

ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

HANDLING OF BI-DIRECTIONAL TEXTS

ECMA TR/53

December 1990

ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

HANDLING OF BI-DIRECTIONAL TEXTS

ECMA TR/53

Free copies of this document are available from ECMA,
European Computer Manufacturers Association
114 Rue du Rhône - CH-1204 Geneva (Switzerland)

Phone: +41 22 735 36 34 Fax: +41 22 786 52 31

December 1990

BRIEF HISTORY

In April 1988, ECMA TC1 decided to produce a Technical Report on specific control functions for texts involving two scripts written in opposite directions.

The requirement for such functions resulted from ECMA activities in the area of coding of bilingual character sets such as Latin/Arabic and Latin/Hebrew. The objective is to provide for the adequate presentation of such texts on character-imaging devices.

An ECMA ad hoc group was instructed to establish the required set of functions and explanations based on ISO 6429, considering in particular:

- horizontal line orientation and top-to-bottom line progression,
- new additional functions to handle bi-directional texts,
- the possible impact of bi-directionality on other control functions already defined,
- the provision of worked examples for the handling of bi-directional text, applying the means as described in this ECMA Technical Report.

This Technical Report is intended to be used as a guidance to implementors for the handling of bi-directional texts.

It is also intended to use this Technical Report as the basis for corresponding improvements of the ISO version of Standard ECMA-48, viz. ISO 6429. It has been developed so as to keep to a minimum the necessary amendments and additions to ISO 6429.

Adopted as an ECMA Technical Report by the General Assembly of December 1990.

Table of Contents

	Page
1. SCOPE	1
2. GENERAL CONSIDERATIONS	1
3. REFERENCES	2
4. REQUIREMENTS FOR HANDLING BI-DIRECTIONAL TEXTS	2
4.1 Direction of Strings	2
4.2 Ordering of Data	2
4.3 Transparency	3
5. BI-DIRECTIONAL DEVICE MODEL	3
5.1 The Character Image Output	3
5.2 The Device Structure	3
5.3 Presentation Component	4
5.4 Data Component	4
5.5 Addressing of Character Positions within Components	4
5.5.1 The data position	5
5.5.2 The active position	5
5.5.3 Relationship between data positions and active positions	5
5.6 Reference Points Movement	5
5.6.1 Implicit movement	6
5.6.2 Explicit movement	6
5.7 The Data Stream	6
5.8 Areas and Fields	6
5.8.1 Area	6
5.8.2 Field	7
5.9 Simplified Presentation of the Device Model	7
6. MODES FOR HANDLING BI-DIRECTIONALITY	8
6.1 BDSM - BI-DIRECTIONAL SUPPORT MODE	9
6.2 DCSM - DEVICE COMPONENT SELECT MODE	10
6.3 Interaction between BDSM and DCSM	10
6.4 Backward Compatibility	10
7. BI-DIRECTIONAL CONTROL FUNCTIONS	10
7.1 NEW CONTROL FUNCTIONS	10
7.1.1 CLL - Cursor Line Limit	10
7.1.2 SCP - Select Character Path	10

7.1.3	SDS - Start Directed String	11
7.1.4	SIMD - Select Implicit Movement Direction	12
7.2	MODIFIED CONTROL FUNCTIONS	12
7.2.1	SAPV - Select Alternate Presentation Variants	12
7.2.3	SRS - Start Reversed String	15
8.	LEVELS OF DEVICE SUPPORT FOR BI-DIRECTIONALITY	15
8.1	Devices with Presentation Component and Data Component	15
8.2	Devices with Presentation Component Only	16
	APPENDIX A - WORKED EXAMPLES	17

1. SCOPE

This Technical Report specifies technical means to handle bi-directional text presentation in cases of:

- texts in a single script of which specific parts need to be presented in opposite directions (e.g. numbers in Arabic or Hebrew);
- texts in different scripts written in opposite directions (like Latin/Arabic, Latin/Hebrew).

This Technical Report is restricted to:

- horizontal line orientation;
- top-to-bottom line progression.

Other associated directions may be the subject of further studies.

2. GENERAL CONSIDERATIONS

2.1 A bi-directional device model is presented to explain bi-directional text handling. As a consequence the following modifications to ISO 6429 will be necessary:

- an extended device concept;
- two new modes;
- a set of new control functions;
- where required, revised definitions of existing control functions.

The control functions are intended to be used imbedded in a stream of character-coded data for interchange with character-imaging input/output devices which are capable to handle bi-directional data.

The architecture of such devices is reflected by the bi-directional device model described in 5.

2.2 The method for the coded representation of the bi-directional control functions defined in this Technical Report is identical with that already used by other International Standards for 7-bit and 8-bit coded character sets such as ECMA-48, ISO 6429, ISO 10538.

It is therefore intended to integrate the control functions and the modes described in this Technical Report into these standards as soon as possible.

2.3 The technical capabilities of the bi-directional devices, to which this Technical Report applies, can vary according to the levels of bi-directional support which is required. Various bi-directional technical means, provided by this Technical Report, may be, therefore, selected for implementation, depending on specific applications.

3. REFERENCES

ISO 2022:1986 Information processing - ISO 7-bit and 8-bit coded character sets - Code extension techniques

ISO 6429:1988 Information processing - Control functions for 7-bit and 8-bit coded character sets

ISO/IEC 10538:1991 Information Technology - Control functions for text communication and processing

ISO 8613-6:1989 Information Processing - Text and Office Systems - Office Document Architecture (ODA) and interchange format - Part 6: Character content architecture

CCITT Rec. T.61:1988 Character repertoire and coded character sets for the international TELETEX service

4. REQUIREMENTS FOR HANDLING BI-DIRECTIONAL TEXTS

Many languages, like languages using the Latin script, are written and read from left-to-right.

Other languages, such as Arabic and Hebrew, are written and read mainly from right-to-left. Numbers, for instance, in these languages, are written and read from left-to-right. Furthermore, texts of languages with opposite presentation directions can be intermixed.

As a result, bi-directional devices should provide:

- support for both left-to-right and right-to-left presentation directions;
- support for text with embedded (nested) sections of left-to-right and right-to-left directions.

4.1 Direction of Strings

Many characters have an inherent directionality. Some have no inherent directionality and abide by context. Examples of such characters are: space, punctuation marks, separators, parentheses and so on.

In order to fully specify the direction of a string constituted of characters with and without inherent directionality, control functions are needed.

Another requirement is to support some presentation variants which depend on the direction of the string: italicized characters, for instance, are right-slanting for left-to-right strings, and left-slanting for right-to-left strings.

4.2 Ordering of Data

The order in which a bi-directional text is exchanged may differ from the order in which it is presented on the image output. For example, "hello" may be presented as "olleh" on a right-to-left device.

Such cases, and all their consequences must be handled as part of bi-directional support.

4.3 Transparency

Applications that are designed to handle bi-directionality can fully control the functionality of a bi-directional device.

There is a need, however, to allow applications not designed to handle bi-directionality to function reasonably well in a bi-directional environment, making the latter transparent to the application.

5. BI-DIRECTIONAL DEVICE MODEL

In order to explain the requirements for handling bi-directional texts, a device model is defined, which is an extension of the device concept defined in ISO 6429.

The concepts introduced below are represented schematically in figures 1 to 3.

A character-imaging device according to this model is a device capable of receiving a data stream that consists of control functions and graphic characters, and is capable of producing character image output. Such output must be readable by a human being according to the various traditional writing conventions such as left-to-right, right-to-left, top-to-bottom and bottom-to-top. The character image output is, in general, produced in the form of one or more rectangular arrays of character positions and lines which are called pages.

If the device is an input/output device rather than merely an output device, it is also capable of transmitting a data stream consisting of control functions and graphic characters; the transmitted data stream is, in general, composed of a combination of data which have been sent to the device and data which have been entered locally into the device, for example by means of an associated keyboard.

5.1 The Character Image Output

The character image output is regarded as being produced in the form of a continuous stream, but, eventually, it may be made available character-by-character, line-by-line, or page-by-page.

A page is composed of lines, each composed of character positions. The size of a character position may be fixed or may depend on the character being imaged.

5.2 The Device Structure

The device described in this model consists either of a Presentation and a Data Component or of a Presentation Component only.

The Presentation Component is used for rendering the character image output. Such a component may, for example, be a display or a printer.

The Data Component is used to store the received information for further processing.

A device with both Data and Presentation Components uses a process to transform the information stored in the Data Component for image output by the Presentation Component. In this Technical Report such a process is referred to as the Presentation process.

5.3 Presentation Component

The Presentation Component is capable of presenting the information in successive lines; each line consisting of successive character positions. The lines, as well as the character positions, are identified by the consecutive numbers 1, 2, 3 ...

Line orientation is either horizontal or vertical. This defines the way in which a line will appear on the device. In this Technical Report only horizontal line orientation is considered.

For horizontal line orientation the direction of line progression can be:

- from top-to-bottom, or
- from bottom-to-top.

The character path is the sequential order of the character positions along a line of the Presentation Component. For horizontal lines the character path can be:

- from left-to-right, or
- from right-to-left.

The lines are numbered according to the established line progression. The character positions are numbered according to the established character path.

5.4 Data Component

The Data Component structures the received data stream into successive lines by executing some control function, each line consists of successive character positions. The lines, as well as the character positions, are identified by the consecutive numbers 1, 2, 3 ...

The lines within the Data Component convey the organization of the data. Whilst they have no specific spacial orientation, for the sake of simplicity they are considered to be horizontal in this description. For the same reason, the line progression within the Data Component is considered to be from top-to-bottom.

The character progression is the sequential order of the character positions along a line of the Data Component.

The line progression and the character progression within the lines reflect the sequence in which data is intended to be read.

5.5 Addressing of Character Positions within Components

This device model uses reference points for both the Data and Presentation Component. Some device functions are performed relative to these reference points. Examples of such device functions are: character insertion, character erasure, reference point movements, etc.

The reference point within the Data Component is called the data position. The reference point within the Presentation Component is called the active position.

5.5.1 The data position

The data position is the character position which is to receive the next character of the data stream whether a graphic character or a control function. The data position is also the reference position against which some control functions are performed within the Data Component.

The line containing the data position is called the data line; the field containing the data position is called the data field; the area containing the data position is called the data area; the page containing the data position is called the data page.

5.5.2 The active position

The active position is the character position which is to image the graphic symbol representing the next graphic character of the received data stream or of the next control function for which a graphic representation is required. The active position is also the reference position against which certain control functions are performed.

It is a common practice to mark the active position by means of a special indicator called Cursor.

The line containing the active position is called the active line; the field containing the active position is called the active field; the area containing the active position is called the active area; the page containing the active position is called the active page.

5.5.3 Relationship between data positions and active positions

There is a relation between the active position and the data position. Some control functions act on, and affect, the data position, whilst other control functions act on, and affect, the active position. When one of these positions is updated, the other is updated accordingly.

Due to the possible differences between character progression and character path, as in some bi-directional environments, the character position coordinates of the data positions and active positions may differ.

In the situation where a Data Component is not present in a device then the characteristics of the Data Component, data position and character progression, are treated as if they are identical with the respective characteristics of the Presentation Component, active position and character path.

5.6 Reference Points Movement

As a result of the received data stream, both the data position and the active positions can be moved within their corresponding components. This movement can be either implicit or explicit.

5.6.1 Implicit movement

An implicit movement is defined with reference to the data position in the Data Component. This movement is performed after a graphic character or a control function, for which a graphic representation is required, is received.

The direction of the implicit movement may be the same as, or opposite to, the character progression. The direction of implicit movement is identical with the character progression, unless modified by an appropriate control function.

If the direction of the implicit movement is the same as the character progression, the data position is moved to the following character position of that line.

If the direction of the implicit movement was modified to be opposite to the character progression, the data position is moved to the preceding character position of that line.

5.6.2 Explicit movement

An explicit movement is defined with reference to either the data position in the Data Component, or the active position in the Presentation Component.

An explicit movement is performed when a control function is performed which causes the data position or the active position to be moved to a specified character position within the corresponding component.

5.7 The Data Stream

The data stream is considered to be a continuous stream. It may be structured in messages, records and/or blocks, but this does not affect the operation of the device as described in this model.

Text within a data stream can be viewed as consisting of character strings. Each such string may contain nested strings. Characters within a string are organized in the order in which they are intended to be read.

Each string has a directionality associated with it. The direction may be established using a control function or a higher-level protocol. If such direction is not established, it is the direction of the currently established character path.

5.8 Areas and Fields

The content of the Presentation Component may be structured into areas of fields of successive character positions.

5.8.1 Area

An area is defined as a string of successive character positions in the Presentation Component. The beginning and the end of an area are indicated by means of control functions, and may occur on different lines.

Two types of areas are defined in this device model:

- A selected area is used to indicate a string of character positions the content of which may be transmitted in form of a data stream, or be transferred to an auxiliary device. The character positions within a line of a selected area are ordered according to the character path of this line.
- A qualified area is similar in its functionality to a selected area, but can impose some additional attributes on its content, e.g. protecting the contents against erasure or manual alteration, restricting the type of characters that can be input into it. The order of character positions within a line of a qualified area can be the same as, or opposite to, the character path of this line.

5.8.2 Field

A field is a string of character positions on the Presentation Component, starting with a character position for which a character tabulation stop is set, and ending with the character position which precedes the following tabulation stop.

A field has an associated alignment defined by the type of the character tabulation stop which introduces the field. The examples of possible alignment types are: leading edge of character, centred within the line, centred on character, trailing edge of character.

The order of character positions within the field, as well as the order of tabulation stops within a line of the Presentation Component is determined by the character path of this line.

5.9 Simplified Presentation of the Device Model

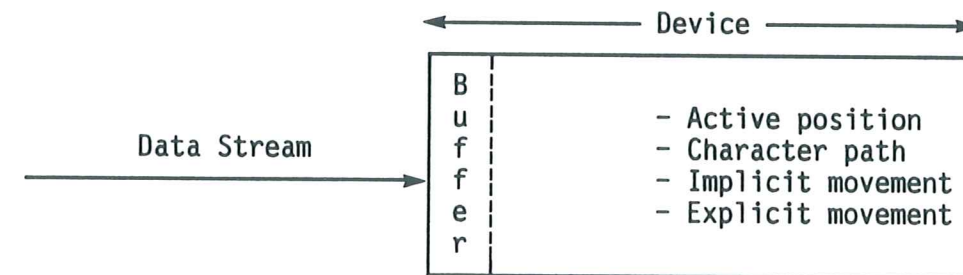


Figure 1 - ISO 6429 : Uni-directional Device with optional Buffer

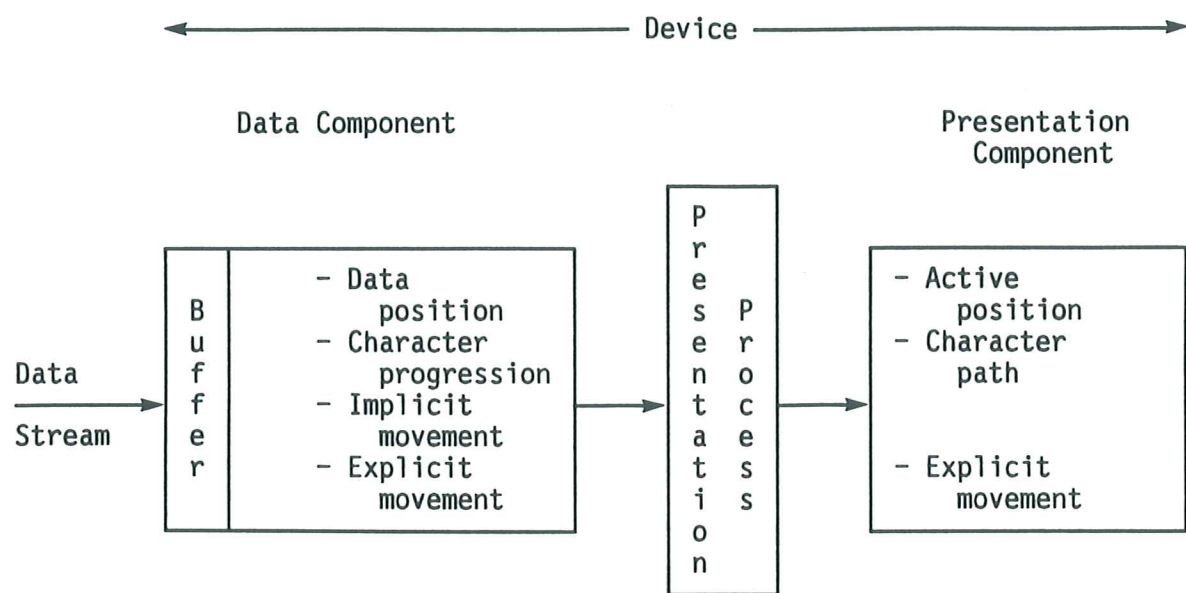


Figure 2 - Bi-Directional Device with Data Component and Presentation Component (see 8.1)

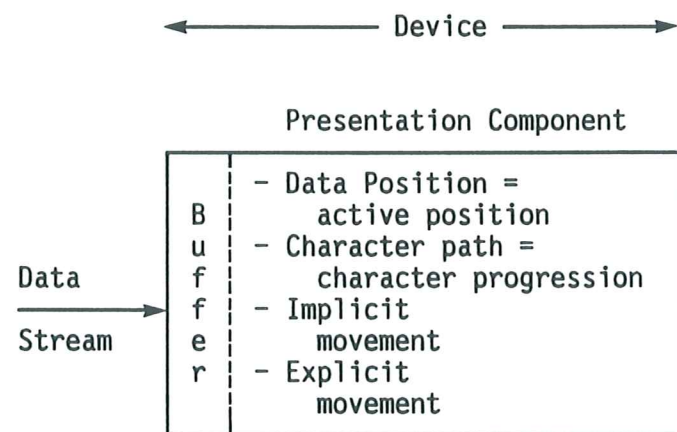


Figure 3 - Bi-Directional Device with Presentation Component only and optional Buffer (see 8.2)

6. MODES FOR HANDLING BI-DIRECTIONALITY

In a bi-directional environment there is a need to support two kinds of software application on bi-directional devices:

- applications designed to specifically handle bi-directional data (bi-directionality-aware applications);
- applications not specifically designed to handle bi-directional data (bi-directionality-unaware applications).

In the case of bi-directionality-unaware applications, the decision how to handle bi-directionality is made by the device because the data stream does not contain control functions specific to bi-directionality. These decisions are based on the following information:

- the interpretation of the data stream structure and of the actual content (semantics);
- the directionality assumed for characters (given the context).

This information cannot always be obtained in a unique and unambiguous manner from the data stream alone.

In the case of bi-directionality-aware applications, the data stream contains (through the bi-directionality-specific control functions) all information needed to achieve the desired result.

A bi-directionality-aware application must be able to fully control the device, and therefore to specify whether control functions apply to either the Data or the Presentation Component.

ISO 6429 does not contain all necessary control functions for handling bi-directionality. It can be completed in two different ways:

- by introducing a significant number of new control functions and substantial modifications of existing control functions; or
- by introducing new device modes, thus keeping to a minimum the number of new control functions and modifications needed.

This Technical Report proposes the introduction of two new modes.

The first one is the BI-DIRECTIONAL SUPPORT MODE (BDSM), which selects between the support for bi-directionality-unaware applications (IMPLICIT state) and the support for bi-directionality-aware applications (EXPLICIT state).

The second one is the DEVICE COMPONENT SELECT MODE (DCSM), which selects whether control functions are performed in the Data Component (DATA state) or in the Presentation Component (PRESENTATION state).

6.1 BDSM - BI-DIRECTIONAL SUPPORT MODE

IMPLICIT:

All control functions are performed in the Data Component. All bi-directional aspects of data are handled by the device itself.

EXPLICIT:

All control functions are performed in the Data Component or in the Presentation Component, depending on the setting of the DEVICE COMPONENT SELECT MODE (DCSM).

6.2 DCSM - DEVICE COMPONENT SELECT MODE

PRESENTATION:

Control functions are performed in the Presentation Component.

NOTE - Control functions affected are: CUB, CUD, CUF, CUP, CUU, CLL, CR, LF, NEL, RI, DCH, DL, EA, ECH, ED, EF, EL, ICH, IL and CPR.

6.3 Interaction between BDSM and DCSM

- i) If BDSM is set to EXPLICIT and DCSM is set to DATA, the control functions are performed on the Data Component.
- ii) If BDSM is set to EXPLICIT and DCSM is set to PRESENTATION, the control functions are performed on the Presentation Component.
- iii) If BDSM is set to IMPLICIT, all control functions are performed on the Data Component irrespective to the setting of DCSM; all bi-directional aspects of the data are handled by the device itself.

6.4 Backward Compatibility

For applications relying on unidirectional left-to-right behaviour of a device, BDSM should be set to EXPLICIT and DCSM to PRESENTATION.

7. BI-DIRECTIONAL CONTROL FUNCTIONS

This clause contains the definitions of the control functions which do not yet exist in ISO 6429, as well as the revised definitions for the existing bi-directional control functions. The definitions are worded in the form that they will have when introduced in ISO 6429.

7.1 NEW CONTROL FUNCTIONS

7.1.1 CLL - CURSOR LINE LIMIT

Notation: (Fs)

Representation: ESC 06/05

CLL causes the active position to be moved to the line limit position of the same line. The line limit position is established by the parameter of SET LINE LIMIT (SLL).

7.1.2 SCP - SELECT CHARACTER PATH

Notation: (Ps1;Ps2)

Representation: CSI Ps1;Ps2 02/00 06/11

No parameter default values

SCP is used to select the character path, relative to the line orientation, for the active line and subsequent lines of the Presentation Component, and as well to update the Presentation Component image for the active line and the Data Component content for the data line. This takes effect immediately.

Ps1 specifies the character path:

- 1 from left-to-right in the case of horizontal lines, or from top-to-bottom in the case of vertical lines;
- 2 from right-to-left in the case of horizontal lines, or from bottom-to-top in the case of vertical lines.

Ps2 specifies the effect on the Presentation Component image and the Data Component content:

- 0 undefined (implementation-dependent)

NOTE - This may also permit the effect to take place after the next occurrence of CR, NEL or any control function which initiates an absolute movement of the active position.

- 1 the active line image of the Presentation Component is updated to present the data line content of the Data Component according to the newly established character path; the data position is moved to the first character position in the data line
- 2 the data line content of the Data Component is updated to correspond to the active line image of the Presentation Component according to the newly established character path; the active position is moved to the first character position in the active line

7.1.3 SDS - START DIRECTED STRING

Notation: (Ps)

Representation: CSI Ps 05/13

Parameter default value: Ps = 0

SDS is used to establish a specific direction for a string of characters, the beginning and end of which are indicated by SDS. This direction may differ from that currently established. The indicated string is to be presented following the preceding text.

The beginning of the string is indicated by SDS with parameter value not equal to 0. The end of the string is indicated by SDS with the parameter value 0.

A string may contain one or more nested strings of characters bracketed by SDS or by START REVERSED STRING (SRS).

After the end of the string, the direction is re-established to that which was in effect prior to the indicated string, and the data position is moved to the character position following the characters presented.

This International Standard does not define the location of the data position within any such string.

The parameter values are:

- 0 end of the string; re-establish direction

- 1 start of the string; direction left to right
- 2 start of the string; direction right to left

NOTES

- 1. A string enclosed by SDS may contain one or more nested strings of characters enclosed by START REVERSED STRING (SRS).
- 2. The effect of receiving a CVT, HT, SCP, SPD or VT control function within an SDS string is not defined by this International Standard.
- 3. The area definition control functions (DAQ, EPA, ESA, SPA, SSA) should not be used within an SDS string.

7.1.4 SIMD - SELECT IMPLICIT MOVEMENT DIRECTION

Notation: (Ps)

Representation: CSI Ps 05/14

Parameter default value: Ps = 0

SIMD is used to select the direction of implicit movement of the data position relative to the character progression. The direction selected remains in effect until the next occurrence of SIMD.

The parameter values are:

- 0 the direction of implicit movement is the same as that of the character progression
- 1 the direction of implicit movement is opposite to that of the character progression

7.2 MODIFIED CONTROL FUNCTIONS

The modifications to the definitions of existing control functions are shown in italics.

7.2.1 SAPV - SELECT ALTERNATE PRESENTATION VARIANTS

Notation: (Ps...)

Representation: CSI Ps... 02/00 05/13

Parameter default value: Ps = 0

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 *the* Latin script
- 2 the decimal digits are presented by means of the graphic symbols used in *the* Arabic script, i.e. the Hindi symbols
- 3 *when the string direction is right-to-left*, each of the graphic characters in the graphic character set(s) in use which is one of a 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

4 *when the string direction is right-to-left*, all graphic characters which represent operators and delimiters in mathematical formulae and which are not symmetrical about a vertical axis are presented as mirrored about that vertical axis

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 medial form

8 the following graphic character is presented in its final 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

17 *when the string direction is right-to-left, the italicized characters are slanted to the left; when the string direction is left-to-right, the italicized characters are slanted to the right.*

7.2.2 SPD - SELECT PRESENTATION DIRECTIONS

Notation: (Ps1;Ps2)

Representation: CSI Ps1;Ps2 02/00 05/03

Parameter default values: Ps1 = 0; Ps2 = 0

SPD is used to select the line orientation, the line progression and the character path, and as well to update the Presentation Component image and Data Component content. This takes effect immediately.

Ps1 specifies line orientation, line progression and character path:

0	Line orientation:	Horizontal
	Line progression:	Top-to-bottom
	Character path:	Left-to-right
1	Line orientation:	Vertical
	Line progression:	Right-to-left
	Character path:	Top-to-bottom
2	Line orientation:	Vertical
	Line progression:	Left-to-right
	Character path:	Top-to-bottom
3	Line orientation:	Horizontal
	Line progression:	Top-to-bottom
	character path:	Right-to-left
4	Line orientation:	Vertical
	Line progression:	Left-to-right
	Character path:	Bottom-to-top
5	Line orientation:	horizontal
	Line progression:	Bottom-to-top
	Character path:	Right-to-left
6	Line orientation:	Horizontal
	Line progression:	Bottom-to-top
	Character path:	Left-to-right
7	Line orientation:	Vertical
	Line progression:	Right-to-left
	Character path:	Bottom-to-top

Ps2 specifies the effect on the Presentation Component image and the Data Component content:

- 0 undefined (implementation-dependent)
NOTE - This may also permit the effect to take place after the next occurrence of CR, FF, or any control function which initiates an absolute movement of the active position
- 1 the Presentation Component image is updated to present the Data Component content according to the newly established characteristics of the Presentation Component; the data position is moved to the first character position in the first line of the Data Component
- 2 the Data Component content is updated to correspond to the Presentation Component image according to the newly established characteristics of the Presentation Component; the active position is moved to the first character position in the first line of the Presentation Component

7.2.3 SRS - START REVERSED STRING

Notation: (Ps)

Representation: CSI Ps 05/11

Parameter default value: Ps = 0

SRS is used to establish a direction for a string of characters, the beginning and end of which are indicated by SRS. This direction is opposite to that currently established. The indicated string is to be presented following the preceding text.

The beginning of the string is indicated by SRS with a parameter value of 1. The end of the string is indicated by SRS with a parameter value of 0.

A string may contain one or more nested strings of characters bracketed by SRS or by START DIRECTED STRING (SDS).

After the end of the string, the direction is re-established to that which was in effect prior to the indicated string, and the data position is moved to the character position following the characters presented.

This International Standard does not define the location of the data position within any such string

The parameter values are:

- 0 end of the string; character path is re-established
- 1 start of the string; character path is reversed.

NOTES

1. A string enclosed by SRS may contain one or more nested strings of characters enclosed by START DIRECTED STRING (SDS).
2. The effect of receiving an CVT, HT, SCP, SPD or VT control function within an SRS string is not defined by this International Standard
3. The area definition control functions (DAQ, EPA, ESA, SPA, SSA) should not be used within an SRS string.

8. LEVELS OF DEVICE SUPPORT FOR BI-DIRECTIONALITY

Due to the diversity of application and device requirements, different levels of bi-directionality support may be considered. This Technical Report suggests the following two:

- devices with both Presentation Component and Data Component;
- devices with Presentation Component only.

8.1 Devices with Presentation Component and Data Component

This level addresses presentation devices, allowing the support of either bi-directionality-aware or bi-directionality-unaware applications.

A significant number of existing bi-directional devices implement only implicit bi-directionality support. This is to take advantage of bi-

directionality-unaware applications, and to provide a limited bi-directionality support.

8.2 Devices with Presentation Component Only

This level addresses presentation devices with limited functionality and high dependency on application software.

The following functions are supported with some possible limitations:

- CURSOR LINE LIMIT (CLL)
- SET PRESENTATION DIRECTION (SPD);
- SET IMPLICIT MOVEMENT DIRECTION (SIMD);
- SELECT ALTERNATE PRESENTATION VARIANTS (SAPV).

The setting of the bi-directional modes is not supported at this level. The implementation may define the default state of the modes.

APPENDIX A

WORKED EXAMPLES

This Appendix provides examples of the use of control functions and modes defined in this Technical Report. These examples cover:

- the combined use of control functions in bi-directional texts,
- the nesting of strings,
- the definition of fields,
- the two bi-directional modes.

The device used in the following examples has both Data Component and Presentation Component.

Each worked example comprises the following sections:

Description

This section gives the general information related to the example.

Data stream

This section shows the data stream required to produce the output image. Each graphic character or control function is shown separated by space. Each control function is shown by its acronym and parameter values. Control functions are shown in bold text. The graphic symbol representing the character SPACE is a low line shown as: .

The data stream is divided into successive segments. This is designed to help understand how the complete image is created.

The graphic characters in the data stream are a, b, c, This is intended to help understanding the effect of bi-directional control functions.

Data Component

This section shows the data as stored in the Data Component prior to presentation on the Presentation Component. Character positions within each line of the Data Component are numbered for ease of reference.

In order to avoid dependency on a particular implementation, only the graphic characters are shown. Any implementation-dependent details (such as character attributes, nesting level, line attributes etc.) are not shown.

For demonstration purposes, only a defined portion of Data Component (8 lines of 60 character positions each) is shown.

Presentation Component

This section shows the output that should occur as the result of the presentation of the data stream.

The numbering of character positions within the Presentation Component is not shown as the numbering scheme changes according to control functions used.

BI-DIRECTIONAL TEXT - WORKED EXAMPLE 1

Description:

This example demonstrates:

- the explicit bi-directionality support,
- the left-to-right presentation direction defined by SPD,
- the embedded strings bracketed by SRS,
- the right-to-left presentation direction defined by SCP,
- the embedded strings bracketed by SDS,
- the use of SAPV in conjunction with various kinds of parentheses.

Data stream:

BDSM(EXPLICIT) SPD(0;1) SAPV(3)

CUP(2;7) SLH(7) SLL(54) a b c d __ SRS(1) e f g h SRS(0) __ i j NEL k l __ SRS(1) m n o p __ SRS(1) q r s t SRS(0) __ u v w SRS(0) __ x y z NEL (a b c SRS(1) { d e f } SRS(0) g h i } NEL

SCP(2;1) CHA(7) SLH(7) SLL(54) a b c d __ SDS(1) e f g h SDS(0) __ i j NEL k l __ SDS(1) m n o p __ SDS(2) q r s t SDS(0) __ u v w SDS(0) __ x y z NEL (a b c SDS(1) { d e f } SDS(0) g h i }

Data Component:

	Character positions					
	1	2	3	4	5	6
	123456789-	123456789-	123456789-	123456789-	123456789-	123456789-
Line number	1	2	3	4	5	6
	abcd_	efgh_	ij			
	k1_mnop_	qrst_	uvw_	xyz		
	(abc{	def}ghi}				
	abcd_	efgh_	ij			
	k1_mnop_	qrst_	uvw_	xyz		
	(abc{	def}ghi}				

Presentation Component:

abcd_hgfe_ij	
k1_wvu_qrst_ponm_xyz	
(abc[fed]ghi}	
	ji_efgh_dcba
	zyx_mnop_tsrq_uvw_lk
	{ihg{def]cba)

BI-DIRECTIONAL TEXT - WORKED EXAMPLE 2

Description:

This example deals with editing of the bi-directional image output starting from the results of the WORKED EXAMPLE 1. It incorporates movements of the active and data positions, insertion, deletion and replacement of graphic characters.

Character insertion is always done according to the presentation direction and nesting level which are established by the data stream.

Data stream:

BDSM(EXPLICIT)

DCSM(DATA) IRM(INSERT)

CUP(2;14) SDS(1) A B C SDS(0)

CUP(3;11) SDS(2) A B C SDS(1) 1 2 3 SDS(0) D E F SDS(0)

CHA(25) DCH(2)

CHA(27) IRM(REPLACE) SDS(2) G H I SDS(0)

DCSM(PRESENTATION) CUP(4;12) IRM(INSERT) SDS(2) A B C SDS(0)

DCSM(DATA) IRM(INSERT)

CUP(5;14) SDS(2) A B C SDS(0)

CUP(6;11) SDS(1) 1 2 3 SDS(2) D E F SDS(0) 4 5 6 SDS(0)

CHA(25) DCH(2)

CHA(28) IRM(REPLACE) G H I

DCSM(PRESENTATION) CUP(7;12) IRM(INSERT) SDS(1) A B C SDS(0)

NOTE A.1

This example is based on the following arbitrary implementation assumptions:

- i) When BDSM is set to EXPLICIT and DCSM is set to DATA, the presentation direction and the nesting level of the character positions which are put into erased state (by ICH and ECH), are acquired from the data position.
- ii) When BDSM is set to EXPLICIT and DCSM is set to PRESENTATION, the presentation direction and the nesting level of the character positions which are put into erased state (by ICH and ECH), are acquired from the active position.

If the direction of the character path is same as the direction of the character progression, the nesting level for the erased character positions is set to that of the data position.

If the direction of the character path is different from the direction of the character progression, the nesting level for the erased character positions is set to that of the data position plus 1. This allows to keep both the Presentation Component and the Data Component as close as possible to the original state.

Data Component before editing:

	Character positions						
	1	2	3	4	5	6	
	123456789-123456789-123456789-123456789-123456789-123456789-						
Line number	1	abcd_efgh_ij					
	2	kl_mnop_qrst_uvw_xyz					
	3	(abc{def}ghi)					
	4	abcd_efgh_ij					
	5	kl_mnop_qrst_uvw_xyz					
	6	(abc{def}ghi)					
	7						
	8						

Data Component after editing:

	Character positions						
	1	2	3	4	5	6	
	123456789-123456789-123456789-123456789-123456789-123456789-						
Line number	1	abcd_efABCgh_ij					
	2	kl_mABC123DEFnop_qt_GHI_xyz					
	3	(abc{ABCdef}ghi)					
	4	abcd_efABCgh_ij					
	5	kl_m123DEF456nop_qt_uGHIxyz					
	6	(abc{deABCf}ghi)					
	7						
	8						

Presentation Component before editing:

abcd_hgfe_ij	
kl_wvu_qrst_ponm_xyz	
(abc{fed}ghi)	
	ji_efgh_dcba
	zyx_mnop_tsrq_uvw_lk
	{ihg{def}cba}

Presentation Component after editing:

abcd_hgABCfe_ij	
kl_IHG_qt_ponFED123CBAm_xyz	
(abc{CBAfed}ghi)	
	ji_efCBAgh_dcba
	zyxGHIm123FED456nop_tq_u_lk
	{ihg{deABCf}cba}

BI-DIRECTIONAL TEXT - WORKED EXAMPLE 3

Description:

This example shows tabulation in a bi-directional environment.

Data stream:

BDSM(EXPLICIT) SPD(0)

CUP(3;10) TALE(2) CHA(15) TATE(2) CHA(30) TCC(2)

CHA(1) HT 1 2 3 HT a b c d HT e f g

NEL HT SRS(1) a b c SRS(0) HT SRS(1) d e f g SRS(0) HT SRS(1) h i j SRS(0)

NEL SCP(2;1)

CHA(10) TALE(2) CHA(15) TATE(2) CHA(30) TCC(2)

CHA(1) HT a b c HT d e f g HT h i j

NEL HT SDS(1) 1 2 3 SDS(0) HT SDS(1) a b c d SDS(0) HT SDS(1) e f g SDS(0)

Data Component:

		Character positions					
		1	2	3	4	5	6
		123456789-123456789-123456789-123456789-123456789-123456789-					
Line	1						
	2						
	3	123	abcd	efg			
	4	abc	defg	hij			
number	5	abc	defg	hij			
	6	123	abcd	efg			
	7						
	8						

Presentation Component:

	123	abcd	efg			
	cba	gfed	jih	gfed	cba	123
			jih			
			efg	abcd		

BI-DIRECTIONAL TEXT - WORKED EXAMPLE 4

Description:

This example demonstrates implicit bi-directionality support. The data stream used in this example is similar to that used in Worked Example 1, with the following exceptions:

- no bi-directionality related control functions are contained within the data stream;
- all characters with implied left-to-right presentation direction are shown in capital letters; all characters with implied right-to-left presentation direction are shown in small letters; any character, not falling into these categories, has a presentation direction established by the device algorithm.

Two possible image outputs can be constructed on the Presentation Component: one for left-to-right presentation direction, and the other for right-to-left presentation direction. The presentation directions used for the whole Presentation Component are established externally to the presented data stream.

The device must have at least one algorithm to handle bi-directionality in an implicit manner. The algorithm used in this example is obviously not unique.

Data stream:

CUP(2;7) SLH(7) SLL(54) A B C D _ e f g h _ I J N E L K L _ m n o p _ Q
R S T _ u v w _ X Y Z N E L (A B C { d e f } G H I } N E L
CHA(7) SLH(7) SLL(54) a b c d _ E F G H _ i j N E L k l _ M N O P _ q r s
t _ U V W _ x y z N E L (a b c { D E F } g h i }

Data Component:

		Character positions					
		1	2	3	4	5	6
		123456789-123456789-123456789-123456789-123456789-123456789-					
Line	1						
	2						
	3	ABCD_efgh_IJ					
	4	KL_mnop_QRST_uvw_XYZ					
number	5	(ABC{def}GHI}					
	6	abcd_EFGH_ij					
	7	kl_MNOP_qrst_UVW_xyz					
	8	(abc{DEF}ghi}					

Presentation Component (left-to-right presentation direction):

	ABCD_hgfeIJ
	KL_ponmQRST_wvuXYZ
	(ABC{]fedGHI}
	_dcbaEFGH_ji
	_lkmNOP_tsrqUVW_zyx
	{cbaDEF}]ihg

Presentation Component (right-to-left presentation direction):

```
      IJ_hgfeABCD_  
XYZ_wvuQRST_pomKL_  
      GHI}]fedABC{(  
      j!EFGH_dcba  
zyxUVW_tsr_MNOP_lk  
      }!hgDEF]{cba(
```

NOTE A.2

As shown above, the absence of bi-directional control functions allows only limited results. The absence of SCP, for instance does not allow to have paragraphs of different presentation directions. The absence of SDS and SRS results in a poor handling of spaces and parentheses. Refer to 6 of this Technical Report for detailed explanations on implicit bi-directionality support.

Files of this document can be freely downloaded from the Ecma web site (www.ecma-international.org). This site gives full information on Ecma, Ecma activities, Ecma Standards and Technical Reports.

Free printed copies can be ordered from:



INTERNATIONAL

Rue du Rhône 114
CH-1204 Geneva
Switzerland

Fax: +41 22 849 6001
documents@ecma-international.org

ecma-international.org