restart, T-Transfer-reject, T-Response-positive, T-Response-negative are sent without using any DDU.

When a symmetrical DDU mode is used, these TDUs are conveyed within D-Data DDUs and must never be concatenated with any other TDU.

If the symmetrical service class is proposed, a symmetrical DDU mode must be selected.

When a TDU is to be confirmed (T-Associate or T-Write with explicit confirmation requested, T-Write (last), T-Release, T-Directory, T-Load, T-Save, T-Rename or T-Delete), the confirmation flag must be set in the corresponding D-Data.

On indication of a DDU exception (unrecoverable error at the DDU level) the TDU layer may either try to recover (send a T-Read-restart, T-Transfer reject or T-P-Exception) or abort the TDU association. If more than five occurrences of the same error are detected then the entity sends a D-U-Abort.

C.3.5 Use of Syntax-based-videotex

The SBV_TPD (Transparent Processable Data) primitives may be used to allow for the exchange of processable data transparently without making use of the DDU. Both transparent processable data primitives (SBV_TPD_Begin and SBV_TPD_End) shall be supported.

The service primitive T-ASSOCIATE is linked to the SBV_TPD_Begin primitive. The service primitives T-RELEASE and T-ABORT are linked to the SBV_TPD_End primitive.

The initiator of the T-ASSOCIATE request must previously send a SBV_TPD_Begin request. The initiator of a T-RELEASE request must send a SBV_TPD_End when receiving a T-RELEASE confirmation.

The precise mechanisms are defined in the ETS 300 223.

The SBV_Establish and the SBV_Release must be processed outside the use of the TDU services.

C.4 Coding of TDUs

C.4.1 Coding of TDUs

The coding is a Type Length Value (TLV) encoding. Each TDU is identified by a Command Identifier and the total length of the TDU is indicated in a Length Indicator.

NOTE 1 – In the basic kernel the TDUs sent by the acceptor of the association only contain a Command Identifier without Length Indicator.

If the symmetrical service class has been selected by the acceptor of the association all the TDU's contain at least a length indicator and may be followed by a parameter field (i.e. LI may be equal to 0).

NOTE 2 – The whole TDU is subject to the 7-8 bit translation specified by the D-Set-mode or by the D-Data conveying this TDU.

C.4.1.1 Structure of TDUs

The general structure of TDUs is the following:

CI field	LI field	Parameter field
----------	----------	-----------------

or

CI field

C.4.1.1.1 Command Identifier field (CI)

The first byte of a TDU identifies the TDU according to the coding in Table C.3.

Table C.3/T.101 – Coding of TDU command identifiers

TDU	CI	
T-Associate	2/0	
T-Release	2/1	
T-Abort	3/8	
T-Access	2/2	*
T-End-Access	2/3	*
T-Directory	2/4	*
T-Load	2/5	*
T-Save	2/6	*
T-Rename	2/7	*
T-Delete	2/8	*
T-Typed-data	2/9	*
T-Write	2/15	
T-Transfer-reject	3/6	
T-Read-restart	3/7	
T-P-Exception	2/14	*
T-Response-positive	3/2	
T-Response-negative	3/3	
* Symmetrical service only.		

C.4.1.1.2 Length Indicator field (LI)

The length indicator is used to indicate the length of the parameter field. The value of the LI field is expressed as a binary number representing the length in bytes of the associated parameter field (i.e. the value of the LI field does not include itself nor the CI field).

LI fields indicating lengths within the range 0-254 shall comprise one byte.

LI fields indicating lengths within the range 255-65534 shall comprise three bytes. The first byte shall be coded 15/15 and the second and third bytes shall contain the length of the associated parameter field with the high order bits in the first of these two bytes.

C.4.1.1.3 Parameter field

The parameter field comprises zero, one or more groups of:

- *Parameter Identifier field (PI)* – A single byte identifying the parameter. The coding of PIs is specified in Table C.4.
- *Length Indicator field (LI)* – This length indicator is used to indicate the length of the following parameter value. This LI is coded as specified in C.4.1.1.2.
- *Parameter Value field (PV)* – The coding of the parameter values is specified in C.4.1.2 below.

NOTE – The data field in T-Write only contains a PV field without any PI nor LI field.

Table C.4/T.101 – Coding of TDU parameter identifiers

Parameter	PI	
User data	4/0	
Called address	4/1	*
Calling address	4/2	*
Result/Reason	4/3	*
Role/function	4/4	*
Application name	4/5	
Appl. resp. timeout	4/6	
Size/recovery/window	4/7	*
Designation	4/8	*
New name	4/9	*
Request identification	4/10	*
Identification	4/11	*
Explicit confirmation first/last, block number	4/12	
Transfer mode	4/13	*
Recovery point	4/15	*
Service class	5/1	
* Symmetrical service only.		

The absence of a parameter in a parameter field means that either the default value applies if defined or that no information is associated with this parameter.

If a PV field is absent, the whole parameter group must be absent (i.e. LI may not be equal to 0).

In order to ensure easier migration to enhanced versions of the protocol the following rules should apply:

- parameter fields with undefined parameter identifiers should be ignored in reception;
- reserved bits in a parameter value are set to zero and ignored in reception.

Bits in a byte are numbered from b₀ (least significant) to b₇ (most significant).

C.4.1.2 Coding of TDUs

C.4.1.2.1 T-Associate

C.4.1.2.1.1 Request

CI T-Associate	= 2/0
LI	= L
* PI Called address	= 4/1
LI	= n < 255
PV	= n bytes (see C.4.1.2.1.3)

*	PI Calling address	= 4/2
	LI	= n < 255
	PV	= n bytes (see C.4.1.2.1.4)
	PI Application name	= 4/5
	LI	= n < 17
	PV	= n bytes (see C.4.1.2.1.5)
	PI App. resp time-out	= 4/6
	LI	= 0/1
	PV	= 1 byte (see C.4.1.2.1.6)
	PI Service class	= 5/1
	LI	= 0/1
	PV	= 1 byte (see C.4.1.2.1.7)
	PI Explicit Confirm.	= 4/12
	LI	= 0/1
	PV	= 1 byte (see C.4.1.2.1.8)
*	PI Identification	= 4/11
	LI	= n < 32
	PV	= n bytes (see C.4.1.2.1.9)
*	PI Request identif.	= 4/10
	LI	= 0/1
	PV	= 1 byte (see C.4.1.2.1.10)
	PI User data	= 4/0
	LI	= n < 255
	PV	= n bytes (see C.4.1.2.1.11)

* These parameters may only be present in the symmetrical service class.

C.4.1.2.1.2 Responses

CI T-Response-positive		= 3/2
LI		= L
	PI Called address	= 4/1
	LI	= n < 255
	PV	= n bytes (see C.4.1.2.1.3)
	PI Result	= 4/3
	LI	= 0/1
	PV	= 2/0
	PI App. resp time-out	= 4/6
	LI	= 0/1
	PV	= 1 byte (see C.4.1.2.1.6)
	PI Identification	= 4/11
	LI	= n < 32
	PV	= n bytes (see C.4.1.2.1.9)
	PI User data	= 4/0
	LI	= n < 255
	PV	= n bytes (see C.4.1.2.1.11)

CI T-Response-negative	= 3/3
LI	= L
PI Result	= 4/3
LI	= n < 65
PV	= 2/0 + reason (see C.4.1.2.1.12)

In the basic kernel, the responses are coded on 1 byte:

T-Response positive	= 3/2
T-Response negative	= 3/3

If the symmetrical service class is proposed, the sender of T-Associate must be able to recognise the Response coding of both the basic kernel and the symmetrical service.

C.4.1.2.1.3 Called address

This parameter is a variable length string of no more than 254 bytes. If used then it is the responsibility of the application to insert an appropriate identification in this field.

C.4.1.2.1.4 Calling address

This parameter is a variable length string of no more than 254 bytes. If used then it is the responsibility of the application to insert an appropriate identification in this field.

C.4.1.2.1.5 Application name

This parameter is a variable length string of no more than 16 bytes containing an application name. The use of initial character 2/1 ('!') is reserved for standardised applications.

C.4.1.2.1.6 Application response time-out

This parameter is coded on 1 byte. The time-out interval is coded in binary form in increments of 1s.

The value 0/0 (default value) has the special meaning of 'timer disabled'.

C.4.1.2.1.7 Service class

This parameter is coded on 1 byte. The bit b_0 is set to 1 to indicate that the basic kernel is used. The bit b_1 is set to 1 to indicate that the symmetrical service is used. All other bits are reserved.

C.4.1.2.1.8 Explicit confirmation

This parameter is coded on 1 byte.

Bits b_0 , b_1 , b_2 , b_4 , b_5 , b_6 , b_7 are reserved.

$b_3 = 0$: explicit confirmation not requested

$b_3 = 1$: explicit confirmation requested

The default value is all bits set to 0 except b_3 set to 1.

If the symmetrical service class is proposed, then bit b_3 must be set to 1.

C.4.1.2.1.9 Identification

This parameter is a variable length string of no more than 31 bytes.

This string contains identification parameters separated by 2/15 (maximum length of each parameter is 12 bytes).

C.4.1.2.1.10 Request identification

This parameter is coded on 1 byte.

$b_0 = 0$: no Identification requested in the response (default value);

$b_0 = 1$: an Identification is requested in the response.

All other bits are reserved.

C.4.1.2.1.11 User data

This parameter is a variable length string, coded according to CCITT Recommendation T.51, of no more than 254 bytes.

C.4.1.2.1.12 Reason (in the Result parameter)

This value is a variable length string of no more than 63 bytes. The default value is: reason not specified. The coding of the reason values is specified in C.4.2.1.

C.4.1.2.2 T-Release

C.4.1.2.2.1 Request

CI T-Release	= 2/1
LI	= L
PI User data	= 4/0
LI	= n < 255
PV	= n bytes (see C.4.1.2.1.11)

C.4.1.2.2.2 Response

CI T-Response-positive	= 3/2
LI	= L
PI Result	= 4/3
LI	= 0/1
PV	= 2/1
PI User data	= 4/0
LI	= n < 255
PV	= n bytes (see C.4.1.2.1.11)

In the basic kernel, the response is coded on 1 byte: 3/2.

C.4.1.2.3 T-Abort

C.4.1.2.3.1 Request

CI T-Abort	= 3/8
LI	= L
PI Reason	= 4/3
LI	= n < 64
PV	= n bytes (see C.4.2.2)

In the basic kernel, this TDU is coded on 1 byte: 3/8.

C.4.1.2.4 T-Access

C.4.1.2.4.1 Request

CI T-Access	= 2/2
LI	= L
PI Role/function	= 4/4
LI	= n < 3
PV	= 1 or 2 bytes (see C.4.1.2.4.3)
PI Size/recovery/win.	= 4/7
LI	= 0/1
PV	= 1 byte (see C.4.1.2.4.4)

PI Transfer mode	= 4/13
LI	= 0/1
PV	= 1 byte (see C.4.1.2.4.5)
PI User data	= 4/0
LI	= n < 255
PV	= n bytes (see C.4.1.2.4.6)

C.4.1.2.4.2 Response

CI T-Response-positive	= 3/2
LI	= L
PI Result	= 4/3
LI	= 0/1
PV	= 2/2
PI Role/function	= 4/4
LI	= n < 3
PV	= 1 or 2 bytes (see C.4.1.2.4.3)
PI Size/recovery/win.	= 4/7
LI	= 0/1
PV	= 1 byte (see C.4.1.2.4.4)
PI Transfer mode	= 4/13
LI	= 0/1
PV	= 1 byte (see C.4.1.2.4.5)
PI User data	= 4/0
LI	= n < 255
PV	= n bytes (see C.4.1.2.4.6)
CI T-Response-negative	= 3/3
LI	= L
PI Result	= 4/3
LI	= n < 65
PV	= 2/2 + reason (see C.4.1.2.1.12)

C.4.1.2.4.3 Role/function

This parameter is coded on 1 or 2 bytes.

The first byte is coded as follows:

$b_0 = 0$: Slave

$b_0 = 1$: Master

If the role is **Slave**, the other bits indicate the TDUs which may be received by this entity:

$b_1 = 1$: T-Read-restart

$b_2 = 1$: T-Typed-data

$b_3 = 1$: T-Directory

$b_4 = 1$: T-Delete

$b_5 = 1$: T-Rename

$b_6 = 1$: T-Save

$b_7 = 1$: T-Load

If the role is **Master**, the other bits indicate the TDUs which may be received by this entity:

$b_1 = 1$: T-Read-restart

$b_2 = 1$: T-Typed-data

The other bits are reserved.

A second byte is reserved for future use and must not be sent in this version of the protocol.

No default value.

C.4.1.2.4.4 Size, recovery, window

This parameter is coded on 1 byte. It indicates:

$b_0 = 0$: no recovery accepted by the Slave outside the mass transfer

$b_0 = 1$: recovery accepted by the Slave outside the mass transfer

This bit has no meaning if the role is Master.

b_3, b_2, b_1 indicate the maximum size of the application data, contained in a T-Write or T-Write (last), accepted by the Receiver:

$b_3, b_2, b_1 =$	000: 512	100: 8192
	001: 1024	101: 16384
	010: 2048	110: 32768
	011: 4096	111: 65528

b_4 : reserved

b_7, b_6, b_5 indicate the maximum number of consecutive T-Write or T-Write (last) (Confirmation required) the Receiver can accept before issuing an answer:

$b_7, b_6, b_5 =$	000: 1	100: 5
	001: 2	101: 6
	010: 3	110: 7
	011: 4	111: 8

Default value: size: 1024 bytes, no recovery, window: 1.

C.4.1.2.4.5 Transfer mode

This parameter is coded on 1 byte as follows:

$b_0 = 0$: Basic Transfer mode not supported by Slave (if bit b_0 of Role/function is set to 0)

basic Transfer mode required by Master (if bit b_0 of Role/function is set to 1)

$b_0 = 1$: Basic Transfer mode supported by Slave (if bit b_0 of Role/function is set to 0)

basic Transfer required by Master (if bit b_0 of Role/function is set to 1).

$b_0 = 0$: Default value

All other bits are reserved.

C.4.1.2.4.6 User data

For Telesoftware and/or Printer Device applications see C.4.4.1.

C.4.1.2.5 T-End-Access

C.4.1.2.5.1 Request

CI T-End-Access = 2/3

LI = L

PI Reason = 4/3

LI = $n < 64$

PV = n bytes (see C.4.2.2)

PI User data	= 4/0
LI	= n < 255
PV	= n bytes (see C.4.1.2.1.11)

C.4.1.2.5.2 Response

CI T-Response-positive	= 3/2
LI	= L

PI Result	= 4/3
LI	= 0/1
PV	= 2/3

PI User data	= 4/0
LI	= n < 255
PV	= n bytes (see C.4.1.2.1.11)

C.4.1.2.6 T-Directory

C.4.1.2.6.1 Request

CI T-Directory	= 2/4
LI	= L

PI User data	= 4/0
LI	= n < 255
PV	= n bytes (see C.4.1.2.6.3)

PI Designation	= 4/8
LI	= n < 255
PV	= n bytes (see C.4.1.2.6.4)

C.4.1.2.6.2 Response

CI T-Response-positive	= 3/2
LI	= L

PI Result	= 4/3
LI	= 0/1
PV	= 2/4

CI T-Response-negative	= 3/3
LI	= L

PI Result	= 4/3
LI	= n < 65
PV	= 2/4 + reason (see C.4.1.2.1.12)

C.4.1.2.6.3 User data

For Telesoftware and/or Printer Device applications see C.4.4.2.

C.4.1.2.6.4 Designation

For Telesoftware and/or Printer Device applications see C.4.4.3.

C.4.1.2.7 T-Load

C.4.1.2.7.1 Request

CI T-Load	= 2/5
LI	= L
PI User data	= 4/0
LI	= n < 255
PV	= n bytes (see C.4.1.2.7.5)
PI Designation	= 4/8
LI	= n < 255
PV	= n bytes (see C.4.1.2.7.4)
PI Recovery point	= 4/15
LI	= n < 3
PV	= n bytes (see C.4.1.2.7.3)

C.4.1.2.7.2 Responses

CI T-Response-positive	= 3/2
LI	= L
PI Result	= 4/3
LI	= 0/1
PV	= 2/5
CI T-Response-negative	= 3/3
LI	= L
PI Result	= 4/3
LI	= n < 65
PV	= 2/5 + reason (see C.4.1.2.1.12)

C.4.1.2.7.3 Recovery point

This parameter is only present if this facility has been proposed and if the recovery is not performed from the beginning of the file.

The following byte(s) (maximum 2 bytes), if present, indicate(s) the block number.

The block number is coded in absolute binary form using all eight bits of each byte. It is coded on 1 byte between 0 and 255 and on 2 bytes between 256 and 65535. If 2 bytes are used then the first byte contains the most significant bits.

The recovery points are numbered consecutively and the first block is coded 0.

C.4.1.2.7.4 Designation

For Telesoftware and/or Printer Device applications see C.4.4.4.

C.4.1.2.7.5 User data

For Telesoftware and/or Printer Device applications see C.4.4.5.

C.4.1.2.8 T-Save

C.4.1.2.8.1 Request

CI T-Save	= 2/6
LI	= L
PI User data	= 4/0
LI	= n < 255
PV	= n bytes (see C.4.1.2.7.5)

PI Designation	= 4/8
LI	= n < 255
PV	= n bytes (see C.4.1.2.7.4)

PI Recovery point	= 4/15
LI	= n < 3
PV	= n bytes (see C.4.1.2.7.3)

C.4.1.2.8.2 Responses

CI T-Response-positive	= 3/2
LI	= L

PI Result	= 4/3
LI	= 0/1
PV	= 2/6

CI T-Response-negative	= 3/3
LI	= L

PI Result	= 4/3
LI	= n < 65
PV	= 2/6 + reason (see C.4.1.2.1.12)

C.4.1.2.9 T-Rename

C.4.1.2.9.1 Request

CI T-Rename	= 2/7
LI	= L
PI User data	= 4/0
LI	= n < 255
PV	= n bytes (see C.4.1.2.7.5)
PI Designation	= 4/8
LI	= n < 255
PV	= n bytes (see C.4.1.2.9.3)
PI New name	= 4/9
LI	= n < 255
PV	= n bytes (see C.4.1.2.9.3)

C.4.1.2.9.2 Responses

CI T-Response-positive	= 3/2
LI	= L

PI Result	= 4/3
LI	= 0/1
PV	= 2/7

CI T-Response-negative	= 3/3
LI	= L

PI Result	= 4/3
LI	= n < 65
PV	= 2/7 + reason (see C.4.1.2.1.12)

C.4.1.2.9.3 Designation, New name

The parameters Designation and New name contain the Transfer name and the New Transfer name of the file. (see C.4.1.2.7.4 for coding).

C.4.1.2.10 T-Delete

C.4.1.2.10.1 Request

CI T-Delete	= 2/8
LI	= L
PI User data	= 4/0
LI	= n < 255
PV	= n bytes (see C.4.1.2.7.5)
PI Designation	= 4/8
LI	= n < 255
PV	= n bytes (see C.4.1.2.7.4)

C.4.1.2.10.2 Responses

CI T-Response-positive	= 3/2
LI	= L
PI Result	= 4/3
LI	= 0/1
PV	= 2/8
CI T-Response-negative	= 3/3
LI	= L
PI Result	= 4/3
LI	= n < 65
PV	= 2/8 + reason (see C.4.1.2.1.12)

C.4.1.2.11 T-Typed-data

C.4.1.2.11.1 Request

CI T-Typed data	= 2/9
LI	= L
PI User data	= 4/0
LI	= n < 255
PV	= n bytes

C.4.1.2.12 T-Write

C.4.1.2.12.1 Request

CI T-Write	= 2/15
LI	= L
PI Explicit confirm	= 4/12
LI	= n < 4
PV	= n bytes (see C.4.1.2.12.4)
PV Data field	= N bytes

The Data field length is implicitly given as:

$$N = L - (n + 2) \text{ bytes}$$

In the symmetrical service N is less or equal to the value negotiated in the T-Access service.

In the basic kernel N is less than 1025 bytes.

C.4.1.2.12.2 Responses to a T-Write (not last)

CI T-Response-positive	= 3/2
LI	= L
PI Result	= 4/3
LI	= $n < 4$
PV	= $2/15 + \text{block number}$ (see C.4.1.2.12.5)

CI T-Response-negative	= 3/3
LI	= L
PI Result	= 4/3
LI	= $n < 4$
PV	= $2/15 + \text{block number}$ (see C.4.1.2.12.5)

In the basic kernel, the response is coded on one byte 3/2 or 3/3.

C.4.1.2.12.3 Responses to a T-Write (last)

CI T-Response-positive	= 3/2
LI	= L
PI Result	= 4/3
LI	= $n < 4$
PV	= $2/15 + \text{block number}$ (see C.4.1.2.12.5)
CI T-Response-negative	= 3/3
LI	= L
PI Result	= 4/3
LI	= $n < 67$
PV	= $2/15 + \text{block number} + \text{reason}$ (see C.4.1.2.12.5)

Symmetrical service:

- a T-Response-positive means file acceptance;
- a T-Response-negative means:
 - if there is a user reason (erroneous file, other reason), file refusal;
 - if there is no reason, last block(s) refusal.

Basic kernel:

The response is coded on 1 byte 3/2 or 3/3.

- a T-Response-positive means file acceptance;
- a T-Response-negative means file refusal.

C.4.1.2.12.4 Explicit confirmation, first/last, block number

This parameter is coded on 1 byte in the basic kernel, and 1 or more bytes in the symmetrical service.

The first byte contains the indication of explicit confirmation and the indication of first/last block.

Bits b_7 , b_6 , b_5 , b_4 , b_2 are reserved:

- $b_3 = 0$: explicit confirmation not requested
- $b_3 = 1$: explicit confirmation requested
- $b_1, b_0 = 00$: block
 - 01: first block
 - 10: last block
 - 11: first and last block

When $b_1 = 1$ (last block) b_3 must be set to 1.

The following byte(s) (maximum 2 bytes), if present, indicate(s) the block number.

The block number is coded in absolute binary form using all eight bits of each byte. It is coded on 1 byte between 0 and 255 and on 2 bytes between 256 and 65535. If 2 bytes are used then the first byte contains the most significant bits.

The recovery points are numbered consecutively and the first block is coded 0.

C.4.1.2.12.5 Result

This parameter contains the TDU code and the block number if present in the T-Write with the same coding. It may also contain the reason of the negative response to the T-Write last.

C.4.1.2.13 T-Transfer-reject

C.4.1.2.13.1 Request

CI T-Transfer-reject	= 3/6
LI	= L
PI Reason	= 4/3
LI	= n < 64
PV	= n bytes (see C.4.2.2)

In the basic kernel, this TDU is coded on 1 byte: 3/6.

C.4.1.2.14 T-Read-restart

C.4.1.2.14.1 Request

CI T-Read-restart	= 3/7
-------------------	-------

C.4.1.2.15 T-P-Exception

C.4.1.2.15.1 Indication

CI T-P-Exception	= 2/14
LI	= L
PI Reason	= 4/3
LI	= n < 64
PV	= n bytes (see C.4.2.2)

C.4.2 Coding of provider and user refusals

The reason codes are contained in the Result parameter of some TDUs to indicate either a user or a provider refusal. The following codes are allocated. If the user reason is not specified then no code is transmitted.

C.4.2.1 Reason codes in a T-Response negative

	<i>Provider</i>	<i>User</i>
Called address incorrect	3/0	4/0
Calling address incorrect	3/1	4/1
Role refused	3/2	4/2
Insufficient primitives handled		4/3
Application name unknown		4/4
Service class refused	3/5	4/5
Erroneous recovery point		4/6
Erroneous designation		4/7
No answer to the request		4/8
Unknown file		4/9
Already existing file		4/10
Erroneous file (T-Write last)		4/11

Erroneous new name	4/12
New name already in use	4/13
Wrong identity	5/0
Erroneous user data	6/0
Erroneous user data, service unknown	6/0, 6/1
Erroneous user data, group forbidden	6/0, 6/2
Other reason (+ optional string of char < 63 bytes)	6/15*

C.4.2.2 Reason in other TDUs

C.4.2.2.1 Provider reason

Repeated negative acknowledgements/repeated errors	7/0
Delay expired	7/1
Unknown message	7/2
Syntax error/missing parameter	7/3
Unrecoverable lower layer error	7/4
Protocol conflict	7/5
Primitive not handled	7/6
Other reason (+ optional string of char < 63 bytes)	6/15*

C.4.2.2.2 User reason

Wrong identification	5/0
Role refused	4/2
Insufficient primitives handled	4/3
Other reason (+ optional string of char < 63 bytes)	6/15*

* The string must contain displayable characters (codes 2/0 to 7/14 of the videotex primary graphic set).

C.4.3 File coding

The file coding scheme is a TLV encoding and uses the same principles as those described in C.4.1.1 for TDU's parameter field encoding.

C.4.3.1 File structure coding

Each file has the following structure:

HEADER	FILE CONTENT
--------	--------------

The description file has the following structure:

HEADER of description file	HEADER of File 1
	HEADER of File n

The structure of headers is given in C.4.3.2.

C.4.3.2 File header coding

The header is a sequence of attributes, each attribute is a group of PI LI PV.

The general structure of a header is the following:

PI FH	LI	PI	LI	PV		PI	LI	PV

The File Header PI (FH) is used to delimit the header of the file, its associated LI indicates the length of the list of attributes contained in the header. This parameter is always present but if there is no header then LI is set to 0/0.

The File Header PI is coded 3/0.

The following file attributes are specified (see Table C.5).

Table C.5/T.101 – Coding of file attributes

Attribute	PI
File type	2/0
Execution order	2/1
Transfer name	2/2
File name	2/3
Date	2/4
File length	2/5
Destination code	2/6
File coding	2/7
Destination name	2/8
Cost	2/9
User field/Application reference	2/10
Load address	2/11
Execute address (absolute)	2/12
Execute address (relative)	2/13
Compression mode	2/14
Device	2/15
File checksum	3/0
Author's name	3/1
Future file length	3/2
Permitted actions	3/3
Legal qualification	3/4
Creation	3/5
Last read access	3/6
Identity of the last modifier	3/7
Identity of the last reader	3/8
Recipient	3/9
Telematic file transfer version	3/10

C.4.3.2.1 File type

PI File type = 2/0
 LI = 0/1
 PV = 1 byte coded as follows

4/0 description file
 4/1 software file

4/2 data file
 4/3 command file
 4/4 text file (default value)
 All other values are reserved for future extension.

C.4.3.2.2 Execution order

PI Execution order = 2/1
 LI = 0/1
 PV = 1 byte coded as follows

 2/0 do not care (default value)
 2/1 highest execution order
 7/15 immediate processing

C.4.3.2.3 Transfer name

PI Transfer name = 2/2
 LI = n < 255
 PV = n bytes string

In the symmetrical service, the value is made up of keywords.

C.4.3.2.4 File name

PI File name = 2/3
 LI = n < 255
 PV = n bytes string

The value is a variable length string containing the file name.

C.4.3.2.5 Date

PI Date = 2/4
 LI = n < 13
 PV = n bytes string

The value is a variable length string of an even number of digits, coded in the range 3/0 to 3/9, specifying the date and time in the format yymmddhhmmss. The string may be truncated from the right to give a lower degree of accuracy.

C.4.3.2.6 File length

PI File length = 2/5
 LI = n < 9
 PV = n bytes

The value is a variable length parameter in which the expected length in bytes is coded in absolute binary form using all eight bits of each byte. If the parameter is coded in more than one byte, then the first byte to be transmitted contains the most significant bits.

NOTE – If a compression algorithm is used (see C.4.3.2.15), the value indicates the size in bytes of the file content before performing the compression algorithm.

C.4.3.2.7 Destination code

PI Destination code = 2/6
 LI = 0/1
 PV = 1 byte coded as follows

 4/0 do not care (default value)
 4/1 foreground memory
 4/2 background memory
 4/3 cassette tape
 All other values are reserved for future extension.

C.4.3.2.8 File coding

PI File coding	= 2/7
LI	= n < 255
PV	= n bytes string

The first byte of the string is coded as follows:

software file:	2/0 text source 2/1 source in tokenized form 2/2 intermediate code 2/3 object code 2/4 execution code (default value)
data file:	3/0 binary code (default value) 3/1 codes 2/0 to 7/14 of the videotex primary graphic set and the basic control codes specified in C.4.3.4.3.1
command file:	4/0 machine dependent (default value) 4/1 standard code
Text file:	5/0 other code 5/1 as for 3/1 above (default value) 5/2 Videotex code profile 2 5/3 Videotex code profile 1 5/4 Videotex code profile 3 5/5 geometric 5/6 photographic 5/7 sound All other values are reserved for future extension.

The default values indicated above depend on the value of the parameter file type and if file type is absent the default value is 5/1.

Subsequent bytes, if present, identify by means of a text string a language (such as “BASIC”, “P-CODE”), or a target processor (such as “6502”). This may optionally be followed by a 2/15 (‘/’) and an identification of the language dialect (such as ‘/MSDOS’).

C.4.3.2.9 Destination name

PI Destination name	= 2/8
LI	= n < 255
PV	= n bytes string

C.4.3.2.10 Cost

PI Cost	= 2/9
LI	= n < 32
PV	= n bytes string

The string must contain displayable characters coded according to CCITT Recommendation T.51.

C.4.3.2.11 User field

PI User field	= 2/10
LI	= n < 255
PV	= n bytes string

The string must contain displayable characters coded according to CCITT Recommendation T.51.

C.4.3.2.12 Load address

PI Load address	= 2/11
LI	= n < 9
PV	= n bytes

The value is a variable length parameter in which the load address is coded in absolute binary form using all eight bits of each byte. If the parameter is coded in more than one byte, then the first byte to be transmitted contains the most significant bits.

C.4.3.2.13 Execute address (absolute)

PI Execute address (absolute) = 2/12
 LI = $n < 9$
 PV = n bytes

The value is a variable length parameter in which the absolute execute address is coded in absolute binary form using all eight bits of each byte. If the parameter is coded in more than one byte, then the first byte to be transmitted contains the most significant bits.

C.4.3.2.14 Execute address (relative)

PI Execute address (relative) = 2/13
 LI = $n < 9$
 PV = n bytes

The value is a variable length parameter in which the relative execute address is coded in absolute binary form using all eight bits of each byte. If the parameter is coded in more than one byte, then the first byte to be transmitted contains the most significant bits.

C.4.3.2.15 Compression mode

PI Compression mode = 2/14
 LI = 0/1
 PV = 1 byte coded as follows

4/0	basic compression mode
4/1	high efficiency compression mode
4/15	“application defined” compression mode

The use of compression algorithm applies to the file content.

The value 4/0 identifies the compression algorithm as defined below.

To repeat a byte the following characters should be transmitted:

X, RPT, N.

where

- X is the byte to be repeated in the range 0/0 to 15/15;
- RPT is the character 1/2;
- N is coded 2/0 + n, n being the number of repetitions in the range 0/1 to 5/15.

When this compression algorithm is in use the RPT code (1/2) should be transmitted twice.

The value 4/1 identifies a compression mode based on the CCITT V.42 *bis* compression mechanism.

NOTE – The same rules as the ones defined in that CCITT Recommendation apply to the use of this algorithm.

The value 4/15 identifies an “application defined” compression algorithm, **not defined in this standard**. It is supposed that the receiver has the knowledge about the used compression algorithm.

All other values are reserved for future use. Absence of this parameter means that no compression algorithm is in use.

C.4.3.2.16 Device

PI Device = 2/15
 LI = $n < 255$
 PV = n bytes string

The value is a variable length string comprising one or more fields. Fields are separated by the character 4/0. The first field should contain a name which is sufficient to distinguish the device type; the use of other fields is unstandardized. The use of the initial character 2/1 ('!') in each field is reserved for standardized devices.

If the first field commences with 2/1, 5/0 (!P), then the required device is a printer. The coding of the data field is specified in C.4.3.4.3.

A third byte may be present in the first field to indicate the minimum printer character repertoire required, as follows:

2/1: codes 2/0-2/2, 2/5-5/10 and 6/1-7/10 of the Videotex primary graphic set, and the control codes specified in C.4.3.4.3.1.

Codes 2/3, 2/4, 5/11-5/15, 7/11-7/15 may also be used but the representation of these characters is not guaranteed in international communications and may be replaced by national application oriented variants.

2/2: the codes specified for 2/1, plus the mosaic characters in the first supplementary mosaic set.

2/3: the codes specified for 2/1, plus the SS2 control code, plus the following G2 characters: 2/3, 2/4, 2/6, 2/12, 2/13, 2/14, 2/15, 3/0, 3/1, 4/1, 4/2, 4/3, 4/8, 4/11, 6/10, 7/10.

The following characters must be displayed: à é ù è â ê î ô û ï ü æ ç Â É È î Ô Ù Æ Ç.

2/4: the codes specified for 2/3, plus SI and SO control codes, plus the characters of the second supplementary mosaic set (G1) except the codes 4/0 to 5/14.

2/5: the alphanumeric repertoire specified in section 2.1.1 of ETS 300 072, Part 1.

2/6: the alphanumeric and mosaic repertoires specified in sections 2.1.1 and 2.1.2 of ETS 300 072, Part 1.

2/7: the codes specified for 4/4, plus DRCS capability.

4/1: the codes specified for 2/1 plus the control codes specified in C.4.3.4.3.2.

4/2: the codes specified for 2/2 plus the control codes specified in C.4.3.4.3.2.

4/3: the codes specified for 2/3 plus the control codes specified in C.4.3.4.3.2.

4/4: the codes specified for 2/4 plus the control codes specified in C.4.3.4.3.2.

4/5: the codes specified for 2/5 plus the control codes specified in C.4.3.4.3.2.

4/6: the codes specified for 2/6 plus the control codes specified in C.4.3.4.3.2.

4/7: the codes specified for 2/7 plus the control codes specified in C.4.3.4.3.2.

4/15: private use.

All other values are reserved for future extension.

There may follow a string of arbitrary length comprising a sequence of codes of additional control and/or graphic characters or characters sequences which are required to be reproduced (for example, national currency signs, accented characters, selected line drawing characters, graphic renditions).

If the repertoire specification byte is 4/15, the subsequent characters may be used to define the repertoire, and possibly also the page format or special stationery to be used.

The second field, if present and of non-zero length, specifies the page format required. It may be either:

- a decimal number coded using the bytes 3/0 to 3/9, indicating that the printer must be able to print rows of at least this number of characters;
- a character indicating that each record is to be formatted to a locally specified line length, line breaks being inserted in place of spaces, or following hyphens ('-', 2/13). Permitted characters are 4/12 ('L') indicating that the text is to be left justified, and 4/10 ('J') indicating that text preceding an inserted line-break should be justified to span the available line-length.

C.4.3.2.17 File checksum

PI File checksum	= 3/0
LI	= 0/4
PV	= 4 bytes

The value is a 4-bytes length parameter containing a 32-bit frame check sequence.

The 32-bit FCS shall be the one's complement of the sum (modulo 2) of:

- a) the remainder of:

$x^k (x^{31} + x^{30} + x^{29} + x^{28} + x^{27} + x^{26} + x^{25} + x^{24} + x^{23} + x^{22} + x^{21} + x^{20} + x^{19} + x^{18} + x^{17} + x^{16} + x^{15} + x^{14} + x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x^1 + 1)$ divided (modulo 2) by the generator polynomial:

$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$, where k is the number of bits in the file content; and

- b) the remainder of the division (modulo 2) by the generator polynomial:

$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$, of the product of x^{32} by the file content.

For the transmitter, the initial content of the register of the device computing the remainder of the division is pre-set to all 1's and is then modified by division by the generator polynomial (as described above) of the file content; the one's complement of the resulting remainder is transmitted as the 32-bit FCS.

For the receiver, the initial content of the register of the device computing the remainder of the division is pre-set to all 1's. The final remainder, after multiplication by x^{32} and then division (modulo 2) by the generator polynomial $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ of the serial incoming protected bits and the FCS, will be "1100 0111 0000 0100 1101 1101 0111 1011 (x^{31} through x^0 , respectively) in the absence of transmission errors.

C.4.3.2.18 Author name

PI Author name	= 3/1
LI	= n < 255
PV	= n bytes string

The string must contain displayable characters coded according to CCITT Recommendation T.51.

C.4.3.2.19 Future file length

PI Future file length	= 3/2
LI	= n < 9
PV	= n bytes

The value is a variable length parameter in which the expected length in bytes is coded in absolute binary form using all eight bits of each byte. If the parameter is coded in more than one byte, then the first byte to be transmitted contains the most significant bits.

The nominal file length refers to the uncompressed file. Extension means an application dependent file operation and is different from the file decompression.

C.4.3.2.20 Permitted actions

PI Permitted actions	= 3/3
LI	= n < 255
PV	= n bytes string

If the first byte of the of the PV field as the value 4/0, the subsequent byte has the following meaning:

bit 0: read	bit 1: insert
bit 2: replace	bit 3: extend
bit 4: erase	bit 5: hide
bit 6: reserved	bit 7: not used

If the first byte of the PV field is coded 4/7, the subsequent string must contain displayable characters coded according to CCITT Recommendation T.51.

C.4.3.2.21 Legal qualification

PI Legal qualification	= 3/4
LI	= n < 255
PV	= n bytes string

The string must contain displayable characters coded according to CCITT Recommendation T.51.

C.4.3.2.22 Creation

PI Creation	= 3/5
LI	= n < 13
PV	= n bytes string

The value is a variable length string of an even number of digits, coded in the range 3/0 to 3/9, specifying the date and time in the format yymmddhhmmss. The string may be truncated from the right to give a lower degree of accuracy.

C.4.3.2.23 Last read access

PI Last read access	= 3/6
LI	= n < 13
PV	= n bytes string

The value is a variable length string of an even number of digits, coded in the range 3/0 to 3/9, specifying the date and time in the format yymmddhhmmss. The string may be truncated from the right to give a lower degree of accuracy.

C.4.3.2.24 Identity of the last modifier

PI Identity of the last modifier	= 3/7
LI	= n < 255
PV	= n bytes string

The string must contain displayable characters coded according to CCITT Recommendation T.51.

C.4.3.2.25 Identity of the last reader

PI Identity of the last reader	= 3/8
LI	= n < 255
PV	= n bytes string

The string must contain displayable characters coded according to CCITT Recommendation T.51.

C.4.3.2.26 Recipient

PI Recipient	= 3/9
LI	= n < 255
PV	= n bytes string

The string must contain displayable characters coded according to CCITT Recommendation T.51.

C.4.3.2.27 Telematic file transfer version

PI Telematic file transfer version	= 3/10
LI	= n < 255
PV	= n bytes string

The string must contain displayable characters coded according to CCITT Recommendation T.51.

C.4.3.3 Coding of the file content

C.4.3.3.1 Description file

The content of the description file is made of the file headers of all the other files part of the application to upload or to download.

Each file header is introduced by the parameter FH followed by the length indicator. The parameter value contains at least the Transfer name and the Size of the file.

C.4.3.3.2 Other file

The other files contain their own data. This data follows the file header. They are not introduced by a parameter identifier.

C.4.3.4 Content of some specific files

C.4.3.4.1 Text file resulting of a T-DIRECTORY Request

This file contains a list of file Transfer names.

Each Transfer name is identified by one or more key words (maximum 8).

If there is more than one keyword they are separated by “/”. A private field may follow the Transfer name, in this case it is introduced by the code 2/0 and it contains displayable characters (coded between 2/1 and 7/14). The maximum length of the file name including separators is of no more than 70 bytes, and the private field length including introducer is of no more than 10 bytes.

Whatever the coding (character coding or videotex coding) each field, including Transfer name and private field, is terminated by CR, LF (0/13, 0/10).

If the coding is videotex, there are no sequence US x,y inside a field and no codes HT, VT, BS. The file is made of pages of 24 rows and 40 characters. A given field does not overlap on two consecutive pages.

C.4.3.4.2 File of group B and C

These files are text files. They may be coded in character code or in videotex code.

If the coding is videotex, all the alphamosaic videotex coding can be used. The file is made of pages of 24 rows and 40 characters.

C.4.3.4.3 Coding of data intended for a standardized printer

This subclause specifies the coding of data intended for the standardized printer (device !P) in the mass transfer phase.

In the auxiliary device application each transfer is intended to commence a new page or form, although the transfer may extend to more than one page or form of text.

Graphic character codes will be encoded according to the ETSI videotex display syntax and to the device parameter of the file header.

Control codes will be interpreted by the terminal as described in C.4.3.4.3.1 and C.4.3.4.3.2.

C.4.3.4.3.1 Basic control codes

C.4.3.4.3.1.1 NULL

Coded 0/0. This character should be ignored.

C.4.3.4.3.1.2 Backspace (BS)

Coded 0/8. The preferred action is to overprint character(s) previously received; the default action is to print only the second character received.

C.4.3.4.3.1.3 Line feed (LF)

Coded 0/10.

C.4.3.4.3.1.4 Clear screen (CS)

Coded 0/12 and with the meaning “new page”.

C.4.3.4.3.1.5 Carriage return (CR)

Coded 0/13.

C.4.3.4.3.2 Extended control codes

The control codes specified for the virtual standardized printer are taken in general from the ETSI videotex display syntax or from ISO 6429. Terminals are expected to recognise the following control code syntaxes:

ESC (2/x ...) 3/x
ESC (2/x ...) 4/x
ESC (2/x ...) 5/x
ESC (2/x ...) 6/x
ESC (2/x ...) 7/x
CSI (3/x ...) (2/x ...) 4/x
CSI (3/x ...) (2/x ...) 5/x
CSI (3/x ...) (2/x ...) 6/x
CSI (3/x ...) (2/x ...) 7/x

Where (2/x ...) denotes zero, one or more occurrences of a code 2/x. ESC = 1/11, CSI = 1/11 5/11 or CSI = 9/11.

Any other single byte from columns 0, 1, 8, 9.

NOTES

- 1 The code sequences ESC 4/x and ESC 5/x have the same meaning as the codes 8/x and 9/x respectively.
- 2 Control codes and code sequences are non-spacing except for spacing Videotex display attributes (see C.4.3.4.3.2.8).

Terminals are expected to interpret and implement the following control codes:

C.4.3.4.3.2.1 Repeat (RPT)

Coded 1/2, X and with the meaning “repeat the last graphic character a further n times”, where n is the binary value of the six least significant bits of X.

C.4.3.4.3.2.2 Active position address (APA)

Coded 1/15, X', X, Y', Y (all parameters taken from columns 4 to 7 of the code table). This sequence causes the active position to be moved to the row specified by the bytes X' and X, and the column specified by the bytes Y' and Y, provided that the specified row number is equal or higher than the current row number on the page. (The effect of attempting to overwrite text on the current row, or of addressing a column greater than that available on the printer, is not specified). The row and column are specified as 12-bit values using the least significant 6 bits of 2 bytes, the first byte carrying the most significant bits. If both row and column address 0 are specified, then the active position should remain in the row and column in which it was situated before the APA sequence was received.

The operation of the following codes may be implementation dependant:

C.4.3.4.3.2.3 Unit separators (US)

Coded 1/15, X, Y (X code taken from columns 2 and 3 of the code table). This code sequence identifies the start of non-alphamosaic-coded data which may be ignored by the terminal unless it is within the specified repertoire of the terminal (e.g. DRCS definitions as required by the repertoire specification byte 4/5). The end of the data string which may be ignored will be marked by a US or APA sequence.

C.4.3.4.3.2.4 Partial line down (PLD)

Coded 9/11, 4/11. The following characters within the same record, or until the next CS, US or PLU character, are intended to be printed as subscripts; if PLU is already in effect then PLD has the effect only of cancelling the PLU.

C.4.3.4.3.2.5 Partial line up (PLU)

Coded 9/11, 4/12. The following characters within the same record, or until CS, US or PLD character, are intended to be printed as superscripts; if PLD is already in effect then PLU has the effect only of cancelling the PLD.

C.4.3.4.3.2.6 Select graphic rendition (SGR)

Coded 9/11, (P1, (3/11, P2, (3/11, P3, (3/11, P4,))) 6/13 where each parameter P_n may take one of the values:

3/1: bold or increased intensity

3/3: italic

3/4: underlined

3/9: crossed-out

Each parameter affects the representation of subsequent text up to the next SGR sequence, or to the end of the transfer if no further SGR is encountered.

C.4.3.4.3.2.7 Graphic size modification (GSM)

Coded 9/11, P1, 3/11, P2, 2/0, 4/2 where the parameters P1 and P2 are each encoded as decimal numbers using the bytes 3/0 to 3/9, specifying the required height and width respectively of subsequent characters as percentages of the default height and width. Each parameter affects the representation of subsequent text up to the next GSM sequence, or to the end of the transfer if no further GSM is encountered.

C.4.3.4.3.2.8 Videotex display attributes

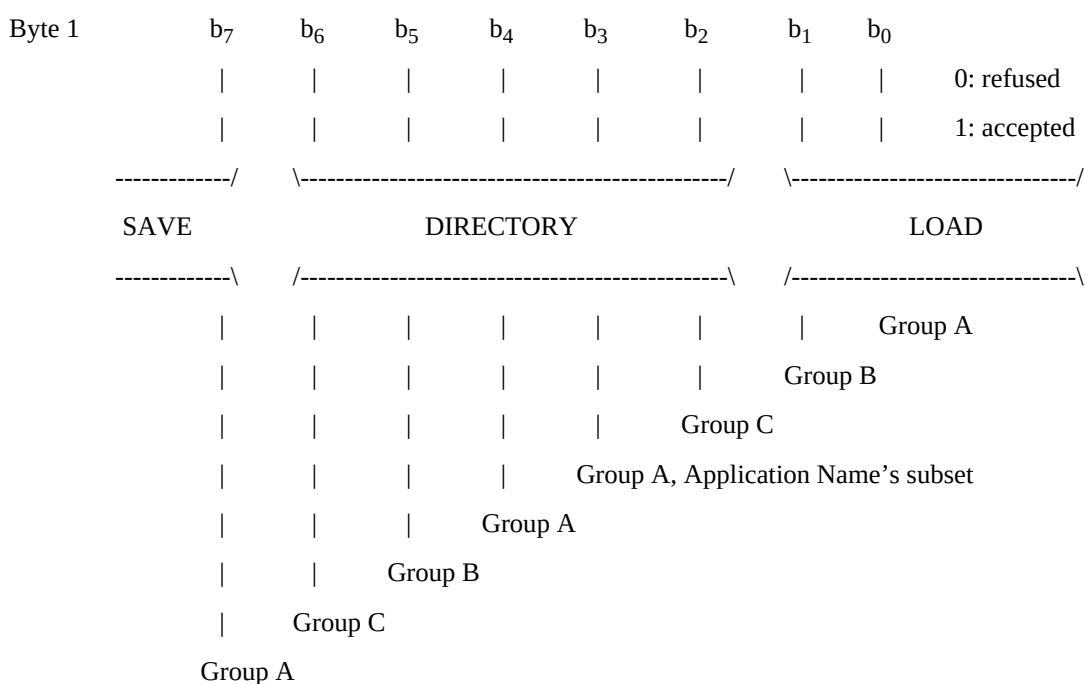
Coded 8/x or 9/x (or alternatively ESC 4/x or ESC 5/x). These attributes may be used to indicate a preferred foreground or background colour or character size for the printed text. (Use of a videotex display attribute to change character size is to be regarded as an alternative to the use of GSM, it does not alter the default height referenced by the GSM command. For example, the code 8/13 (double height) has an identical effect in the terminal to the code sequence 9/11, 3/2, 3/0, 3/0, 3/11, 3/1, 3/0, 3/0, 2/0, 4/2.

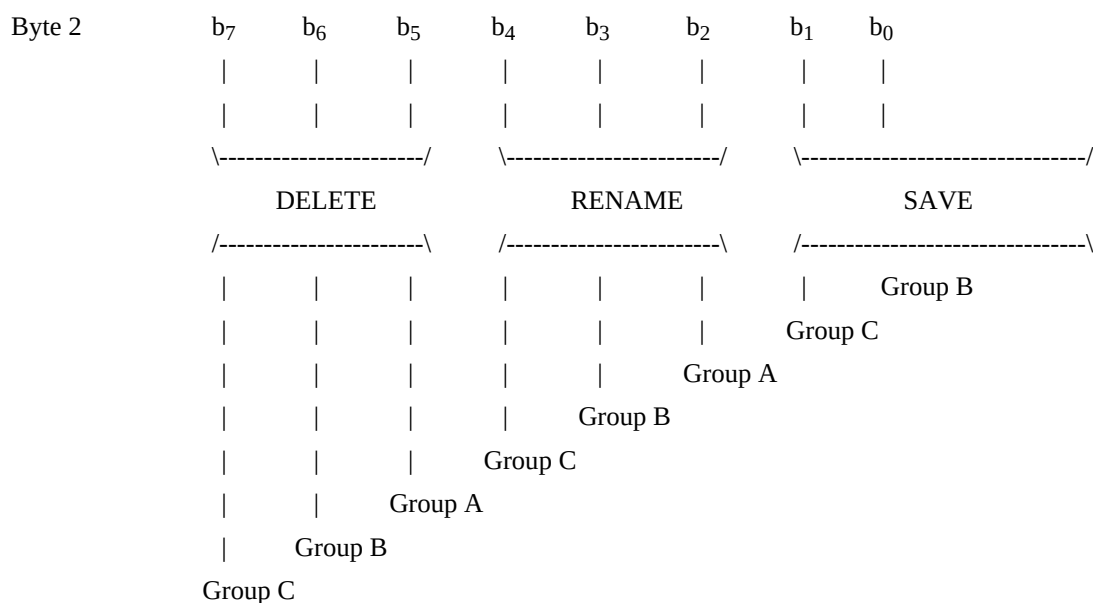
C.4.4 Coding of parameters for the telesoftware and printer device applications

C.4.4.1 User data in T-Access and in T-(Access) Response-positive

This parameter is used by the Slave to indicate in which groups the TDUs may be received:

Slave only:





Default value

The primitives T-DIRECTORY, T-LOAD, T-SAVE, T-RENAME, T-DELETE, if they are authorised, are only used in group A, which is equivalent to byte 1 equal to 8/9 and byte 2 equal to 2/4.

C.4.4.2 User data in T-Directory

This parameter is coded on up to 254 bytes, the first byte is coded as follows:

- 3/0: group A, description file subset
- 3/1: group A, default value
- 3/2: group B
- 3/3: group C.

C.4.4.3 Designation in T-Directory

The Designation parameter is made up of one or more elementary words separated by the codes 2/11 (displayed “+”), 2/15 (displayed “/”), 2/8 [displayed “(”] and 2/9 [displayed “)”].

In a byte sequence forming a correct sequence (see C.2.6 of Part 9 of the symmetrical service Telesoftware and printer device applications),

- 2/11 will be interpreted as the logical operator OR,
- 2/15 will be interpreted as the logical operator AND.

Parenthesis are used to indicate the scope of these logical operators.

The maximum number of elementary words is 8.

An elementary word is a byte sequence of no more than 12 bytes. Each byte can take any value between 2/1 and 7/14 excepted 2/8, 2/9, 2/11 and 2/15, moreover this word cannot contain more than one byte 2/10 (displayed “*”).

C.4.4.4 Designation in T-Load and T-Save

The Designation parameter contains the Transfer name.

A Transfer name is made up of one or more key words. If a Transfer name is made up of several key words, then they are separated by the code 2/15 (displayed “/”).

The maximum number of key words in a Transfer name is 8. The maximum length of a Transfer name is 70 bytes including all separators.

A key word is a byte sequence of no more than 12 bytes. Each byte can take any value between 2/1 and 7/14 excepted 2/8, 2/9, 2/10, 2/11 and 2/15.

C.4.4.5 User data in T-Load, T-Save, T-Rename, T-Delete

This parameter is coded on up to 254 bytes, the first byte is coded as follows:

- 3/1: group A, Default value
- 3/2: group B
- 3/3: group C

C.5 D-Protocol specification

C.5.1 Definitions

DDU Dialogue Data Unit

VPDE Videotex Presentation Data Element (see ETS 300 072)

Videotex frame: The data that is retrieved by a single command from a videotex terminal.

BCS Block Check Sequence

ED Error Detection Mechanism

translation mode: 4 translation modes are defined in order to provide for 8-bit transparency according to the videotex environment in which the protocol is to be used, these modes are specified in C.5.5.

- Mode 1: No translation;
- Mode 2: 3-in-4 coding;
- Mode 3: Shift scheme 8-bits;
- Mode 4: Shift scheme 7-bits.

DDU modes: 7 DDU modes are defined in order to adapt the DDU delimitation codes to the videotex environment in which the protocol is to be used. These modes are specified in C.5.5.

- Mode A: 1 byte response encoding;
- Mode B: 1 byte plus terminator response encoding;
- Mode C: Symmetrical encoding with a one byte terminator;
- Mode D: Redefinable response encoding;
- Mode E: Redefinable response encoding with an End Delimiter;
- Mode F: Symmetrical encoding with an End Delimiter;
- Mode G: Symmetrical encoding.

Other modes might be added to cover possible identified requirements related to the studies on the symmetrical service.

C.5.2 General overview of the protocol

C.5.2.1 Structure of DDUs

The Dialogue Data Units are used to delimit TDU's. DDU's are introduced by the Videotex processable data delimiter and are delimited by their internal syntax (see C.5.5). Optional 8-bit transparency capabilities (translation modes) and optional error detection facilities (BCS) are provided, but either or both of these features may be omitted if they are already provided by other means.

C.5.2.2 Flags

Flags are associated with D-Set-mode and D-Data DDU's. These flags are used by the error recovery mechanism. Only one of these flags may be set in a given DDU.

The more flag indicates, if set that this is not the last processable data VPDE in the videotex frame, further processable data VPDEs should be expected without request.

The poll flag indicates, if set, that no further processable data VPDEs will be sent until requested, and that the receptor is expected to transmit a D-Response-positive if this block and all previous blocks have been received correctly.

The confirmation flag indicates, if set that no further processable data VPDEs will be sent until requested, and that the Slave is expected to return the TDU acknowledgement, after this block and all previous blocks have been received correctly and processed by the application. This flag is set according to the semantic of the TDU which is conveyed by the corresponding D-Set mode or D-Data.

In symmetrical service:

The no flag is used when ED is not used and if the D-Data DDU contains only a T-Abort, a T-Typed-data, a T-Transfer-reject, a T-Read-restart, a T-P-Exception or a T-Response.

The poll flag is used only if ED is in use.

C.5.2.3 Error detection and recovery

This subclause gives an overview of the error detection and recovery mechanism.

The error detection is provided by using a sequence numbering or both a sequence numbering and a checksum. The use of error detection can be selected by sending a D-Set-mode indicating the use of either sequence numbering (mode E and F only) or both sequence numbering and checksum (all modes).

The main principles of the error recovery mechanism are described below. The complete specification is given in C.5.4. This mechanism makes use of the sequence numbering and the BCS.

On receipt of an erroneous DDU (e.g. detected with an incorrect BCS or DDU out of sequence) the receiver must send a D-Response-negative.

If ED is in use, the TDU's contained in the D-Data DDU's are delivered to the upper layer when a poll or confirmation flag is contained in the last correctly received D-Data.

If ED is not in use the TDU's contained in the D-Data DDU's are delivered as soon as DDU's are received.

When a D-Response-negative is received the sender must send again the DDUs following the last DDU with a poll or confirmation flag set and for which a D-Response-positive or a T-Response has been received.

D-Response-negative is only used when ED is in use.

Two timers are defined for the purpose of error detection:

- *General receive inactivity timer* – This timer is only active while processable-data reception is active. It is started by the reception of a processable-data delimiter and restarted by the reception of any data. It is stopped by the reception of a complete DDU, unless the More flag has been set.

- *DDU-request timer* – This timer is started by the transmission by the receptor of a D-Response-positive or a D-Response-negative. It is stopped by the reception of a DDU with a sequence code indicating either the required sequence number or an unnumbered DDU.

C.5.3 Repertoire and use of Dialogue Data Units

This subclause describes the DDUs and their parameters (see Table C.6).

Table C.6/T.101 – Dialogue data units

DDU	Parameters	Mandatory/Optional
D-Set-mode	Translation mode	M
	DDU mode	M
	Flags	M
	Use of sequence numbering	O
	Use of BCS	O
	Define D-Response-positive	O
	Define D-Response-negative	O
	Inactivity timer	O
	Request timer	O
D-Data	Sequence Code	O
	Translation mode	M
	Flags	M
	Reset	O
	Define D-Response-positive	O
	Define D-Response-negative	O
D-U-Abort	None	–
D-Response-positive	None	–
D-Response-negative	None	–
M Mandatory O Optional		

C.5.3.1 D-set-mode

D-Set-mode announces the intended use of processable data VPDE's. D-Set-mode selects the translation mode and the DDU mode. D-Set-mode resets the sequence number to 0 for both direction of transmission and resets all DDU variables to default unless specified by one of its optional parameters.

C.5.3.1.1 Content of D-set-mode DDU

C.5.3.1.1.1 Translation mode

This parameter defines the 7-8 bits translation technique to be used. The values of this parameter specify that the processable data facility is to be used and define the data coding scheme in use as described in C.5.5. This parameter takes effect immediately, and will therefore affect the decoding of any subsequent variable-length parameter values in the same VPDE as well as subsequent VPDE's. Translation mode 1 may not be used when DDU mode C is used.

C.5.3.1.1.2 DDU-mode

This parameter specifies the DDU mode to be used as described in C.5.5. This parameter takes effect immediately.

C.5.3.1.1.3 Flags

This parameter indicates which flags are associated with D-Set-mode (see C.5.2.2).

C.5.3.1.1.4 Use of ED

This parameter indicates that either sequence numbering or both BCS and sequence numbering will be used.

C.5.3.1.1.5 Define D-Response-positive

This parameter may only be used when DDU mode D or E or F are selected, it specifies the coding of D-Response-positive. The redefinition has immediate effect. The default value is 3/0.

C.5.3.1.1.6 Define D-Response-negative

This parameter may only be used when DDU mode D or E or F are selected, it specifies the coding of D-Response-negative. The redefinition takes effect after the reception of a D-Response-positive. The default value is 3/1.

NOTE – To guard against possible deadlock in the event of double error, the sender should recognize both the previous and new response negative until a response positive has been received.

C.5.3.1.2 Sending D-Set-mode

In the basic kernel, D-Set-mode may only be sent by the host. In the symmetrical service D-Set-mode may be sent by both entities.

D-Set-mode must not be preceded by any other processable data VPDE in a videotex frame.

If the poll flag is set, the sender must wait until a D-Response-positive or a D-Response-negative is received.

If the confirmation flag is set the sender must wait until the reception of a TDU.

C.5.3.1.3 Receiving D-Set-mode

On reception of D-Set-mode, the processable data facility is activated and the translation mode and DDU modes are selected according to the parameters of D-set-mode.

If the DDU is accepted and if the poll flag is set, a D-Response-positive must be returned.

If the confirmation flag is set the DDU layer waits for a local T-Response or an incoming DDU.

If D-set mode or one of its parameters is rejected, a D-U-Abort must be returned.

C.5.3.2 D-Data

This DDU is used to carry data for the upper layer (TDU's).

C.5.3.2.1 Content of D-Data DDU

C.5.3.2.1.1 Sequence code

This parameter contains a sequence number if the sequence numbering is in use (see C.5.4.2), otherwise this parameter is absent.

C.5.3.2.1.2 Translation mode

As specified in C.5.3.1.1.1 above.

C.5.3.2.1.3 Flags

This parameter indicates which flags are associated with D-Data (see C.5.2.2).

C.5.3.2.1.4 Reset

This optional parameter enables selective reset of the sequence number to 0, and/or of the code(s) for D-Response-positive and D-Response-negative to default. Default is no reset. In case of conflict with the redefinition parameters (see C.5.3.2.1.5 and C.5.3.2.1.6) the most recently received parameter takes precedence. In the case of reset of a D-Response-negative, the reset takes effect after a D-Data with a poll or confirmation flag set has been accepted (see C.5.3.1.1.6).

C.5.3.2.1.5 D-Response-positive

As specified in C.5.3.1.1.5 above but default is no change.

C.5.3.2.1.6 D-Response-negative

As specified in C.5.3.1.1.6 above but default is no change.

C.5.3.2.2 Sending D-Data

In the basic kernel, D-Data may only be sent by the host. In the symmetrical service D-Data may be sent by both entities.

If the poll flag is set the sender must wait until the reception of a D-Response-positive or negative.

If the confirmation flag is set the sender must wait until the reception of a TDU.

C.5.3.2.3 Receiving D-Data

On reception of D-Data, the data is delivered to the upper layer if the poll or confirmation flag is set. If the more flag is set the data is buffered.

On reception of a valid D-Data, if the poll flag is set a D-Response-positive must be sent, if the confirmation flag is set the DDU layer waits for a local T-Response or an incoming DDU.

If ED is in use and the D-data is invalid (see C.5.4.6) a D-Response-negative must be sent.

C.5.3.3 D-U-Abort

This DDU is used to abruptly terminate the use of processable data VPDE.

C.5.3.3.1 Content of D-U-Abort DDU

This DDU contains no parameter.

C.5.3.3.2 Sending D-U-Abort

Sending D-U-Abort causes the immediate termination of the DDU layer.

D-U-Abort may be sent at any time by the master after having sent a D-Set-mode. No other processable VPDE must be sent if a D-Set-mode is sent.

D-U-Abort may be sent at any time by the slave after having received a D-Set-mode. All other processable VPDEs will be discarded until a D-Set-mode is received.

C.5.3.3.3 Receiving D-U-Abort

Receiving a D-U-Abort causes the immediate termination of the DDU layer. All other processable VPDE will be discarded until a D-Set-mode is transmitted.

C.5.3.4 D-Response-positive

D-Response-positive is used to acknowledge a received DDU with the poll flag set.

C.5.3.4.1 Content of D-Response-positive DDU

No parameter is associated with this DDU.

C.5.3.4.2 Sending D-Response-positive

A D-Response-positive is sent on the correct reception of a DDU with the poll flag set.

When sending a D-Response-positive the DDU-request timer is started.

C.5.3.4.3 Receiving D-Response-positive

After reception of a D-response positive another DDU may be sent.

C.5.3.5 D-Response-negative

D-Response-negative is used to indicate an error in the process of receiving DDUs.

C.5.3.5.1 Content of D-Response-negative DDU

No parameter is associated with this DDU.

C.5.3.5.2 Sending D-Response-negative

A D-Response-negative may be sent on detection of a DDU error or timer expiration (see C.5.4). When sending a D-Response-negative the DDU-request timer is started.

C.5.3.5.3 Receiving D-Response-negative

On reception of D-response negative, the transmission is resumed from the DDU following the last acknowledged DDU (see C.5.4.6).

C.5.4 Error detection and recovery mechanism

C.5.4.1 Use of BCS

A checksum (BCS) is associated with each DDU (D-Set-mode, D-Data) if the “Use of BCS” parameter is set in the initial D-Set-mode DDU.

It is recommended not to use BCS when a data link layer is available.

The BCS is used to detect DDU transmission errors. The BCS encoding is specified in C.5.5.

C.5.4.2 Use of sequence numbering

Sequence numbering of the DDU (D-Set-mode, D-Data) is used if the “Use of BCS” or the “Use of Sequence Numbering” is set in the D-Set-mode DDU.

The sequence number is set to 0 when sending or receiving the D-Set-mode DDU. This sequence number is incremented by one with each D-Data.

The first D-Data after D-Set-mode has a sequence number of 1. The sequence number is reset to 0 by using the “reset” parameter in D-Data.

An internal variable Sn contains the value of the next sequence number.

Another internal variable S_a contains the value of the sequence number associated with the last acknowledged D-Data.

S_n and S_a are set to 0 when sending (resp. receiving) D-Set-mode or D-Data with the “reset” parameter.

S_n is incremented by one after having sent a D-Data without the “reset” parameter. If ED is in use, the “sequence code” parameter contains the current value of S_n plus one. When S_n is equal to 31 the incrementation of S_n sets S_n to 0.

S_a is set to S_n when receiving (resp. sending) a D-Response-positive or a T-Response (which is considered as a positive response by the DDU layer).

NOTE 1 – When Symmetrical DDU modes are used (C, F, G) the sequence numbering mechanism applies to both directions independently (i.e. a set of S_n and S_a is associated with each direction of transmission).

D-Set-mode resets both sets of variables while the “reset” parameter of D-Data resets only the set of variables associated with the direction of transmission of D-Data.

A sequence number is detected as invalid when ED use has been selected and:

- no recovery has been initiated and the sequence code indicates a sequence number different from $S_n + 1$;
or
- recovery has been initiated and the sequence code indicates a sequence number greater than S_n or lower than $S_a + 1$,

NOTE 2 – All these value comparisons are modulo 32.

C.5.4.3 Use of flags

If ED is in use, data (i.e. TDU's) are delivered to the upper layer upon reception of a valid DDU with a poll or confirmation flag set or upon reception of a valid D-U-Abort.

NOTE – Valid DDU's are DDU's which do not lead to detection of DDU exception or error (see C.5.4.6 and C.5.4.7).

If BCS is not in use, data are delivered to the upper layer as soon as DDU's are received.

Due to limitations of sequence number encoding, the maximum number of subsequent D-Data without poll or confirmation flag must not exceed 31.

C.5.4.4 Size of DDUs

A parameter of D-Set-mode indicates whether the maximum size of the data contained in D-Data must not exceed 2048 bytes or if there is no limit on this size.

If the basic kernel is used or if ED is in use this parameter must indicate that this size is limited to 2048 bytes.

When ED is in use, for buffering reasons, the maximum size of data between two poll or confirmation flags should not exceed 2048 octets.

NOTE – All these sizes are the number of bytes which are transmitted on the line taking into account the translation mode mechanism.

C.5.4.5 Use of Timers

Two timers are defined for the purpose of DDU loss detection:

- *General receive inactivity timer* – This timer is only active while processable-data reception is active. It is started by the reception of a processable-data delimiter and restarted by the reception of any data. It is stopped by the reception of a complete DDU, unless the More flag has been set.
- *DDU-request timer* – It is started by the transmission of a D-Response-positive or a D-Response-negative if ED is in use. It is stopped by the reception of a DDU with a sequence code indicating the required sequence number if the ED is in use or by reception of a DDU without sequence code.

C.5.4.6 Actions in the event of DDU errors

In the event of the following DDU reception errors, the entity which detects the error should transmit a D-Response-negative to request retransmission:

- a DDU with an invalid sequence number (see C.5.4.2) (only when ED is in use);
- a DDU in excess of permitted length (only when ED is in use);
- a sequence of DDU's without poll or confirmation flags conveying more than 2048 octets (only when ED is in use);
- a BCS error.

The entity which receives a D-Response-negative must send again all the DDU's and their content from the last acknowledged DDU (DDU starting from $S_a + 1$).

C.5.4.7 Actions in the event of DDU exceptions

The following exceptions should not cause transmission of a D-Response-negative but are indicated to the upper layer and may cause higher level error recovery to be initiated:

- Reception of D-Set-mode DDU on an already established DDU connection.
- Unacceptable parameters or parameter values in a received D-Data DDU, (if ED is not in use, the sequence code parameter value is not significant).
- Protocol error (syntactically incorrect DDU, or unrecognized DDU identifier, or unexpected DDU, or, if ED is not in use, DDU in excess of permitted length).
- Excessive retransmission of D-Response-negative (more than 5 retransmissions).
- Unrecoverable error from the data link layer (if such indication is available).
- Expiration of general receive inactivity timer or DDU request timer when no sequence numbering is in use.
- A VPDE other than a processable data VPDE is received during the reception of a processable data VPDE [that is a single 1/15 (US) character followed by a character other than 3/14]. In this case the subsequent VPDE should be processed.

C.5.4.8 Actions in the event of timer expiration

If the general receive inactivity timer or the DDU request timer expires, and if the sequence numbering is in use their recovery may be attempted by issuing a D-Response-negative. The number of retransmission of D-Response-negative is limited to 5 (see C.5.4.7).

C.5.5 Coding

C.5.5.1 Translation modes

The following transcoding schemes apply to all variable length DDU parameter and the whole of each TDU.

C.5.5.1.1 Mode 1 (no translation)

Under this scheme no translation of data is performed, except that all US (1/15) characters in the processable data are represented by two contiguous US characters in the transmitted data stream.

C.5.5.1.2 Mode 2 (3-in-4 coding)

Each group of three bytes in the processable data is mapped into four bytes for transmission as shown in Table C.9. Any remaining group of one or two bytes at the end of a block of data is mapped into two or three bytes respectively, with undefined bits set to zero. [A block consists of all bytes of a DDU to which the translation algorithm is applied (see C.5.5.3.1).]

C.5.5.1.3 Mode 3 (shift scheme – 8 bits)

In this scheme, bytes of processable data are each mapped into one or two bytes of transmitted data as shown in Table C.7; in mode 3 the most significant bit of each transmitted byte is taken into account. Note that most of the conversions are optional. Either entity should be prepared to accept any mixture of converted or unconverted data in these cases.

TABLE C.7/T.101

Code conversion for 8 bits shift scheme (mode 3)

Processable data	Sender's conversion	Optional/Mandatory		Transmitted data
		Mode C	A, B, D, E, F, G	
0/0-0/12	$7/14 x + 5/0$	–	O	7/14, 5/0-5/12
0/13	$7/14 x + 5/0$	M	O	7/14, 5/13
0/14-01/14	$7/14 x + 5/0$	–	O	7/14, 5/14-6/14
1/15	$7/14 x + 5/0$	M	M	7/14, 6/15
2/0	7/13	–	O	7/13
2/1-7/10	none			2/1-7/10
7/11	$7/11 x - 5/8$	M	M	7/11, 2/3
7/12, 7/13	$7/11 x - 5/8$	–	M	7/11, 2/4-2/5
7/14	$7/11 x - 5/8$	M	M	7/11, 2/6
7/15	$7/11 x - 5/8$	–	M	7/11, 2/7
8/0-13/0	$7/11 x - 5/8$	–	O	7/11, 2/8-7/8
13/1-15/15	$7/14 x + 5/0$	–	O	7/14, 2/1-4/15
– Irrelevant O Optional M Mandatory				

C.5.5.1.4 Mode 4 (shift scheme – 7 bits)

In this scheme bytes of processable data are each mapped into one or two bytes of transmitted data as shown in Table C.8; in mode 4 the most significant bit of each transmitted byte is not taken into account. Note that some of the conversions are optional. Either entity should be prepared to accept any mixture of converted or unconverted data in these cases.

TABLE C.8/T.101

Code conversion 7 bits shift scheme (mode 4)

Processable data	Sender's conversion	Optional/Mandatory	Transmitted data
0/0-1/14	7/14, $x + 5/0$	O	7/14, 5/0-6/14
1/15	7/14, 6/15	M	7/14, 6/15
2/0	7/13	O	7/13
2/1-7/10	none	–	2/1-7/10
7/11-7/15	7/11, $x - 5/8$	M	7/11, 2/3-2/7
8/0-13/0	7/11, $x - 5/8$	M	7/11, 2/8-7/8
13/1-15/15	7/14, $x + 5/0$	M	7/14, 2/1-4/15
– Irrelevant O Optional M Mandatory NOTE – In mode 4 the transmitted bytes may have the most significant bit set.			

TABLE C.9/T.101

3-in-4 coding scheme

Processable-data sequence	3 bytes	3 bytes	..	1, 2 or 3 octets
Transmitted-data sequence	4 bytes	4 bytes	..	2, 3 or 4 bytes respectively

Within each group the bits are mapped as follows, where 'b_{xy}' denotes bit y of byte x in the user data. Bit 7 is not taken into account by this scheme but may be determined by characteristics of the transmission path in use (for example, if parity is required).

a) *Three bytes of processable data*

Transmitted	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
1st byte	X	1	b ₁₇	b ₁₆	b ₂₇	b ₂₆	b ₃₇	b ₃₆
2nd byte	X	1	b ₁₅	b ₁₄	b ₁₃	b ₁₂	b ₁₁	b ₁₀
3rd byte	X	1	b ₂₅	b ₂₄	b ₂₃	b ₂₂	b ₂₁	b ₂₀
4th byte	X	1	b ₃₅	b ₃₄	b ₃₃	b ₃₂	b ₃₁	b ₃₀

b) *Two bytes of processable data at end of sequence*

Transmitted	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
1st byte	X	1	b ₁₇	b ₁₆	b ₂₇	b ₂₆	0	0
2nd byte	X	1	b ₁₅	b ₁₄	b ₁₃	b ₁₂	b ₁₁	b ₁₀
3rd byte	X	1	b ₂₅	b ₂₄	b ₂₃	b ₂₂	b ₂₁	b ₂₀

NOTE – in the case of the checksum, b₀-b₇ of the checksum map to b₁₀-b₁₇, and b₈-b₁₅ of the checksum map to b₂₀-b₂₇ respectively (see C.5.5.4).

c) *One byte of processable data at end of sequence*

Transmitted	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
1st byte	X	1	b ₁₇	b ₁₆	0	0	0	0
2nd byte	X	1	b ₁₅	b ₁₄	b ₁₃	b ₁₂	b ₁₁	b ₁₀

C.5.5.2 DDU modes

The DDUs structure depends on the DDU mode.

C.5.5.2.1 Basic kernel

Delimitation:

In mode A, B and D the D-Set mode and D-Data are delimited using a length indicator (LI).

In mode C the DDUs are delimited with an end delimiter (0/13).

In mode E the DDUs are delimited with an end delimiter (1/15).

Coding of terminal responses (D-Response-positive, D-Response negative, D-U-Abort)

In mode A only the command identifier (CI) is transmitted.

In mode B the CI is transmitted followed by 1/12.

In mode C the CI is transmitted preceded by 1/15, 3/14 (US '>') and followed by the end delimiter 0/13. The TDU's sent by the terminal are transmitted in a D-Data preceded by 1/15, 3/14 and followed by 0/13.

In mode D and E the CI of D-U-Abort is transmitted, and the string defined in D-Set-mode or D-Data parameter (or its default value) is transmitted for D-Response-positive or negative.

C.5.5.2.2 Symmetrical service

In the symmetrical service the TDUs are transmitted in D-Set-mode or in D-Data.

C.5.5.3 Coding of DDUs

C.5.5.3.1 Structure of DDUs

The general structure of DDUs is the following:

SD	CI	Sequence code	LI1	Parameter field	LI2	Data field	TRAILER
----	----	---------------	-----	-----------------	-----	------------	---------

SD Start delimiter coded 1/15, 3/14 (US '>')

This start delimiter is present in all DDUs except in DDUs sent by the terminal in mode A, B, D, E.

CI Command identifier

This field is a one byte field coded as described in C.5.5.3.2 (except in mode D, E and F for D-Response-positive or negative where it is replaced with a redefinable string).

Sequence code

This field is present in D-Set-mode and D-Data only if ED use is selected.

Its coding is given in C.5.5.3.2.2.1.

LI1 This field indicates the length of the parameter field.
It is coded as described in C.5.5.3.1.1.
In D-Set-mode this field is always present.
In D-Data this field is only present in mode D and E.
In other DDUs this field is absent.

Parameter field

This field is coded as described in C.5.5.3.1.2.
This field may only be present when LI1 is present and different from 0.

LI2 This field indicates the length of the data field, it is coded as described in C.5.5.3.1.1
This field is only present in D-Set-mode and D-Data in modes A,B, D, E, F and G.

Data field

This field may only be present in D-Set-mode and D-Data DDUs.

TRAILER

In mode A,B,D and G

This field is only present in D-Set-mode and D-Data if use of BCS has been selected.
Its coding is given in C.5.5.4.

In mode C

This field contains BCS in the same conditions of that in the previous modes.
The trailer is ended in all DDUs in mode C by an End Delimiter.
It is coded 0/13 (Carriage Return).

In mode B

This field is present in the terminal responses. It is coded 1/12 ('#').

In mode E and F

This field contains either the sequence US,3/14, 2/X followed either by BCS if BCS is in use or the next US sequence.

X contains the flag, in this mode the flag value in the CI parameter is set to "00".

The translation modes specified in C.5.5.1 apply to all bytes of D-Set-mode and D-Data except SD, End Delimiter, CI, BCS (see C.5.5.4) and Sequence code fields.

C.5.5.3.1.1 Length Indicator field (LI)

The length indicator is used to indicate the length of a field. The value of the LI field is expressed as a binary number representing the length in octets of the associated parameter field (i.e. the value of the LI field does not include itself).

LI fields indicating lengths within the range 0-254 shall comprise one byte.

LI fields indicating lengths within the range 255-65534 shall comprise three bytes. The first byte shall be coded 15/15 and the second and the third bytes shall contain the length of the associated parameter field with the high order bits in the first of these two bytes.

C.5.5.3.1.2 Parameter field

The parameter field comprises zero one or more groups of:

- *Parameter Identifier field (PI)* – A single byte identifying the parameter. The coding of PIs is specified in Table C.11.

- *Length Indicator field (LI)* – This length indicator is used to indicate the length of the following parameter value. This LI is coded as specified in C.5.5.3.1.1.
- *Parameter Value field (PV)* – The coding of the parameter values is specified in C.5.5.3.2.

The absence of a parameter in a parameter field means that the default value applies if defined.

If a PV field is absent, the whole parameter group must be absent (i.e. LI may not be equal to 0).

In order to ensure easier migration to enhanced versions of the protocol the following rules should apply:

- parameter group containing an undefined parameter identifier should be ignored in reception.
- reserved bits in a parameter value are set to zero by the sender and ignored in reception.
- the parameters may appear in any order in a parameter field. They must not be repeated.

Bits in a byte are numbered from b_0 (least significant) to b_7 (most significant).

C.5.5.3.2 DDU encoding

See Table C.10.

TABLE C.10/T.101

Coding of DDU command identifiers

DDU	CI/BCS sequence code	CI/no BCS sequence code	CI/no BCS no/sequence code
D-Set-mode	7/x	6/x	4/x
D-Data	5/x	5/x	5/x
D-U Abort	3/9	3/9	3/9
D-Response-positive	3/0	3/0	3/0
D-Response-negative	3/1	3/1	3/1

C.5.5.3.2.1 x values

bits b_1, b_0 = 0 0: translation mode 4

= 0 1: translation mode 2

= 1 0: translation mode 3

= 1 1: translation mode 1

bits b_3, b_2 = 0 0: no flag set

= 0 1: confirmation flag set

= 1 0: more flag set

= 1 1: poll flag set

In mode E and F the flags are contained in the trailer. In this case in the CI parameter the bits b_3, b_2 are set to “00”; in the trailer bits b_3, b_2 of 2/x are set to “11” and the flags are coded in b_1, b_0 according to Table C.10.

C.5.5.3.2.2 Parameters encoding

See Table C.11.

TABLE C.11/T.101

Coding of DDU parameter identifiers

Parameter	PI
Sequence code	4/0-5/15, 6/0
Define D-Response-positive	2/1
Define D-response-negative	2/2
DDU mode	2/3
Inactivity timer	2/4
Request timer	2/5
Reset	2/6
NOTE – Translation mode, flags are part of the DDU code.	

C.5.5.3.2.3 D-Set-mode

SD	= 1/15, 3/14
CI D-Set-mode	= 7/x or 6/x or 4/x
* Sequence code	= 4/0 (see C.5.5.3.2.3.1)
LI1	= 1 byte (see C.5.5.3.1.1)
PI DDU mode	= 2/3
LI	= 0/1
PV	= 1 byte (see C.5.5.3.2.3.2)
* PI Define D-Resp-pos.	= 2/1
LI	= n < 17
PV	= string (see C.5.5.3.2.3.3)
* PI Define D-Resp-neg	= 2/2
LI	= n < 17
PV	= string (see C.5.5.3.2.3.4)
* PI Inactivity timer	= 2/4
LI	= 0/1
PV	= 1 byte (see C.5.5.3.2.3.5)
* PI DDU request timer	= 2/5
LI	= 0/1
PV	= 1 byte (see C.5.5.3.2.3.6)
* LI2	= 1 or 3 bytes (see C.5.5.3.1.1)
* Data	= n bytes
* Trailer	
* BCS	= 3 bytes (see C.5.5.4)
* End delimiter	= 1 byte
or	
* End delimiter	= 1/15, 3/14, 2/x (see C.5.5.3.1)
* BCS	= 3 bytes (see C.5.5.4)

* The presence of the field is optional and subject to the conditions indicated in the referred subclause.

C.5.5.3.2.3.1 Sequence code

This field is only present if error detection is provided.

C.5.5.3.2.3.2 DDU mode parameter value

This parameter is coded on 1 byte. Bits b_3 , b_5 , b_6 , b_7 are reserved.

bits b_2 , b_1 , b_0	= 0 0 0: mode A
	= 0 0 1: mode B
	= 0 1 0: mode C
	= 0 1 1: mode D
	= 1 0 0: mode E
	= 1 0 1: mode F
	= 1 1 0: mode G
bit b_4	= 0: D-Data size is limited to 2048 bytes
	= 1: D-Data size is not limited

C.5.5.3.2.3.3 Define D-Response-positive parameter value

String of maximum length 16 bytes redefining the D-Response-positive DDU. The default value is 3/0.

NOTE – It is the responsibility of the Master to ensure that the characters of the string cause no problem in the bearer service.

This parameter is only present in mode D, E and F.

C.5.5.3.2.3.4 Define D-Response-negative parameter value

String of maximum length 16 bytes redefining the D-Response-negative DDU. The default value is 3/1.

NOTE – It is the responsibility of the Master to ensure that the characters of the string cause no problem in the bearer service.

This parameter is only present in mode D, E and F.

C.5.5.3.2.3.5 Inactivity timer parameter value

This parameter is coded on 1 byte. The time-out interval is coded in binary form in increments of 1 second. The value 0/0 (default value) has the special significance of 'timer disabled'.

C.5.5.3.2.3.6 DDU Request timer parameter value

This parameter is coded on 1 byte. The time-out interval is coded in binary form in increments of 1 second. The value 0/0 (default value) has the special significance of 'timer disabled'.

This timer is only present if ED is in use.

C.5.5.3.2.4 D-Data

	SD	= 1/15, 3/14
	CI D-Data	= 5/x
*	Sequence code	= 4/0-5/15, 6/0 (see C.5.5.3.2.4.1)
*	LI1	= 1 byte (see C.5.5.3.1.1)
*	PI Define D-Resp-pos.	= 2/1
	LI	= n < 17
	PV	= string (see C.5.5.3.2.4.2)
*	PI Define D-Resp-neg.	= 2/2
	LI	= n < 17
	PV	= string (see C.5.5.3.2.4.3)
*	PI Reset	= 2/6
	LI	= 0/1
	PV	= 1 byte (see C.5.5.3.2.4.4)

*	LI2	= 1 or 3 bytes (see C.5.5.3.1.1)
*	Data	= n bytes
*	Trailer	
*	BCS	= 3 bytes (see C.5.5.4)
*	End delimiter	= 1 byte
	or	
*	End delimiter	= 1/15, 3/14, 2/x (see C.5.5.3.1)
*	BCS	= 3 bytes (see C.5.5.4)

* The presence of the field is optional and subject to the conditions indicated in the referred subclause.

C.5.5.3.2.4.1 Sequence code coding

bits	b ₇ ,	b ₆ ,	b ₅ ,	b ₄ ,	b ₃ ,	b ₂ ,	b ₁ ,	b ₀	
	0	1	0	0	0	0	0	0	} Sequence number module 32
	0	1	0	0	0	0	0	1	
	0	1	0	—	—	—	—	—	
	0	1	0	—	—	—	—	—	
	0	1	0	1	1	1	1	1	
	0	1	1	0	0	0	0	0	6/0 reset sequence number to 0

(The sequence number of the current DDU is 0 and the next one will be coded 4/1).

This field is only present if error detection is in use.

C.5.5.3.2.4.2 Define D-response-positive parameter value

As for C.5.5.3.2.3.2 above but the default value of this parameter is “no change”.

C.5.5.3.2.4.3 Define D-response-negative parameter value

As for C.5.5.3.2.3.3 above but the default value of this parameter is “no change”.

C.5.5.3.2.4.4 Reset value

This parameter is coded on 1 byte. Bits b₂,b₃,b₄,b₅,b₆,b₇ are reserved.

bits b₁, b₀ = 0 0: no reset (default value)

= 0 1: D-Response-positive is reset to default value

= 1 0: D-Response-negative is reset to default value

= 1 1: D-Response-positive and D-Response-negative are reset to default values

C.5.5.3.2.5 D-U-Abort

CI D-U-Abort = 3/9

C.5.5.3.2.6 D-Response-positive

CI D-Response-positive = 3/0

C.5.5.3.2.7 D-Response-negative

CI D-Response-negative = 3/1

C.5.5.4 BCS coding

The BCS applies to all transmitted bytes of the D-Set mode or D-Data DDUs from the first byte following the start delimiter up to and including the last byte preceding the BCS field.

The BCS field is coded in 3 bytes using the 3-in-4 coding (see C.5.5.1.2).

The BCS is a 16 bits value calculated as follows:

The checksum is the 16-bit frame checking sequence specified in CCITT Recommendation X.25, formed by division by the polynomial:

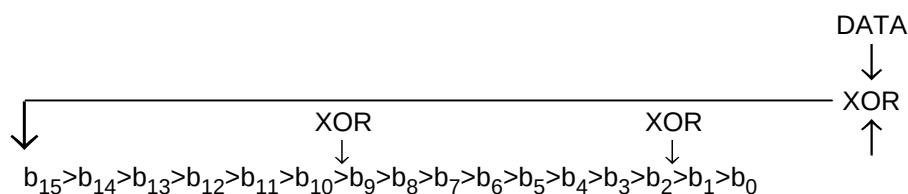
$$x^{16} + x^{12} + x^5 + 1$$

This may be calculated according to the following procedure.

The checksum, $b_{15} \dots b_0$, is initialised to all 1's. It is then modified in turn for each bit of the data. For this purpose the data is treated as 8-bit bytes, and the least significant bits in each byte are taken first; if the most significant bit of each data byte is a parity bit it is set to zero before the checksum calculation. The modification consists of:

- an exclusive-OR of b_0 with the next data bit;
- a one-bit shift in which a zero bit is brought into b_{15} of the checksum;
- an exclusive-OR of the result of a) with the new bits b_{15} , b_{10} and b_3 of the checksum.

This may be illustrated as follows:



The checksum is finally complemented (1's complement) and mapped into three bytes as described in C.5.5.1.2 and illustrated in Table C.9, b).

C.5.5.4.1 Example of CRC calculation

Figure C.5 illustrates the calculation of the CRC checksum.

Next data																		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Bit	Byte
	1 0 0 1 1 0 1	1 1 0 0 0 1 0	1 1 1 0 0 1 1	1 1 1 0 0 0 1	1 1 1 1 0 0 0	1 1 1 0 0 0 1	1 1 1 1 0 0 0	1 1 1 1 0 0 0	1 1 1 1 1 0 0	1 1 1 1 1 1 0	1 1 1 1 1 1 1	1 1 1 1 1 1 1	1 1 1 1 0 1 0	1 1 1 1 0 0 1	1 1 1 1 0 0 0	1 1 1 1 1 0 0	1 1 1 1 1 1 0	2 / 7
	0 0 1 0 1 0 1	1 0 0 1 1 0 0	0 1 0 0 1 0 1	1 0 1 0 1 0 1	1 1 0 0 1 0 0	0 1 0 0 1 0 0	1 0 1 0 1 0 0	0 1 0 0 0 1 1	0 0 1 0 0 0 0	0 0 0 1 0 0 0	1 0 0 1 0 1 0	1 1 0 0 1 0 1	1 1 0 0 1 1 1	0 1 1 0 1 1 1	1 0 1 0 0 1 1	0 1 0 1 0 0 1	0 0 0 1 0 1 0	4 / 0
	1 1 1 1 0 1 0	1 1 1 1 1 0 1	0 1 1 1 1 1 1	1 0 1 1 1 1 0	1 1 0 1 1 1 1	1 0 0 1 0 0 1	0 1 0 0 1 0 0	0 0 0 1 1 0 0	1 0 0 1 1 0 1	0 1 0 0 0 1 1	0 0 0 0 0 0 0	0 0 1 0 0 1 0	0 1 1 1 1 1 1	1 0 1 1 1 1 1	1 1 0 1 1 1 1	1 1 0 1 1 1 1	0 0 0 1 1 1 0	4 / 0
	1 0 0 0 1 1 1	0 1 0 0 0 1 1	1 0 0 0 0 0 1	1 1 0 0 0 0 0	0 1 1 0 0 0 0	0 0 1 0 0 1 1	1 0 1 0 0 0 1	0 1 0 1 0 0 0	0 0 1 1 0 1 0	1 0 0 0 1 1 1	1 1 0 0 0 0 1	0 1 0 0 1 0 0	1 1 1 0 1 0 1	1 0 1 1 0 1 0	1 1 0 1 1 0 1	1 1 1 0 1 0 0	1 1 1 1 0 0 0	1 / 1 5
	0 1 1 0 1 1 0	1 0 1 0 1 1 1	1 1 0 1 1 0 1	1 1 0 1 1 0 1	0 1 1 0 1 0 0	0 1 0 0 1 0 1	1 0 1 0 0 1 0	1 1 0 0 1 0 1	0 1 1 0 1 0 0	0 0 1 0 1 0 1	1 0 0 1 0 1 0	1 1 0 1 1 0 1	0 0 0 1 0 0 0	1 0 0 0 1 0 0	0 0 0 0 1 0 0	1 0 0 0 0 1 1	0 1 1 1 0 0 0	3 / 1 4
	1 0 0 0 1 0 0	0 1 0 0 0 1 0	1 0 0 0 0 0 1	1 1 0 0 0 0 0	1 1 1 0 0 0 0	1 1 1 1 0 0 0	1 1 1 1 0 0 0	0 1 1 1 0 0 0	1 0 1 1 0 1 0	0 1 1 1 1 1 1	1 0 1 1 1 1 1	0 1 1 1 1 1 1	0 0 1 1 1 1 1	0 0 1 1 1 1 1	0 0 0 1 1 1 1	0 0 0 0 1 0 1	0 0 0 1 0 0 1	3 / 0
	1 0 0 0 0 0 0	0 1 0 0 0 1 0	0 1 0 0 0 1 1	1 0 1 0 0 0 0	0 1 0 0 0 0 0	1 0 1 0 0 0 0	0 1 0 0 0 0 0	0 1 0 0 0 0 0	0 1 0 0 1 0 0	0 0 0 0 1 1 1	1 0 0 1 1 1 1	1 0 1 1 1 1 1	0 1 1 1 1 1 1	1 0 1 1 1 1 1	1 0 1 1 1 1 1	1 0 1 0 1 1 1	Final value Complement	

These 16-bits map for transmission into:

1st byte: X 1 1 1 0 1 0 0

2nd byte: X 1 0 0 1 0 0 0

3rd byte: X 1 1 0 1 0 1 1

Appendix I

(to Part 9 of Annex C to Recommendation T.101)

A compression algorithm

```
/******
Izhuf.c
written by Haruyasu Yoshizaki 11/20/1988
some minor changes 4/6/1989
comments translated by Haruhiko Okumura 4/7/1989
modified for Btx-FIF by Info TeSys GmH 1991-11-04
: use malloc() instead of huge static areas
: files have to be opened outside of this module
: FCS-check included
: length field "long(Filesize)" has been removed from file
*****/
#include <stdio.h>
#include <io.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
typedef unsigned char BYTE;
#define ushort unsigned short
#define EXIT_FAILURE 0
#define EXIT_SUCCESS 1
static int infile, outfile;
static long codesize;
static long InpSize, OutSize;
static char wterr[] = "Can't write.";
extern unsigned long FCS_32(BYTE CH, unsigned long crc_reg); static void near Error(char *message)
{
    printf("\r\n%s\n", message);
    exit(EXIT_FAILURE);
}
/****** LZSS compression *****/
#define N 4096 /* buffer size */
#define F 60 /* lookahead buffer size */
#define THRESHOLD 2
#define NIL N /* leaf of tree */
#define text_buf_len (N + F - 1)
static BYTE *text_buf;
static int near match_position;
static int near match_length;
static short *lson; /* [N + 1]; */
static short *rson; /* [N + 257]; */
static short *dad; /* [N + 1]; */
#define N_CHAR (256 - THRESHOLD + F)
/* kinds of characters (character code = 0..N_CHAR-1) */ #define T
(N_CHAR * 2 - 1) /* size of table */
#define R (T - 1) /* position of root */
#define MAX_FREQ 0x8000 /* updates tree when the */
/* root frequency comes to this value. */ struct lzh_env
{
    BYTE InpBuf[2048];
    short InpPos;
    short InpMax;
```

```

    BYTE   OutBuf[1024];
    short   OutPos;
    short   OutMax;
    unsigned freq[T + 1]; /* frequency table */
    int      prnt[T + N_CHAR];
                /* pointers to parent nodes, except for the */
/* elements [T..T + N_CHAR - 1] which are used to get */ /* the positions of leaves corresponding to the codes. */
    int      son[T]; /* pointers to child nodes (son[], son[] + 1) */
};
static struct lzh_env *LZHENV;
static int near InpByt(void)
{
    if (LZHENV->InpPos >= LZHENV->InpMax)
    {
LZHENV->InpMax = read(infile, LZHENV->InpBuf, LZHENV->InpMax);
        if (LZHENV->InpMax == 0) return(EOF);
        LZHENV->InpPos = 0;
        InpSize += LZHENV->InpMax;
        printf("\r  %6ld\r", InpSize);
    }
    return(LZHENV->InpBuf[LZHENV->InpPos++]);
} /* InpByt */
static int near OutByt(short BY)
{
    if (LZHENV->OutPos == LZHENV->OutMax)
    {
write(outfile, LZHENV->OutBuf, LZHENV->OutPos); OutSize += LZHENV->OutPos; printf("\r\t\t\t\t\t %6ld\r",
OutSize); LZHENV->OutPos = 0;
    }
    LZHENV->OutBuf[LZHENV->OutPos++] = (BYTE) BY;
    return(1);
} /* OutByt */
static void near InitTree(void) /* initialize trees */ {
    register int i;
    for (i = N + 1; i <= N + 256; i++)
        rson[i] = NIL; /* root */
    for (i = 0; i < N; i++)
        dad[i] = NIL; /* node */
} /* InitTree */
static void near InsertNode(register int r) /* insert to tree */ {
    register int p;
    int i, cmp;
    unsigned char *key;
    unsigned c;
    cmp = 1;
    key = &text_buf[r];
    p = N + 1 + key[0];
    rson[r] = lson[r] = NIL;
    match_length = 0;
    for ( ; )
    {
        if (cmp >= 0)
        {
            if (rson[p] != NIL)
                p = rson[p];
            else
            {
                rson[p] = r;

```

```

        dad[r] = p;
        return;
    }
}
else
{
    if (lson[p] != NIL)
        p = lson[p];
    else
    {
        lson[p] = r;
        dad[r] = p;
        return;
    }
}
for (i = 1; i < F; i++)
    if ((cmp = key[i] - text_buf[p + i]) != 0) break;
if (i > THRESHOLD)
{
    if (i > match_length)
    {
        match_position = ((r - p) & (N - 1)) - 1;
        if ((match_length = i) >= F)
            break;
    }
    if (i == match_length)
    {
        if ((c = ((r - p) & (N - 1)) - 1) < match_position) {
            match_position = c;
        }
    }
}

dad[r] = i = dad[p];
lson[r] = lson[p];
rson[r] = rson[p];
dad[lson[p]] = r;
dad[rson[p]] = r;
if (rson[i] == p)
    rson[i] = r;
else
    lson[i] = r;
dad[p] = NIL; /* remove p */
} /* InsertNode */
static void near DeleteNode(register int p) /* remove from tree */ {
    register int q;
    if (dad[p] == NIL)
        return; /* not registered */ if (rson[p] == NIL)
        q = lson[p];
    else
    {
        if (lson[p] == NIL)
            q = rson[p];
        else
        {
            q = lson[p];

```



```

        {
if (! OutByt(putbuf))          /* LOBYTE putbuf */ {
        Error(wterr);
        }
        codesize += 2;
        j      -= 8;
        putbuf  = c << (1 - j);
        }
        else
        {
                putbuf <= 8;
                codesize++;
        }
        }
        putlen = j;
}      /* Putcode */
/* initialization of tree */
static void near StartHuff(void)
{
        register int i, j;
lson  = calloc(1, (N + 1) * sizeof(short)); rson  = calloc(1, (N + 257) * sizeof(short));
dad   = calloc(1, (N + 1) * sizeof(short)); text_buf = calloc(1, text_buf_len); memset(text_buf, ' ', (N - F));
LZHENV = calloc(1, sizeof(struct lzh_env));
LZHENV->InpMax = LZHENV->InpPos = sizeof(LZHENV->InpBuf);
        LZHENV->OutMax = sizeof(LZHENV->OutBuf);
getbuf = getlen = putbuf = putlen = 0; match_position = match_length = 0; codesize = InpSize = OutSize = 0;
        for (i = 0; i < N_CHAR; i++)
        {
LZHENV->freq[i]  = 1; LZHENV->son[i]    = i + T; LZHENV->prnt[i + T] = i;
        }
        i = 0;
        j = N_CHAR;
        while (j <= R)
        {
LZHENV->freq[j] = LZHENV->freq[i] + LZHENV->freq[i + 1]; LZHENV->son[j] = i;
        LZHENV->prnt[i] = LZHENV->prnt[i + 1] = j;
        i += 2;
        j++;
        }
        LZHENV->freq[T] = 0xffff;
        LZHENV->prnt[R] = 0;
}      /* StartHuff */
/* reconstruction of tree */
static void near reconst(void)
{
        register int i, k, j;
        unsigned f, l;
/* collect leaf nodes in the first half of the table */
/* and replace the freq by (freq + 1) / 2. */
        for (i = k = 0; i < T; ++i)
        {
                if (LZHENV->son[i] >= T)
                {
LZHENV->freq[k] = (LZHENV->freq[i] + 1) / 2; LZHENV->son[k] = LZHENV->son[i];
                        k++;
                }
        }
}
/* begin constructing tree by connecting sons */

```

```

    for (i = 0, j = N_CHAR; j < T; i += 2, j++)
    {
        k = i + 1;
        f = LZHENV->freq[j] = LZHENV->freq[i] + LZHENV->freq[k]; for (k = j - 1; f < LZHENV->freq[k]; k--);
        k++;
        l = (j - k) * 2;
        memmove(&LZHENV->freq[k + 1], &LZHENV->freq[k], l);
        LZHENV->freq[k] = f;
        memmove(&LZHENV->son[k + 1], &LZHENV->son[k], l);
        LZHENV->son[k] = i;
    }
    /* connect prnt */
    for (i = 0; i < T; i++)
    {
        if ((k = LZHENV->son[i]) >= T)
        {
            LZHENV->prnt[k] = i;
        }
        else
        {
            LZHENV->prnt[k] = LZHENV->prnt[k + 1] = i;
        }
    }
} /* reconst */
/* increment frequency of given code by one, and update tree */ static void near update(register int c)
{
    register int l;
    int i, j, k;
    if (LZHENV->freq[R] == MAX_FREQ)
    {
        reconst();
    }
    c = LZHENV->prnt[c + T];
    do
    {
        k = ++LZHENV->freq[c];
        /* if the order is disturbed, exchange nodes */
        if (k > LZHENV->freq[l = c + 1])
        {
            while (k > LZHENV->freq[++l]);
            l--;
            LZHENV->freq[c] = LZHENV->freq[l];
            LZHENV->freq[l] = k;
            i = LZHENV->son[c];
            LZHENV->prnt[i] = l;
            if (i < T) LZHENV->prnt[i + 1] = l;
            j = LZHENV->son[l]; LZHENV->son[l] = i;
            LZHENV->prnt[j] = c;
            if (j < T) LZHENV->prnt[j + 1] = c;
            LZHENV->son[c] = j;
            c = l;
        }
    } while ((c = LZHENV->prnt[c]) != 0); /* repeat up to root */
} /* update */
//static unsigned code, len;
static void near EncodeChar(unsigned c)

```

```

{
    register unsigned i;
    register int k, j;
    i = j = 0;
    k = LZHENV->prnt[c + T];

    /* travel from leaf to root */
    do
    {
        i >>= 1;
/* if node's address is odd-numbered, choose bigger brother node */ if (k & 1) i += 0x8000;
        j++;
    } while ((k = LZHENV->prnt[k]) != R);
    Putcode(j, i);
// code = i;
// len = j;
    update(c);
} /* EncodeChar */
static void near EncodePosition(register unsigned c)
{
    register unsigned i;
    /* output upper 6 bits by table lookup */
    i = c >> 6;
    Putcode(p_len[i], (unsigned) p_code[i] << 8);
    /* output lower 6 bits verbatim */
    Putcode(6, (c & 0x3f) << 10);
} /* EncodePosition */
static void near EncodeEnd(void)
{
    if (putlen)
    {
        if (! OutByt(putbuf >> 8)) /* HIBYTE putbuf */ {
            Error(wtterr);
        }
        codesize++;
    }
} /* EncodeEnd */
static int near DecodeChar(void)
{
    register unsigned c;
    c = LZHENV->son[R];
    /* travel from root to leaf, */
/* choosing the smaller child node (son[]) if the read bit is 0, */
/* the bigger (son[]+1) if 1 */
    while (c < T)
    {
        c += GetBit();
        c = LZHENV->son[c];
    }
    c -= T;
    update(c);
    return(c);
} /* DecodeChar */
static int near DecodePosition(void)
{
    register unsigned i, j, c;
/* recover upper 6 bits from table */
    i = GetByte();
    c = ((unsigned) d_code[i] << 6);
    j = d_len[i];

```

```

/* read lower 6 bits verbatim */
    j -= 2;
    while (j--)
    {
        i = (i << 1) + GetBit();
    }
    return(c | (i & 0x3f));
} /* DecodePosition */
/* compression */
void LZH_Encode(int fi, int fo, unsigned long *CRC_32) /* compression */ {
    register int r, s;
    int i, c, len, last_match_length;
    long textsize;
    unsigned long INPCRC = 0xffffffff;
    infile = fi;
    outfile = fo;
    textsize = filelength(infile);
    StartHuff();
    InitTree();
    r = N - F;
    for (s = 0; s < F && (c = InpByt()) != EOF; ++s)
    {
        INPCRC = FCS_32((BYTE) c, INPCRC); text_buf[r + s] = (BYTE) c;
    }
    len = s;
    for (s = 1; s <= F; ++s) InsertNode(r - s);
    InsertNode(r);
    s = 0;
    do
    {
        if (match_length > len)
            match_length = len;
        if (match_length <= THRESHOLD)
        {
            match_length = 1;
            EncodeChar(text_buf[r]);
        }
        else
        {
            EncodeChar(255 - THRESHOLD + match_length); EncodePosition(match_position);
        }
        last_match_length = match_length;
    } while (i < last_match_length && (c = InpByt()) != EOF; ++i) {
        INPCRC = FCS_32((BYTE) c, INPCRC);
        DeleteNode(s);
        text_buf[s] = (BYTE) c;
        if (s < F - 1) text_buf[s + N] = (BYTE) c;
        s = (s + 1) & (N - 1);
        r = (r + 1) & (N - 1);
        InsertNode(r);
    }
    while (i++ < last_match_length)
    {
        DeleteNode(s);
        s = (s + 1) & (N - 1);
        r = (r + 1) & (N - 1);
        if (--len) InsertNode(r);
    }
    } while (len > 0);
    EncodeEnd();
}

```

```

        if (LZHENV->OutPos)
        {
write(outfile, LZHENV->OutBuf, LZHENV->OutPos); OutSize += LZHENV->OutPos;
        }
printf("In : %6ld bytes\n", InpSize); printf("Out: %6ld bytes\n", OutSize);
// printf("Out/In: %.3f\n", (double) InpSize / OutSize);
        free(LZHENV);
        free(text_buf);
        free(dad);
        free(lson);
        free(rson);
        *CRC_32 = INPCRC;
} /* Encode */

/* recover */ void LZH_Decode(int fi, int fo,
long filesize, unsigned long *CRC_32)
{
    register int i, j, k, r, c;
    unsigned long count;
    unsigned long OUTCRC = 0xffffffff;
    infile = fi;
    outfile = fo;
    StartHuff();
    r = N - F;
    for (count = 0; count < filesize; )
    {
        if ((c = DecodeChar()) < 256)
        {
            if (! OutByt(c))
            {
                Error(wterr);
            }
            OUTCRC = FCS_32((BYTE) c, OUTCRC); text_buf[r++] = (BYTE) c;
            r &= (N - 1);
            count++;
        }
        else
        {
            i = (r - DecodePosition() - 1) & (N - 1); j = c - 255 + THRESHOLD;
            for (k = 0; k < j; ++k)
            {
                c = text_buf[(i + k) & (N - 1)]; if (! OutByt(c))
                {
                    Error(wterr);
                }
                OUTCRC = FCS_32((BYTE) c, OUTCRC); text_buf[r++] = (BYTE) c;
                r &= (N - 1);
                count++;
            }
        }
    }
    if (LZHENV->OutPos) write(outfile, LZHENV->OutBuf, LZHENV->OutPos);
    printf("\r\t\t\t\t %6ld\r", OutSize);
    free(LZHENV);
    free(text_buf);
    free(dad);
    free(lson);
    free(rson);
    *CRC_32 = OUTCRC;
} /* Decode */
/* end of FIFLZH.C */

```

Part 10 – Terminal Facility Identifier

TABLE OF CONTENTS

	<i>Page</i>
Foreword	562
Introduction	562
C.1 Scope	563
C.2 Normative references	563
C.3 Definitions	565
C.4 Abbreviations.....	565
C.5 Overview	565
C.6 Coding	566
C.6.1 Compatibility provision for ITU-T Recommendation T.101	566
C.6.2 Conformance to the SRM	566
C.6.3 Profiles and non-Latin language capabilities	567
C.6.4 Audio capabilities	569
C.6.5 Modem capability	570
C.6.6 Photographic capabilities	571
C.6.7 Terminal configuration delimiter	572
C.6.8 Non-Latin language capability	572
C.6.9 Determination of the terminal current profile and capabilities	573
C.7 Defaults.....	573
C.8 Formal description of the terminal response.....	573
C.8.1 Introduction.....	573
C.8.2 Notation used	573
C.8.3 Formal description of the terminal requests and response	574

Foreword

This part has been established by the Terminal Equipment (TE) Technical Committee of the European Telecommunications Standards Institute (ETSI).

This issue (1994) supersedes the previous version in order to take into account the introduction of

- the photographic described in ETS 300 177 which corresponds to Recommendation T.101 Annex F;
- the audio syntax described in ETS 300 149 which corresponds to Recommendation T.101 Annex E;
- the Videotex Enhanced Man Machine Interface Service (VEMMI) described in ETS 300 382 which corresponds to ITU-T Recommendation T.107

This Recommendation describes the Terminal Facility Identifier (TFI) which may be used to ascertain the capabilities of either a physical Videotex terminal or another Videotex service.

Introduction

The Terminal Facility Identifier (TFI) may be used to ascertain the capabilities of a “terminal” (where a terminal may actually be a physical terminal or another Videotex service).

Videotex terminals have not all been designed to support all coding techniques; for example, photographic and audio, or support of different types of local facilities such as telesoftware or various types of modems. In addition, a specific terminal may support one or more of the defined base data syntaxes DS I, DS II or DS III. The TFI may be used to determine:

- each of the national or regional Videotex services;
- the terminal profile and additional terminal capabilities;
- to which parts of the Service Reference Model (SRM) the terminal conforms;
- the current profile and capabilities of the terminal.

C.1 Scope

This Recommendation specifies the data syntax to be used by Videotex services for terminal capabilities identification.

This Recommendation is applicable to both the Videotex service and to Videotex terminals. Those terminals may communicate with the Videotex service via the Public Switched Telephone Network (PSTN) or the Integrated Services Digital Network (ISDN). Connection of terminals to other networks is for further study.

It also applies to any equipment (e.g. within the system of another Videotex service) which acts as a Videotex terminal.

C.2 Normative references

- CCITT Recommendation F.300 (1988), *Videotex service*.
- CCITT Recommendation T.100 (1988), *International information exchange for interactive videotex*.
- ITU-T Recommendation T.107, *Videotex Enhanced Man Machine Interface service (VEMMI)*
- ETS 300 072 (1990), *Terminal Equipment (TE); Videotex presentation layer protocol, Videotex presentation layer data syntax*.

NOTE – Annex A of this ETS is equivalent to Recommendation T.101, Annex C, Part 1, 3, 4, 5, 6, 8 and 11. Annex B and Annex C, of this ETS are equivalent to Recommendation T.101, Annex C, Appendix I and Appendix II respectively.

- ETS 300 073 (1990), *Videotex presentation layer protocol; Geometric Display (CEPT Recommendation T/TE 06-02, Edinburgh 1988)*.

NOTE – This ETS is equivalent to Recommendation T.101, Annex C, Part 2.

- ETS 300 074 (1990), *Videotex presentation layer protocol; Transparent data (CEPT Recommendation T/TE 06-03, Edinburgh 1988)*.

NOTE – This ETS is equivalent to Recommendation T.101, Annex C, Part 7.

- ETS 300 075 (1990), *Terminal Equipment (TE); Videotex processable data*.

NOTE – This ETS is equivalent to Recommendation T.101, Annex C, Part 9 (also called “Telesoftware”).

- ETS 300 177, *Terminal Equipment (TE); Videotex, Photographic syntax*.
- ETS 300 149, *Terminal Equipment (TE); Videotex, Audio syntax*.
- CCITT Recommendation G.711 (1988), *Pulse Code Modulation of voice frequencies*.
- CCITT Recommendation G.721 (1988), *32 kbit/s adaptive differential pulse code modulation (ADPCM)*.
- CCITT Recommendation G.722 (1988), *7 kHz audio-coding within 64 kbit/s*.

- CCITT Recommendation G.723 (1988), *Extensions of Recommendation G.721 adaptive differential pulse code modulation to 20 and 40 kbit/s for digital circuit multiplication equipment application.*
- CCITT Recommendation J.41 (1988), *Characteristics of equipment for the coding of analogue high quality sound programme signals for transmission on 384 kbit/s channels.*
- CCITT Recommendation J.42 (1988), *Characteristics of equipment for the coding of analogue medium quality sound programme signals for transmission on 384-kbit/s channels.*
- I-ETS 300 036, *European digital cellular telecommunications system (phase 1); Full-rate speech transcoding (GSM 06.10).*
- CCITT Recommendation V.17 (1990), *Recommendation for a 2-wire modem for facsimile applications with rates up to 14 400 bit/s.*
- CCITT Recommendation V.21 (1988), *300 bits per second duplex modem standardised for use in the general switched telephone network.*
- CCITT Recommendation V.22 (1988), *1200 bits per second duplex modem standardized for use in the general switched telephone network and on point-to-point 2-wire leased telephone-type circuits.*
- CCITT Recommendation V.22 bis (1988), *2400 bits per second duplex modem using the frequency division technique standardized for use on the general switched telephone network and on point-to-point 2-wire leased telephone-type circuits.*
- CCITT Recommendation V.23 (1988), *600/1200 bits per second modem standardized for use in the general switched telephone network.*
- CCITT Recommendation V.26 bis (1988), *2400/1200 bits per second modem standardized for use in the general switched telephone network.*
- CCITT Recommendation V.26 ter (1988), *2400 bits per second duplex modem using the echo cancellation technique standardized for use on the general switched telephone network and on point-to-point 2-wire leased telephone-type circuits.*
- CCITT Recommendation V.27 ter (1988,) *4800/2400 bits per second modem standardized for use in the general switched telephone network.*
- CCITT Recommendation V.29 (1988), *9600 bits per second modem standardised for use on point-to-point 4-wire leased telephone-type circuits.*
- CCITT Recommendation V.32 (1988), *A family of 2-wire, duplex modems operating at data signalling rates of up to 9600 bit/s for use on the general switched telephone network and on leased telephone-type circuits.*
- CCITT Recommendation V.32 bis (1991), *A duplex modem operating at data signalling rates of up to 14 400 bit/s for use on the general switched telephone network and on leased point to point 2-wire telephone type circuits.*
- ITU-T Recommendation V.34 (1994), *A modem operating at data signalling rates up to 28 800 bit/s for use on the general switched telephone network and on leased point-to-point 2-wire telephone-type circuits.*
- CCITT Recommendation H.221 (1988), *Frame structure of a 64 kbit/s channel in audio-visual teleservices.*
- CCITT Recommendation V.42 (1988), *Error correcting procedures for DCEs using asynchronous to synchronous conversion.*
- CCITT Recommendation V.42 bis (1990), *Data compression procedures for data circuit-terminating equipment (DCE) using error correcting procedures.*
- ISO 11172-3, *Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s – Audio part.*

C.3 Definitions

capability; facility: Coding method described by the Videotex data syntax and supported by a given Videotex terminal.

data syntax: Videotex coding technique as described in CCITT Recommendation T.101.

facility: See “Capability”.

profile: Consistent subset of the Service Reference Model.

service Reference Model: List of functionalities a terminal should comply with.

videotex Host Computer: Refer to CCITT Recommendation F.300.

C.4 Abbreviations

DRCS	Dynamically Redefinable Character Set
DS I	Data Syntax according to ITU-T Recommendation T.101, Annex B
DS II	Data Syntax according to ITU-T Recommendation T.101, Annex C
DS III	Data Syntax according to ITU-T Recommendation T.101, Annex D
ETS	European Telecommunication Standard
ETSI	European Telecommunications Standards Institute
ISDN	Integrated Services Digital Network
PSTN	Public Switched Telephone Network
PDE	Presentation Data Element
SRM	Service Reference Model
TFI	Terminal Facility Identifier
US	Unit Separator
VPCE	Videotex Presentation Control Element
VPDE	Videotex Presentation Data Element
VEMMI	Videotex Enhanced Man Machine Interface

C.5 Overview

The Videotex data syntax allows for the use of a variety of different coding techniques (text, mosaics, graphics, audio, photographic pictures, etc.).

The Videotex terminals have not all been designed to support all the possible coding techniques; some of them are unable to display graphic or photographic images, to output audio information. Some others partly support such facilities. Some facilities are grouped together in order to establish a so called “Videotex profile” which represents a coherent and consistent set of functionalities and, therefore, is intended to be supported by a large population of terminals.

In addition, terminals may be connected to different networks, use different modems, etc.

To maximize the chance of successful communication, the Videotex host computer may ask the terminal about the capabilities it supports.

The terminal may answer indicating either:

- the data syntax and the profile(s); or
- precise list of facilities it supports.

The coding is described in clause C.6 below and the formal description of the terminal response is given in clause C.8.

C.6 Coding

To request the terminal profile, the code US 2/0 4/0 is transmitted to the terminal.

The terminal shall reply with US 2/0 followed by a series of bytes from columns 4 to 7 of the code table and terminating with the code 4/0 except in some cases when the Escape code (7/15) is used (see C.6.3).

Host		Terminal
... Identification Command:	US 2/0 4/0	----->
<-----	US 2/0 x/y...x/y Z	TFI response...

Intermediate codes x/y shall be transmitted by the terminal between US 2/0 and the final code Z. The value of the final code Z is described in C.6.3. Intermediate codes identify the capabilities of the terminal and shall be selected from the following list:

4/1-4/15	=	reserved for internal use of each data syntaxes DS I, DS II and DS III (DS II is defined by this annex)
5/0, 5/1	=	defined in C.6.4
5/2	=	modem capability (as defined in C.6.5)
5/3	=	reserved
5/4	=	data syntax and profile (rank/facility) identification:
3/1	=	DS I
3/2	=	DS II
3/3	=	DS III
3/4 to 3/13	=	reserved
3/14	=	private
3/15	=	terminator code
5/5	=	photographic mode (as defined in C.6.6)
5/6	=	support of ISO 9281 switching
5/7	=	support of VEMMI data
5/8-5/15	=	reserved
6/0-7/15	=	reserved for internal use of each data syntax DS I, DS II and DS III.

The values in the range 4/1-4/15 and 6/0-7/15 terminated by the code 3/15 are used for the profile identification. The codes following a particular data syntax identification pertain to that particular data syntax.

C.6.1 Compatibility provision for ITU-T Recommendation T.101

Terminal Facility Identifier (TFI) implementations of terminals which are intended to be used in systems where more than one data syntax is in use, should be based on ITU-T Recommendation T.101. In that case, the profile identification of data syntax II (4/1-4/15 and 6/0-7/15) are included between the codes 5/4, 3/2 and 3/15. The code 5/6 indicates the support of ISO 9281 switching. The code 5/7 indicates the support of VEMMI. The end of the TFI response is indicated by the value 4/0 only and not by the last capability byte (see C.6.3). The detailed description is contained in ITU-T Recommendation T.101, clause C.11

C.6.2 Conformance to the SRM

NOTE – This subclause is present within this annex only to maintain compatibility with existing systems.

The following codes have been assigned for identifying parts of the SRM:

4/0	Terminator
4/1	Alphamosaic
4/2	Geometric
4/3	Photographic
4/4	Define DRCS

4/5	Define colour
4/6	Define format
4/7	Transparent data
4/8	Reset
4/9	Processable data
4/11	Timing control

Example:

A terminal conforming to the following parts of the SRM: Alphamosaic, Define DRCS, Define colour, Define format, Reset shall transmit:

US 2/0, 4/1, 4/4, 4/5, 4/6, 4/8, 4/0

If different levels of conformance are defined in the future (as for the Geometric display or Processable data), then the level of conformance may be indicated by one or more bytes from column 3 of the code table following the relevant byte from column 4.

For the Geometric mode (primary byte 4/2) the following sub-levels shall be defined:

3/0-3/7	reserved
3/8-3/15	the sub-levels to be defined are for further study.

Example:

A terminal which conforms to the following parts of the SRM: Alphamosaic, Geometric (sub-level coded 3/9), Define Dynamically Redefinable Character Set (DRCS), Define colour, Define format and Reset shall transmit:

US 2/0 4/1 4/2 3/9 4/4 4/5 4/6 4/8 4/0

The sub-levels of processable data are for further study.

C.6.3 Profiles and non-Latin language capabilities

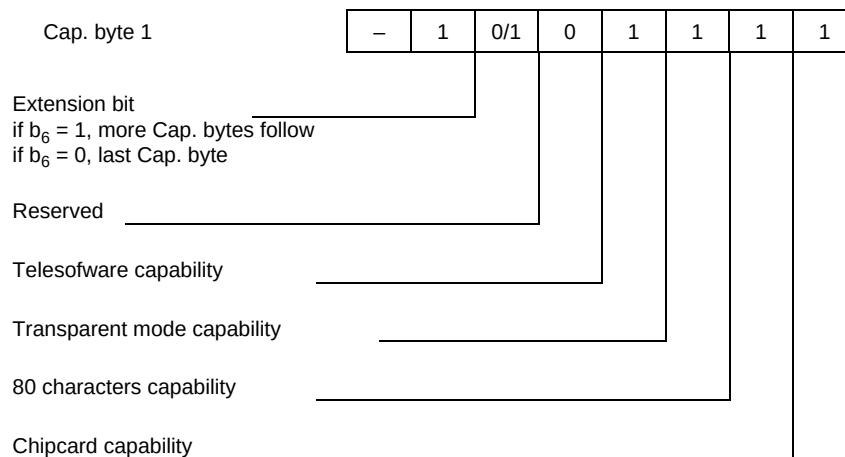
The following codes for terminal profiles have been assigned:

6/0	Alphamosaic profile 1 of this Annex which corresponds to ETS 300 072
6/1	Alphamosaic profile 2 of this Annex which corresponds to ETS 300 072
6/2	Alphamosaic profile 3 of this Annex which corresponds to ETS 300 072
6/3	Alphamosaic profile 4 of this Annex which corresponds to ETS 300 072
6/4	Alphamosaic Chinese profile 5 of this Annex
6/5	Reserved for Alphamosaic profiles
6/6	Non-final capability byte indicator
6/7	Terminal configuration delimiter
6/8	Geometric profile x1 of this Annex which corresponds to ETS 300 073
6/9	Geometric profile x2 of this Annex which corresponds to ETS 300 073
6/10-6/15	Reserved for future Geometric profiles
7/0	Any photographic profile
7/1	Reserved for CEPT-DPCM image coding profile (ETS 300 072 Part 3, Appendix A)
7/2	Reserved for CEPT-ADCT image coding profile (ETS 300 072 Part 3, Appendix B)
7/3	Greek capability
7/4	Arabic capability
7/5	Reserved (Chinese capability)

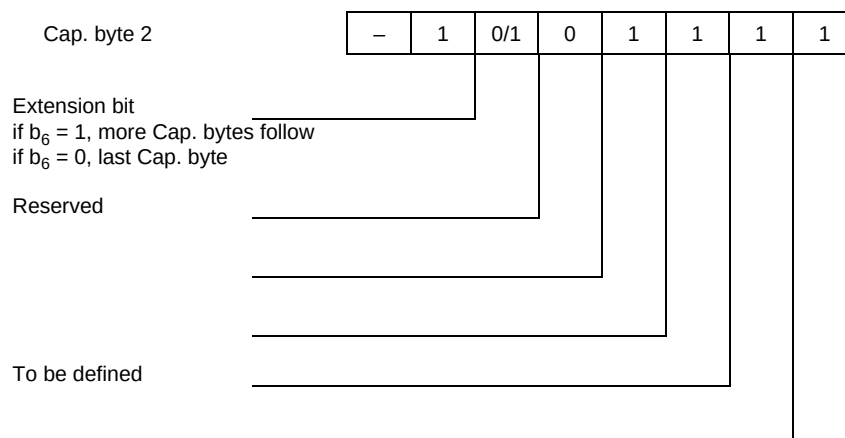
7/6	Hebrew capability
7/7	Cyrillic capability
7/8-7/13	Reserved for future use
7/14	ASCII profile introducer

The following byte from columns 4-7 specifies the ASCII capabilities and the following ASCII capabilities have been reserved:

4/1	Reserved for VT52
4/2	Reserved for VT100-Series
4/3	Reserved for VT200-Series
4/4	Reserved for Teletype
4/5	Reserved for VT300-Series
4/6-7/13	Reserved
7/14	Allocated for private use
7/15	Escape code. The following bytes (Cap. bytes) from columns 4-7 specify additional terminal capabilities by using a bitwise identification (Cap. bits) in the following way:



The Telesoftware capability bit and the Transparent mode capability bit shall be used to identify the use of these data types associated with the profiles. Subclause C.6.2 gives codes that are used to identify SRM conformance to ETS 300 074 (Transparent data) and ETS 300 075 (Processable data).



The end of the TFI sequence shall be determined either:

- by the code 4/0; or
- by the last Cap. byte (extension bit set to 0), except when the non-final capability byte indicator is used.

The non-final capability byte indicator has the value 6/6 and shall be the very first byte of the ETSI data syntax terminal identification response.

For a terminal supporting only the ETSI data syntax, the non-final capability byte shall immediately follow the US 2/0 sequence. The non-final capability indicator indicates that the capability byte shall not be the final byte of the TFI. The end of the TFI sequence shall then be 4/0.

If a terminal supports more than one alpha-mosaic, ASCII, geometric, photographic or special profile, it may send the codes assigned to each profile it supports. If a terminal sends more than one code for the same display mode it shall be capable of handling the switching sequences defined in Annex C, clause 5 of ETS 300 072.

The first alpha-mosaic profile shall identify the preferred profile.

Examples:

a) US 2/0 6/1 4/0: alphamosaic profile 2

b) US 2/0 6/0 6/9 7/15 4/6 =

–	1	0	0	0	1	1	0
---	---	---	---	---	---	---	---

Alphamosaic profile 1 with geometric profile x2, transparent mode and 80 character capability.

NOTE – In the above examples throughout this subclause, the most significant bit is represented by the symbol “–”. It corresponds to the parity bit in a 7-bit environment and is not defined in an 8-bit environment.

C.6.4 Audio capabilities

When receiving a profile request US 2/0 4/0, the terminal supporting an audio capability shall issue, together with the other codes used for identifying profiles or parts of the SRM, one or more audio conformance code(s) followed by one or more couple(s) of octets indicating respectively the algorithm supported and the audio bitrate.

The audio conformance codes have been assigned as follows:

- 5/0 Audio capability, block mode (as described in ETS 300 149)
- 5/1 Audio capability with framing (as described in CCITT Recommendation H.221).

Algorithm identification:

- 3/0 PCM A-law described in CCITT Recommendation G.711
- 3/1 PCM μ -law described in CCITT Recommendation G.711
- 3/2 ADPCM described in CCITT Recommendations G.721/G.723
- 3/3 sub-band ADPCM described in CCITT Recommendation G.722
- 3/4 RPE-LTP coding method (I-ETS 300 036)
- 3/5 near instantaneous (CCITT Recommendation J.41)
- 3/6 sub-band ADPCM (CCITT Recommendation J.42)
- 3/7 MPEG Audio (ISO 11172-3)
- 3/8-3/14 reserved for future use
- 3/15 forbidden (CCITT terminator code for alphamosaic facility)

Bitrate audio coding:

- 3/0 8 kbit/s
- 3/1 16 kbit/s
- 3/2 24 kbit/s
- 3/3 32 kbit/s
- 3/4 40 kbit/s
- 3/5 48 kbit/s
- 3/6 56 kbit/s
- 3/7 64 kbit/s

3/8	13 kbit/s (I-ETS 300 036)
3/9	2.4 kbit/s (provisional)
3/10	4.8 kbit/s (provisional)
3/11	128 kbit/s (provisional)
3/12	192 kbit/s (provisional)
3/13	384 kbit/s (provisional)
3/14	256 kbit/s (provisional)
3/15	forbidden (CCITT terminator code for alphamosaic facility).

Example:

When receiving a profile request, a terminal which conforms to the alpha-mosaic, geometric and photographic parts of the SRM and which supports the audio capability in block mode using PCM A-law 64 kbit/s or ADPCM 32 kbit/s shall transmit:

US 2/0 4/1 4/2 4/3 5/0 3/0 3/7 3/2 3/3 4/0

C.6.5 Modem capability

When receiving a profile request US 2/0 4/0, the terminal supporting the modem capability shall issue together with the other codes used for identifying profiles or parts of the SRM, or an audio capability, the modem conformance code followed by one or more byte(s) in order to indicate the type of modems supported by the terminal.

The modem conformance code has been assigned as follows:

5/2 modem capability

Each byte following the modem conformance code shall take one of the following values:

3/0	not applicable or unknown capability (default)
3/1	no modem used
3/2	modem running in asynchronous mode, according to the following specification:
4/0	reserved code (ETSI termination code)
4/1	unknown speed (default)
4/2	modem according to CCITT Recommendation V.21
4/3	modem according to CCITT Recommendation V.22
4/4	modem according to CCITT Recommendation V.22 <i>bis</i>
4/5	modem according to CCITT Recommendation V.23
4/6	modem according to CCITT Recommendation V.32
4/7	modem according to CCITT Recommendation V.32 <i>bis</i>
4/8	modem according to CCITT Recommendation V.26 <i>ter</i>
4/9	modem according to ITU-T Recommendation V.34
4/10-4/13	reserved codes
4/14	reserved code for non-CCITT modem
4/15	reserved code
3/3	modem running in synchronous mode, according to the following specification:
4/0	reserved (terminator code)
4/1	unknown speed (default)
4/2	modem according to CCITT Recommendation V.26 <i>bis</i>
4/3	modem according to CCITT Recommendation V.26 <i>ter</i>
4/4	modem according to CCITT Recommendation V.27 <i>ter</i>
4/5	modem using the modulation technique described in CCITT Recommendation V.29
4/6	modem according to CCITT Recommendation V.32
4/7	modem according to CCITT Recommendation V.32 <i>bis</i>

4/8	modem according to CCITT Recommendation V.17
4/9	modem according to CCITT Recommendation V.22
4/10	modem according to CCITT Recommendation V.22 <i>bis</i>
4/11	modem according to ITU-T Recommendation V.34
4/12-4/13	reserved codes
4/14	reserved code for non-CCITT modem
4/15	reserved code
3/4	modem using an embedded error correction or compression mechanism, whose type is (are) identified by the following byte(s):
4/0	reserved (terminator code)
4/1	unknown mechanism
4/2	correction mechanism as described by CCITT Recommendation V.42
4/3	compression mechanism as described by CCITT Recommendation V.42 <i>bis</i>
4/4-4/15	reserved codes.

The compression or correction mechanism only applies to the immediate preceding defined modem type. In this case, the asynchronous or synchronous type of any additional modem shall be repeated.

Example:

When receiving a profile request, a terminal which conforms to alphamosaic profile 3 and may use either a V.23 modem, or a V.32/V.42 modem, or a V.27 *ter* modem shall transmit:

US 2/0 6/2 5/2 3/2 4/5 3/3 4/6 3/4 4/2 3/3 4/4 4/0

3/5-3/14	reserved
3/15	forbidden (CCITT terminator code for alphamosaic facility).

C.6.6 Photographic capabilities

When receiving a profile request US 2/0 4/0, the terminal supporting a photographic capability shall issue, together with the other codes, the photographic conformance code.

The photographic conformance code has been assigned as follows:

5/5	photographic conformance code.
-----	--------------------------------

This conformance code may be followed by one or more codes identifying the profile(s) in use:

3/0	reserved
3/1	compatible photographic profile P1 (CIF, sequential)
3/2	compatible photographic profile P2 (P1 + 2-1-1 sequential)
3/3	compatible photographic profile P3 (P2 + 4-2-2 sequential)
3/4	compatible photographic profile P4 (P3 + spectral selection c mode)
3/5	compatible photographic profile P5 (P4 + successive approximation progressive mode)
3/6-3/13	reserved
3/14	private choice of photographic profile P _{PRIV}
3/15	forbidden (CCITT terminator code for alphamosaic facility).

For a given profile, it is possible to indicate if the terminal has a specific display capability whose type is identified by the following byte:

4/0	reserved (terminator code)
4/1	monochrome capability
4/2	T.4 decoding facility
4/3	T.6 decoding facility
4/4-4/15	reserved.

It is assumed that, in absence of any additional information, the terminal has a colour display capability. In addition, a monochrome terminal shall always be able to process received colour information and to have it displayed in monochrome.

Example:

When receiving a profile request, a terminal which conforms to alphamosaic profile 4 together with photographic P1 (colour and T.4 decoding facility) and photographic P2 (monochrome and T.6 decoding facility) shall transmit:

US 2/0 6/3 5/5 3/1 4/2 3/2 4/1 4/3 4/0

C.6.7 Terminal configuration delimiter

A given Videotex terminal may consist of several Videotex logical terminals.

Each logical terminal may have several characteristics: profiles, modem, etc.

The terminal configuration delimiter is used to delimit logical terminals within one single physical terminal.

When receiving a profile request US 2/0 4/0, the terminal may transmit a list of supported logical terminals. In this list, the logical terminals shall be delimited with the character 6/7.

Example:

(In this example, the character “/” is used to represent the terminal configurations delimiter):

- alphamosaic profile 4: 6/3;
- alphageometric profile 1, x2/alphageometric profile 2, x1: 6/0 6/9 6/7 6/1 6/8;
- alphamosaic profile 4 / VT100, photographic profile P1: 6/3 6/7 7/14 4/2 5/5 3/2.

NOTE – The Videotex service may switch the terminal into a specific configuration by issuing the mechanism from ETS 300 072, Annex C, clause 5.

C.6.8 Non-Latin language capability

Several codes in the range 7/3-7/7 are reserved to express the non-Latin language capability.

In order to indicate that a given alphamosaic profile is to be used with a non-Latin language, the alphamosaic profile code shall be immediately followed by one single non-Latin language capability code.

Example 1:

A terminal supporting alphamosaic profile 1, alphamosaic profile 3 for both Greek and Chinese and a chip card capability shall transmit:

US 2/0 6/0 6/2 7/3 6/2 7/5 7/15 4/1

Example 2:

A terminal supporting two configurations: a first one consisting of alphamosaic profile 4 and Greek profile 2; a second configuration consisting of a Chinese profile 2 with a chip card capability shall transmit:

US 2/0 6/3 6/1 7/3 6/7 6/1 7/5 7/15 4/1

Example 3:

A terminal supporting three configurations:

- configuration 1: alphamosaic profile 1 with chip card;
- configuration 2: alphamosaic Greek profile 2 with telesoftware;
- configuration 3: alphamosaic profile 4;

shall transmit:

US 2/0 6/6 6/0 7/15 4/1 6/7 6/1 7/3 7/15 4/8 6/7 6/3 4/0

C.6.9 Determination of the terminal current profile and capabilities

In the context of a multistandard Videotex network, it could be useful to determine the terminal profile and capabilities currently in use.

To request the terminal current profile and capabilities, the sequence US 2/0 4/15 4/0 is transmitted to the terminal.

Host	Terminal
... Current profile Command:	US 2/0 4/15 4/0 - - - - - >
< - - - -	US 2/0 x/y...x/y Z; TFI response...

The terminal shall reply with the sequence US 2/0 followed by a series of bytes (x/y) from the columns 4 to 7 of the code table and terminated by Z which is the code 4/0, except in some cases when the escape code (7/15) is used (see C.6.3). Intermediate codes identify the terminal current profile and capabilities. In this case the terminal configuration delimiter (6/7) shall never be used.

Example:

A terminal which supports the three following configurations:

- Configuration 1 – Alphamosaic profile 1 with chip card capability;
- Configuration 2 – Alphamosaic profile 4 and alphamosaic profile 2 with Greek capability;
- Configuration 3 – Alphamosaic profile 3.

shall respond

for configuration 1 in use: US 2/0 6/0 7/15 4/1

for configuration 2 and profile 4 in use: US 2/0 6/3 4/0

for configuration 2 and profile 2 with Greek capability in use: US 2/0 6/1 7/3 4/0

for configuration 3 in use: US 2/0 6/2 4/0.

C.7 Defaults

If no response is received from a terminal within a specified time (depending upon the transmission network being used), then the basic terminal used within that network is assumed.

C.8 Formal description of the terminal response

C.8.1 Introduction

This formal grammar is a description of the TFI response.

C.8.2 Notation used

<symbol>	non-terminal
<a/b>	terminal: a, b = 0..15
<symbol>*	0 or more occurrences
<symbol>+	1 or more occurrences
<symbol>O	optional (0 or 1 occurrences)
<symbol-1> := <symbol-2>	symbol-1 has the syntax of symbol-2
<symbol-1> <symbol-2>	symbol-1 or alternative symbol-2
[comment]	explanation

C.8.3 Formal description of the terminal requests and response

TFI-global-request	::=	1/15 2/0 4/0
TFI-current-prof-request	::=	1/15 2/0 4/15 4/0
TFI-response	::=	1/15 2/0 <response>
response	::=	<response-et> <response-ci>
response-et	::=	<response-66> <old-response>
response-66	::=	6/6 <response-n> 4/0
response-n	::=	<config-1> <config-n>*
[In case of a response to a TFI-current-prof-request, "response-n" shall be restricted to:		
response-n	::=	<config-1>]
config-1	::=	<response-1> <cap-bytes>O
response-l	::=	<ds-II-prof> <com-prof>O
config-n	::=	6/7 <config-1>

old-response	::=	<response-1> <config-p>*,<caps-bits-or-end>
[In case of a response to a TFI-current-prof-request, "old-response" shall be restricted to:		
old-response	::=	<response-1><caps-bits-or-end>]
config-p	::=	6/7 <response-1>
response-ci	::=	5/4 <ds-list> <com-prof>O <end of resp>
ds-II-prof	::=	<srm>*<etsi-prof>O
etsi-prof	::=	<alphan-prof>+<geom-prof>*<cept-photo-prof>*<ascii-prof>O
[In case of a response to a TFI-current-prof-request, only one occurrence of <alphan-prof> shall appear and only one occurrence of <geom-prof> and/or only one occurrence of <cept-photo-prof> may appear]		
com-prof	::=	<audio-prof>O <photo-prof>O <9281-switch>O <modem-prof>O <VEMMI>O
audio-prof	::=	<audio-intro> <audio-arg-and-rate>
audio-intro	::=	5/0 5/1
audio-arg-and-rate	::=	3/x 3/y [x=0..7, y=0..14]
photo-prof	::=	5/5 <photo-id>+
photo-id	::=	3/x <monochrome>O [x=0..14] <T.4decoding>O <T.6decoding>O
monochrome	::=	4/1
T.4decoding	::=	4/2
T.6decoding	::=	4/3
9281-switch	::=	5/6
VEMMI	::=	5/7
cap-bits-or-end	::=	<cap-bytes> <end of resp>
ds-list	::=	<ds-I-list>O <ds-II-list> <ds-III-list>O
ds-I-list	::=	3/1 <ds-I-prof> 3/15 [for the description of <ds-I-prof>, refer to CCITT Recommendation T.101, Annex B]
ds-II-list	::=	3/2<response-n> 3/15
ds-III-list	::=	3/3 <ds-III-prof> 3/15 [for the description of <ds-III-prof>, refer to CCITT Recommendation T.101, Annex D]
srm	::=	4/x [x=1...11]
end-of-resp	::=	4/0
alphan-prof	::=	6/x <Non-Latin-language>O [x=0..4]
Non-Latin-language	::=	7/x [x=3,7]
geom-prof	::=	6/8 6/9
cept-photo-prof	::=	7/0 7/1 7/2
ascii-prof	::=	7/14 <ascii-prof-id>+
[In case of a response to a TFI-current-prof-request, only one occurrence of <ascii-prof-id> shall appear]		
ascii-prof-id	::=	4/x [x=1..5]
cap-bytes	::=	7/15 <cap-byte-1> <cap-byte-2>O
modem-prof	::=	5/2 <modem-type>+
[In case of a response to a TFI-current-prof-request, only one occurrence of <modem-type> shall appear]		
modem-type	::=	<default> <nil> <modem-list>
default	::=	3/0
nil	::=	3/1
modem-list	::=	<asyn> <syn>
asyn	::=	3/2 4/x <cc>O [x = 1...9]
syn	::=	3/3 4/x <cc>O [x = 1...12]
cc	::=	3/4 4/x [x = 1...3]

Part 11 – Timing Control

TABLE OF CONTENTS

	<i>Page</i>
C.1 Introduction	575
C.2 Wait	575
Appendix I – Videotex Service Reference Model: Conformance	576
Appendix II – Videotex Service Reference Model: Profiles	581

C.1 Introduction

When displaying data on a terminal, it may be desirable to introduce certain timing controls such as the instruction to wait for a specified period before continuing the display of data. These controls are coded in Control Videotex Presentation Data Elements (Control VPDE) in the following form:

US 2/4 < control identifier > < data >

The Control Identifier determines the type of control and may be followed by optional data. The following Control Identifier has been coded:

2/1 : WAIT

The remaining codes for the Control Identifier are for future standardization.

C.2 Wait

The Wait Control VPDE is coded as follows:

US 2/14 2/1 < period >

The period specifies the wait time, in tenths of a second, to be observed between the decoding of the Wait Control VPDE and the start of the execution of the VPDE following the Wait Control VPDE. Data received during a wait period is stored in the terminal after it has been scanned for the inclusion of a General Display Reset (Part 8, C.3.3) or a Service Break to Row X (Part 8, C.3.3). If a General Display Reset is detected the wait period is terminated and all the data stored in the terminal between the start of the wait and the detection of the General Display Reset is deleted. The General Display Reset is executed immediately.

If a Service Break to Row X is detected it is executed as soon as possible. The original wait will then continue. No data stored in the terminal is deleted.

The period is decimally coded, most significant digit first, by codes in the range 3/0 to 3/9 inclusive and is terminated by the code 3/11.

A Wait Control VPDE indicating a wait period of 30 seconds would be coded as follows:

US 2/14 2/1 3/3 3/0 3/0 3/11

Appendix I
(to Part 11 of Annex C of Recommendation T.101)

Videotex Service Reference Model: Conformance

To conform to Data Syntax II a selection of specific facilities which should be supported by all services have been made. The degree to which a service implements these facilities is determined by the service profile. The service profiles in use are described in Appendix II

The rules for transcoding between the recognized profiles are for further study.

The following definitions apply:

RECOGNIZE: Means to determine the syntactic form but not necessarily the semantics of a code sequence.

EXECUTE: Means to process a code sequence to allow the display of information conveyed by the code sequence and by subsequent code sequences.

PRESENT: Means to display the information conveyed by a code sequence and in the case of a control function, to display information affected by the control function.

Part 0 – general

All Videotex Presentation Control Elements (VPCEs) shall be recognized (see Part 0, C.1.1.1)

Part 1 – Alphamosaic Displays

I.1.1 Alpha-numeric characters

I.1.1.1 Simple Alpha-numeric characters

The Latin primary set of graphic characters (Part 1, Table C.8), excluding character positions 2/3, 2/4, 4/0, 5/11 to 5/14, 6/0 and 7/11 to 7/14 (which may be presented using fall-back characters) shall be recognised and presented.

Terminals with a Greek character capability shall recognise and present the Greek primary set of graphic characters (Part 1, Table C.13).

I.1.1.2 Extended Alpha-numeric characters

All Alpha-numeric characters of the repertoire (see Part 1, C.2.1.1) shall be recognized and presented.

I.1.2 Mosaic and other characters

I.1.2.1 Simple Mosaic and other characters

The block-mosaic character (see Part 1, C.2.1.2) shall be recognized and presented.

I.1.2.2 Extended Mosaic and other characters

All mosaic, line drawing and miscellaneous characters of the repertoire shall be recognized and presented.

I.1.3 Format Effectors

I.1.3.1 Simple format effectors

The following format effectors; APB, APF, APD, APR, APU, CS and APH shall be recognized and executed.

I.1.3.2 Extended format effectors

All format effectors shall be recognized and executed (see Part 1, C.2.2)

I.1.4 Attribute Controls

I.1.4.1 Parallel attribute controls

The following shall be recognized and presented:

- Foreground colours;
- Background colours;
- Start lining and stop lining;
- Normal size;
- Double height;
- Double width;
- Double size;
- Flash and steady;
- Conceal display and stop conceal;
- Inverted polarity and normal polarity;
- Start box and end box.

I.1.4.2 Serial attribute controls

The following shall be recognised and presented:

- Alpha and Mosaic foreground colours;
- New background and black background;
- Start lining and stop lining;
- Normal size and double height;
- Flash and steady;
- Conceal Display and Stop Conceal;
- Start box and end box.

I.1.4.3 Extended attribute controls

The following parallel and serial attribute controls shall be recognized and presented:

- *Flash states* – Flash, Steady, Inverted Flash and Reduced Intensity Flash
- *Flash rates* – Normal Flash, Fast Flash (three phases), Increment Flash and Decrement Flash
- Marking

NOTE – The fall back for Flash rate and Flash state is Normal Flash. The fall back for Marking is Non-Marking.

I.1.4.4 Full Row attribute controls

The following shall be recognized and presented:

- Foreground Colours;
- Background Colours;
- Lining;
- Normal Size;
- Normal Flash;

- Steady;
- Invert;
- Window;
- Conceal;
- Protecting.

NOTE – The fall back for Protecting is Non-Protecting.

I.1.4.5 Full Screen Attribute Controls

The following shall be recognized and presented:

- Background Colour

I.1.4.6 Scrolling

I.1.4.6.1 Simple Scrolling

Implicit scrolling shall be executed.

I.1.4.6.2 Extended Scrolling

The definition of one scrolling area shall be recognized and executed.

Implicit and explicit scrolling shall be recognized and executed.

I.1.5 Device controls

All device control functions shall be recognized (see Part 1, C.2.4).

Cursor On and Cursor Off shall be executed.

I.1.6 Coding Structure

7-bit or 8-bit coding shall be executed.

I.1.7 Invocation of Character Sets

The invocation of character sets for the 7- or 8-bit environment as appropriate shall be executed (see Part 1, C.3.1.2 and C.3.1.3).

I.1.8 Designation of Character Sets

The designation of 4 fixed character sets shall be executed (see Part 1, C.3.4.3).

Note that no designation sequence is required for the set as this is fixed, as long as only Latin based characters are used.

I.1.8.1 For terminals with a Greek character capability, designation of the Greek primary set shall be executed (see Part 1, C.3.4.4).

I.1.8.2 For terminals with an Arabic character capability, designation of the Arabic primary set shall be executed (see Part 1, C.3.4.5).

I.1.8.3 For terminals with a Hebrew character capability, designation of the Hebrew secondary set shall be executed (see Part 1, C.3.4.6).

I.1.8.4 For terminals with a Chinese character capability, designation of the Chinese multibyte primary set shall be executed (see Part 1, C.3.4.7).

I.1.9 Colour Table Controls

The invocation of four colour tables shall be recognized and executed (see Part 1, C.3.5.6).

Part 2 – Geometric displays

Conformance to the Geometric Display is defined in Part 2.

Part 3 – Photographic displays

I.3.1 General

All header and transfer units (see Part 3, C.2, C.3 C.4 and C.5) shall be recognized.

For further study.

Part 4 – Define DRCS

I.4.1 Designation and Identification

The designation of one DRCS set shall be executed (see Part 4, C.5).

NOTE – The library identification of the DRCS set is given in the down loading sequence.

The default Identify Character Set (ICS) (see Part 4, C.2.1) shall be executed.

I.4.2 Character Matrices

I.4.2.1 Preferred Character Matrices

Select Dot Composition (SDC) type 2 (see Part 4, C.2.2.2) shall be executed for the following character matrices:

12×10 6×10 6×5

12×12 6×12 6×6

I.4.2.2 8 dot type Matrices

8×10 4×10 4×5

The 8 dot type matrix set may be implemented as an alternative but where this is done the means shall be provided within the system for translating from and to systems having character matrices of a 12×10 type and its derivatives.

I.4.3 Bits Per Dot

I.4.3.1 Basic DRCS

Basic DRCS shall be executed and presented (see Part 4, C.2.2.2)

I.4.3.2 Colour DRCS

Colour DRCS shall be executed and presented with 4 and 16 colours per character (see Part 4, C.2.2.2), but 4 colour DRCS is acceptable for a transitional period.

I.4.4 Coding Method

The direct coding method shall be executed (see Part 4, C.2.3).

I.4.5 Addressing Capability

One DRCS set of 94 characters of basic DRCS shall be executed and presented.

NOTE – The presentation of the characters will be dependant upon the capabilities of the display device.

Part 5 – Define COLOUR

I.5.1 Structure of the Colour Map

I.5.1.1 Size

The colour map shall consist of 32 colours (see Part 5, C.2).

I.5.1.2 CLUTs

Four fixed CLUTs (each of 8 colours) shall be executed and presented.

I.5.1.3 DCLUTs

Two DCLUTs shall be executed and presented:

- one for 4 colour DRCS;
- one for 16 colour DRCS.

I.5.2 Definition

The RGB loading method shall be executed (see Part 5, C.3.3.2).

I.5.2.1 Colour Map

The definition of colours 16 to 31 of a single colour map (ICT 2/0 2/0) shall be executed (see Part 5, C.3.1).

I.5.2.2 DCLUTs

The definition of a single DCLUT for use with 4 colour DRCS (ICT 2/2 2/0) shall be executed (see Part 5, C.3.1).

I.5.2.3 Resolution

A Select Unit Resolution (SUR) of 4 for the colour map shall be executed (see Part 5, C.3.1).

A Select Unit Resolution (SUR) of 5 for the DCLUT for use with 4 colour DRCS shall be executed (see Part 5, C.3.1).

I.5.2.4 Reset

Part 6 – Define FORMAT

I.6.1 Coding

All codings of Define FORMAT shall be recognized.

I.6.2 Format and Wrap-around

I.6.2.1 Simple

A format of 24 rows of 40 columns shall be presented with automatic wrap-around.

I.6.2.2 Extended

A format of 20 rows (US 2/13 4/2) of 40 columns shall be presented (see Part 6, C.2).

The wraparound ON and OFF commands (see Part 6, C.2) shall be executed.

Part 7 – Transparent Data

Not Applicable.

Part 8 – Reset

All reset sequences shall be recognized (see Part 8).

All those which affect the implemented display modes shall be executed.

Part 9 – Processable Data

Conformance to Processable Data is defined in Part 9.

Part 10 – Terminal Facility Identifier

Conformance to the Terminal Facility Identifier is defined in Part 10.

Part 11 – Timing Control

All timing control shall be executed (see Part 11).

Appendix II

(to Part 11 of Annex C of Recommendation T.101)

Videotex Service Reference Model: Profiles

Recognized Profiles

The following profiles are recognized for existing Videotex services using Data Syntax II.

II.1 Profile 1

This is the profile including the first horizon (see Figure 1 of the Recommendation). It uses 7-bit coding and satisfies the following parts of the conformance requirements described in Appendix I:

Part 0	General
Part 1	Alphamosaic Displays (8-bit coding)
Part 4	Define DRCS (except C.4.2.2)
Part 5	Define Colour
Part 6	Define Format
Part 8	Reset
Part 10	Terminal Facility Identifier

II.2 Profile 2

This is the profile of a service which implements only parallel attribute controls and extended format effectors.

It uses 7-bit coding and satisfies the following parts of the conformance requirements described in Appendix I:

Part 1	Alphamosaic displays
--------	----------------------

The following paragraphs only:

I.1.1.1	Simple alpha-numeric characters
---------	---------------------------------

I.1.1.2 Extended alpha-numeric characters

The alpha-numeric repertoire should be recognized. Wherever a particular graphic character cannot be presented exactly, a fallback presentation shall be shown instead. The fallback presentation should, by preference, be an approximation of the intended graphic.

I.1.2.1 Simple mosaic and other characters

I.1.3 Format effectors

I.1.4.1 Parallel attribute controls

NOTES

- 1 Background colour controls, start lining and stop lining require a space when used with alpha-numeric characters.
- 2 Start Box and End Box require a space.
- 3 Double Height and Double Size controls cannot be used on adjacent rows.

I.1.4.6.1 Implicit scrolling of the whole screen

I.1.5 Device controls

I.1.6 Coding structure -7-bit coding

I.1.7 Invocation of character sets

I.1.8 Designation of non-Latin Character Sets (for terminals with corresponding capability)

I.1.8.1 Designation of a Greek Character set (for terminals with the Greek Characters capability)

I.1.8.2 Designation of an Arabic Character set (for terminals with the Arabic Characters capability)

The terminals with the Arabic Characters capability may be in one of the following states:

- Latin state with the Latin primary set as G0, the Latin secondary state as G2 and the second mosaic supplementary set as G1;
- Arabic state with the Arabic primary set as G0 and the second mosaic supplementary set as G1;
- Arabic – Latin state with the Arabic primary set as G0, the Latin primary set as G1, the Latin secondary set as G2 and the second mosaic supplementary set as G3;
- Latin – Arabic state with the Latin primary set as G0, the Arabic primary set as G1, the Latin secondary set as G2 and the second mosaic supplementary set as G3.

I.1.8.3 Designation of an Hebrew Character set (for terminals with the Hebrew Characters capability)

I.1.8.4 Designation of a Chinese multibyte Character set (for terminals with the Chinese Characters capability)

Part 6 Format and wraparound

The following paragraph only:

I.6.2.1 Simple format and wrap-around

II.3 Profile 3

This is the profile of a service which implements only serial attribute controls and simple format effectors.

It uses 7-bit coding and satisfies the following parts of the conformance requirements described in Appendix I:

Part 1 Alphamosaic displays

The following paragraphs only:

I.1.1.1 Simple alpha-numeric characters

I.1.2.1 Simple mosaic and other characters

- I.1.3.1 Simple format effectors
- I.1.4.2 Serial attribute controls (excluding Stop Conceal)
 - NOTE – Attribute controls usually require a space.
- I.1.5 Device controls
- I.1.6 Coding structure – 7-bit coding
- I.1.7 Invocation of character sets
- I.1.8 Designation of Greek Character Sets (for terminals with a Greek character capability)
- Part 6 Define Format
- The following paragraph only:
- I.6.2.1 Simple format and wrap-around

II.4 Profile 4

This is the profile of a service which implements only serial attribute controls and extended format effectors.

It uses 7-bit coding and satisfies the following part of the conformance requirements described in Appendix I:

- Part 1 Alphamosaic displays
- The following paragraphs only:
- I.1.1.1 Simple alpha-numeric characters
- I.1.2.1 Simple mosaic and other characters
- I.1.3 Format effectors
- I.1.4.2 Serial attribute controls (excluding Stop Conceal)
 - NOTE – Attribute controls usually require a space.
- I.1.5 Device controls
- I.1.6 Coding structure – 7-bit coding
- I.1.7 Invocation of character sets
- I.1.8 Designation of Greek Character Sets (for terminals with a Greek character capability)
- Part 6 Define Format
- The following paragraph only:
- I.6.2.1 Simple Format and wrap-around

II.5 Profile 5

This is the profile of a service which implements only parallel attribute controls and extended format effector controls

It uses 7-bit coding and satisfies the following part of the conformance requirements described in Appendix 1:

- Part 0 Recognizes DRCS header only
- Part 1 Alpha mosaic display
- The following paragraphs only:
- I.1.1.1 Primary set of graphic characters
 - The general currency symbol \$ is replaced by the Chinese currency sign ¥.
- I.1.1.2 Extended alpha numeric characters
- I.1.1.3 Chinese coded ideogram set for communication
- I.1.2.1 Mosaic set
 - Only the second supplementary set of mosaic in Annex C applies.

- I.1.3 Format effectors
 - I.1.4.1 Parallel attribute controls
 - I.1.4.4 Full row attributes
 - The following shall be recognised and presented:
 - Foreground colour
 - Background colour
 - normal flash
 - lining
 - invert polarity
 - steady
 - conceal
 - box
 - I.1.4.5 Full screen attributes
 - The following shall be recognised and presented:
 - background colour
 - I.1.4.6.1 Implicit scrolling of the whole screen
 - I.1.5 Device controls
 - Only the cursor control is used.
 - I.1.6 Coding structure – 7-bit coding
 - I.1.8 Designation of the Chinese Videotex Character set (for designation of the character set, see C.3.4.7 of T.101 Annex C, Part 1)
 - I.4.1 Designation and identification
 - I.4.2 Character matrices
 - I.4.2.1 Selectable character matrices
 - I.4.3.1 Basic DRCS
 - I.4.4 Coding: direct coding is used
 - I.4.5 Addressing capability
 - Rach DRCS set can contain 94 addressed characters.
- Part 6 Define Format
- The following paragraph only:
- I.6.2.1 Simple format and wraparound

II.6 Switching between Videotex Profiles

Switching to a given profile is made effective when receiving the sequence ESC 2/5 4/4 <profile> 4/0, where <profile> is a parameter indicating which profile is to be taken into account.

<profile> identifier:

- either a single profile within a display mode; or
- a combination of profiles, one and only one per display mode.

In the alphamosaic display mode, the following values are used:

- 6/0 alphamosaic profile 1
- 6/1 alphamosaic profile 2
- 6/2 alphamosaic profile 3
- 6/3 alphamosaic profile 4
- 6/4 alphamosaic profile 5