ECMA Technical Report TR/53

2nd Edition - June 1992

ECMA

Standardizing Information and Communication Systems

# Handling of Bi-Directional Texts

**Phone:** +41 22 849.60.00 - **Fax:** +41 22 849.60.01 - **URL:** http://www.ecma.ch - **Internet:** helpdesk@ecma.ch

ECMA Technical Report TR/53
June 1992

ECMA

Standardizing Information and Communication Systems

# Handling of Bi-Directional Texts

**Phone:** +41 22 849.60.00 - **Fax:** +41 22 849.60.01 - **URL:** http://www.ecma.ch - **Internet:** helpdesk@ecma.ch

MB Tr-53.doc 01-07-98 16,06

## Brief History

In April 1988, ECMA TC1 decided to produce a Technical Report on specific control functions required for presentation of texts in scripts using different writing directions.

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

An ECMA TC1 ad-hoc group was formed and instructed:

- to establish the new additional control functions required in Standard ECMA-48 for handling bi-directional texts;
- to determine the required modifications to existing control functions in Standard ECMA-48 for handling bi-directional texts;
- prepare the necessary explanatory information for inclusion in Standard ECMA-48;
- to provide worked examples for the handling of bi-directional text, applying the new and the modified control functions described in this ECMA Technical Report and included in the fifth edition of Standard ECMA-48;
- to consider in particular a horizontal line orientation and a top-to-bottom line progression.

This ECMA Technical Report explains the particularities to be considered when handling bi-directional texts. It is also intended to be used as a guidance to implementors of bi-directional text applications in a character-coded environment.

This ECMA Technical Report was used as the basis for making the necessary enhancements and changes to Standard ECMA-48. The changes in and the additions to Standard ECMA-48 are also reflected in the latest edition of the corresponding International Standard, viz. ISO/IEC 6429:1992.

Adopted as an ECMA Technical Report by the General Assembly of June 1992.

**Table of contents**

# 1 Scope

This ECMA Technical Report specifies technical means to handle bi-directional text in character-imaging devices in the case of:

- texts in a single script of which specific parts need to be presented in an opposite direction (for example, numbers in Arabic or Hebrew);
- texts in different scripts presented in opposite directions (like Latin and Arabic or Latin and Hebrew);
- texts presented with a horizontal line orientation and a top-to-bottom line progression.

Other presentation directions may be the subject of further studies.

# 2 General Considerations

In this ECMA Technical Report a bi-directional device model is used to explain the handling of bi-directional texts. To be able to use the device model with Standard ECMA-48 the following modifications to that Standard were necessary:

- the device concept had to be extended;
- two new modes had to be added;
- a set of new control functions had to be included;
- the definitions of a number of existing control functions had to be changed.

The fifth edition of Standard ECMA-48 includes the appropriate changes and additions.

The control functions are intended to be used imbedded in character coded data for interchange with character imaging input/output devices that are capable of handling bi-directional texts.

The architecture of such devices is reflected by the bi-directional device model described in 6.

The method for the coded representation of the new bi-directional control functions is identical with that already used in Standard ECMA-48.

The technical capabilities of the bi-directional devices to which this Technical Report and the extended specifications of the fifth edition of Standard ECMA-48 apply, can vary according to the levels of bi-directional support which is required or provided. Different bi-directional technical means may, therefore, be selected for implementation, depending on the particularities of the specific application.

# 3 References

| | |
|---|---|
| ECMA-48 | Control functions for coded character sets (1991); |
| ISO/IEC 6429:1992 | Information technology - Control functions for coded character sets. |

# 4 Notations and definitions

## 4.1 Notation

In this ECMA Technical Report the same convention as in Standard ECMA-48 has been adopted to assist the reader. Capital letters are used to refer to a specific control function, mode, mode setting, or graphic character in order to avoid confusion, for example, between the concept "space" and the graphic character SPACE.

It is intended that this convention and the acronyms of the modes and the control functions be retained in all translations of the text.

## 4.2 Definitions

In this ECMA Technical Report the same definitions as in the fifth edition of Standard ECMA-48 apply.

# 5 Requirements for the handling of bi-directional texts

Many languages, like the 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 in these languages, for instance, are written and read from left-to-right. Finally, texts of languages with opposite presentation directions can be intermixed.

As a consequence, bi-directional character-imaging devices have to provide support:

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

## 5.1 Directions of strings

Many graphic characters have an inherent directionality. Others have no inherent directionality and abide by context. Examples of both types of the graphic characters in question are: Space, punctuation marks, separators, parentheses, and others.

In order to fully specify the directionality of a string of text constituted of graphic characters with and graphic characters without inherent directionality, control functions need to be imbedded in the text string.

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

## 5.2 Ordering of data

The order in which the graphic characters in a string of a bi-directional text (data stream) are interchanged may differ from the order in which the graphic characters are presented in the graphic image output. For example, "hello" may be presented as "olleh" on a right-to-left device.

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

## 5.3 Transparency

Applications that are designed to handle bi-directional data streams can fully control the functionality of a bi-directional device. Such applications are called "bi-directionality-aware" applications.

On the other hand, there is a need to allow applications not designed to handle bi-directional data streams to function reasonably well in a bi-directional environment, making this environment "transparent" to the application. Such applications are called "bi-directionality-unaware" applications.

# 6 The bi-directional device model

To explain the requirements for and the methods of handling bi-directional texts, a device model is defined in this ECMA Technical Report. This model was also used to extend the uni-directional device concepts defined in earlier editions of Standard ECMA-48.

Different devices complying with the model are represented schematically in figures 1 to 3 in 6.6.

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

In addition to receiving a data stream, a character-imaging device may also be capable of transmitting a data stream consisting of graphic characters and control functions. 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 an associated keyboard.

## 6.1 The device structure

A uni-directional device as referred to in this ECMA Technical Report is shown in figure 1.

A bi-directional device as described in this model consists of either

- an input component, a data component, a presentation component, and a graphic image output as shown in figure 2, or
- an input component, a presentation component only, and a graphic image output as depicted in figure 3.

### 6.1.1 The input component

The input component is used for receiving the data stream. In addition, the input component may receive data from a manual input device such as a keyboard or a mouse.

This ECMA Technical Report does not deal with the input component.

### 6.1.2 The data component

The data component is used to store the information received from the input component and to make it available to a presentation process that transforms the information for the presentation component for subsequent graphic image output. A data component is generally not provided in uni-directional devices.

The data component structures the information in successive lines; each line consisting of successive character positions.

The lines in the data component convey the organizational aspects of the information. In the data component lines have no orientation; to simplify matters their orientation is considered to be horizontal only.

The sequential order of the lines is called the line progression. In the data component the line progression is considered to be from top-to-bottom only. The lines are counted in the direction of the line progression and are numbered consecutively by the numbers 1, 2, 3 ...

The sequential order of the character positions along a line in the data component is called the character progression. The character progression is considered to be from left-to-right only. The character positions along a line are counted in the direction of the character progression and are numbered consecutively by the numbers 1, 2, 3 ...

At any time there is a unique character position in the data component which is available for the next graphic character or relative to which certain control functions are to be executed. This character position is called the active data position. The active data position can be moved implicitly or explicitly or indirectly.

The line in the data component containing the active data position is called the active data line, the field in the data component containing the active data position is called the active data field, the area in the data component containing the active data position is called the active data area, the page in the data component containing the active data position is called the active data page.

### 6.1.3 The presentation component

The presentation component is used for receiving the information from the data component through the presentation process and for producing the graphic image output. This output may, for example, be rendered on a display or a printer. A presentation component is provided in bi-directional as well as in uni-directional devices.

The presentation component structures the information into successive lines; each line consisting of successive character positions.

The lines in the presentation component convey the graphic image output aspects and their orientation can be considered to be either horizontal or vertical. This ECMA Technical Report deals only with horizontal line orientation.

The sequential order of the lines is called the line progression. The lines are counted in the direction of the line progression and are numbered consecutively by the numbers 1, 2, 3 ... For horizontal line orientation the direction of the line progression can be considered to be either from top-to-bottom or from bottom-to-top. This ECMA Technical Report deals only with the line progression from top-to-bottom.

The sequential order of the character positions along a line in the presentation component is called the character path. For horizontal line orientation the character path can be either from left-to-right or from right-to-left. The character positions along a line are counted in the direction of the character path and are numbered consecutively by the numbers 1, 2, 3 ...

At any time there is a unique character position in the presentation component which is available for the next graphic character or relative to which certain control functions are to be executed. This character position is called the active presentation position. The active presentation position can be moved implicitly or indirectly. In the case where a device has no data component, the active presentation position can also be moved implicitly. It is common practice to mark the active presentation position in a graphic image output by a special visible indicator called the cursor.

The line in the presentation component containing the active presentation position is called the active presentation line, the field in the presentation component containing the active presentation position is called the active presentation field, the area in the presentation component containing the active presentation position is called the active presentation area, the page in the presentation component containing the active presentation position is called the active presentation page.

#### 6.1.4 The graphic image output

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

The graphic image output usually consists of pages that are composed of a predetermined number of lines. The lines are composed of a predetermined number of character positions. The size of a character position may be fixed or may depend on the graphic symbol of the character being imaged.

In the case of a character-imaging device with a data component and a presentation component, the graphic image output is created in the presentation component by the presentation process from the information in the data component.

In the case of a character-imaging device with a presentation component only, the graphic image output is created in the presentation component from the information in the input component.

### 6.2 Relationship between the active data position and the active presentation position

The relation between the active data position and the active presentation position depends on whether the device is a uni-directional one or a bi-directional one and whether it has both a data and a presentation component or a presentation component only.

In a uni-directional device, no distinction can be made between an active data position and an active presentation position; they are considered to be equivalent and are referred to as active position only.

In the case where a bi-directional device has a presentation component only, all references to the data component, to the active data position, to the character progression, etc. are to be considered as if they were references to the presentation component, to the active presentation position, to the character path, etc.

In a bi-directional device with a data component and a presentation component, the active presentation position is the character position in the presentation component that corresponds to the active data position in the data component. Because the direction of the character progression (in the data component) may be different from the direction of the character path (in the presentation component) in particular bi-directional applications, the coordinates of the active data position in the data component and the active presentation position in the presentation component may also be different.

Some control functions act on and affect the active data position, while other control functions act on and affect the active presentation position. Examples of such functions are: character insertion, character erasure, reference point movements, etc. When one of the active positions is moved, the other is subjected to a corresponding displacement, although not necessarily in the same direction. This displacement is referred to as indirect movement of the 'other' position.

### 6.3 Movement of the active positions

As a result of the content of the received data stream, the active data position as well as the active presentation position can be moved in their corresponding components. The movement can be implicit, explicit or indirect.

#### 6.3.1 Implicit movement

An implicit movement of the active data position in the data component is performed after receiving a graphic character or after receiving a control function for which a graphical representation is required.

In a bi-directional device the direction of the implicit movement may be different from the direction of the character progression. The direction is the same as the direction of the character progression unless it is modified by an appropriate control function.

If the direction of the implicit movement is the same as that of the character progression and the active data position is not the last character position (line limit position) of the active line, the active data position is moved to the following character position of that line.

If the direction of the implicit movement is opposite to that of the character progression and the active data position is not the first character position (line home position) of the active line, the active data position is moved to the preceding character position of that line.

After an implicit movement of the active data position, the active presentation position in the presentation component is updated accordingly; this is referred to as indirect movement (see 6.3.3).

In a bi-directional device without a data component the implicit movement applies to the active presentation position in the presentation component and is then the same as the direction of the character path.

*NOTE*

*In the following situation, the effect of an attempt to move the active data position is not defined:*

- *an attempt to perform an implicit movement when the active data position is the last character position (line limit position) of a line and the direction of the implicit movement is the same as that of the character progression, or when the active data position is the first character position (line home position) of a line and the direction of the implicit movement is opposite to that of the character progression.*

### 6.3.2 Explicit movement

In the data component an explicit movement of the active data position is performed when a control function is executed which causes the active data position to be moved to a specified character position. After an explicit movement of the active data position, the active presentation position in the presentation component is updated accordingly; this is referred to as indirect movement (see 6.3.3).

In the presentation component an explicit movement of the active presentation position is performed when a control function is executed which causes the active presentation position to be moved to a specified character position. After an explicit movement of the active presentation position, the active data position in the data component is updated accordingly; this is referred to as indirect movement (see 6.3.3).

*NOTE*

*In the following situation, the effect of an attempt to move the active data position or the active presentation position is not defined:*

- *an attempt to perform an explicit movement to a non-existing character position, for example beyond the last character position (line limit position) of a line, or beyond the last line of a page, i.e. beyond the page limit position.*

### 6.3.3 Indirect movement

In the data component an indirect movement of the active data position is performed to reflect an explicit movement (see 6.3.2) of the active presentation position in the presentation component.

In the presentation component an indirect movement of the active presentation position is performed to reflect an implicit movement (see 6.3.1) or an explicit movement (see 6.3.2) of the active data position in the data component.

## 6.4 Data stream and data organization

The data stream is considered to be a continuous stream. It may be structured in messages, records, and/or blocks, but that does not affect the operation of the character-imaging device at the abstract level of description in this ECMA Technical Report.

The text in a data stream can be viewed as being constructed from character strings. Each such string may contain nested strings. The graphic characters in the strings are organized in the order in which they are intended to be read by a human being.

Each string has a direction associated with it. This association may be accomplished by using an appropriate control function or by using a higher-level protocol. If the direction of a string is not determined in this way, the direction is taken to be the same as that of the currently established character path.

## 6.5 Areas, fields and tabulation

The information in the presentation component may be structured into strings of successive character positions. These strings are called areas or fields. They may have attributes associated with them.

### 6.5.1 Areas

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

For the purpose of the device model in this ECMA Technical Report two types of area are defined. Further information on the different areas and their use can be found in Standard ECMA-48.

**Selected areas**

A selected area is used to indicate a string of character positions in the presentation component. The contents of this string may be transmitted in form of a data stream, or be transferred to an auxiliary device. The character positions in a line of a selected area are ordered in the same direction as the direction of the character path currently established for that line.

**Qualified areas**

A qualified area is similar in its functionality to a selected area. It is used to indicate a string of character positions in the presentation component. With the string certain attributes or restrictions may be associated. These can be, among others: protecting the contents of the string against erasure or manual alteration, restricting the kinds of graphic characters that can be included in the string. The character positions in a line of a qualified area can be ordered in the same direction as the character path or opposite to the direction of the character path currently established for that line.

### 6.5.2 Fields and tabulation

A field is a string of successive character positions in the presentation component. The beginning of a field is marked by a character tabulation stop set at the first character position of the field. The type of tabulation stop used to mark the beginning of the field may also indicate the kind of alignment to be used with the first character of the field. Examples of the different kinds of alignment are: leading edge of the character, centred within the line, centred on the character, trailing edge of the character. The end of the field is indicated by the next character tabulation stop; the last character position of the field is the one immediately preceding this tabulation stop.

The character positions in the field, as well as the character tabulation stops in a line of the field are ordered in the same direction as the direction of the character path currently established for that line.

## 6.6 Simplified presentation of the device model

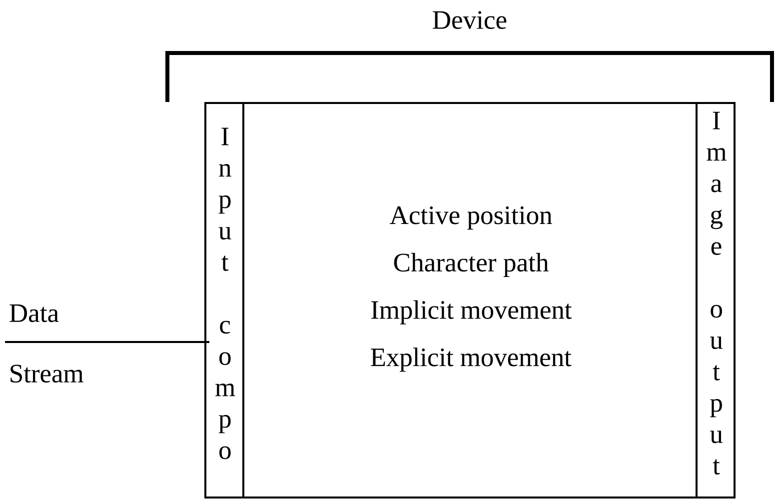


**Figure 1 - Uni-directional device**

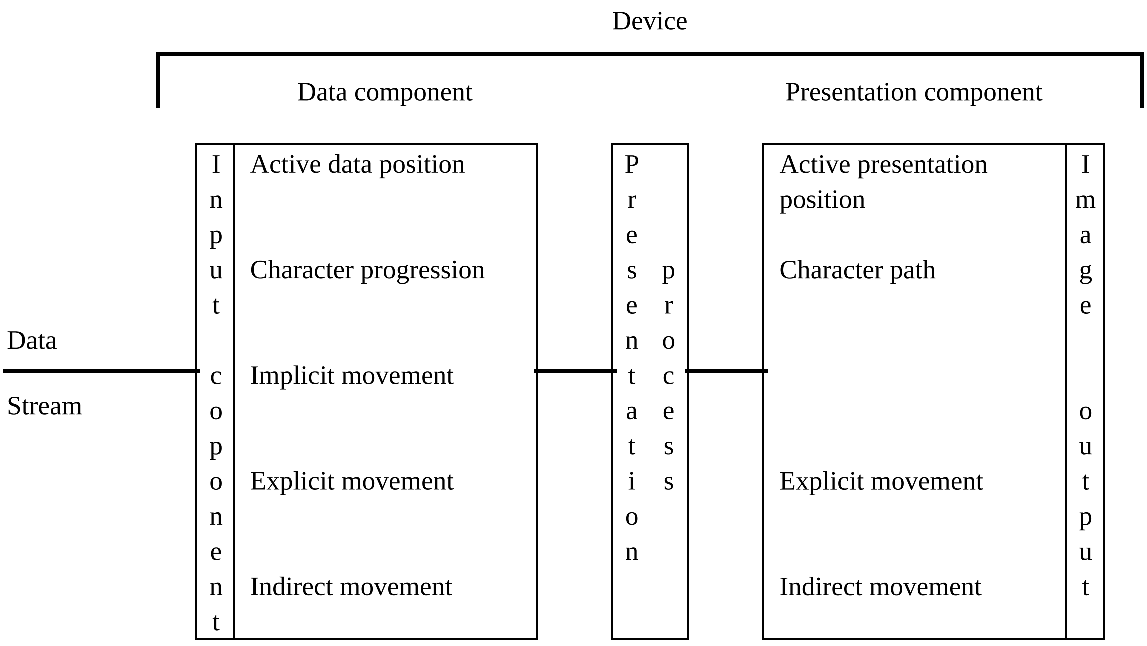


**Figure 2 - Bi-directional device with data and presentation component**

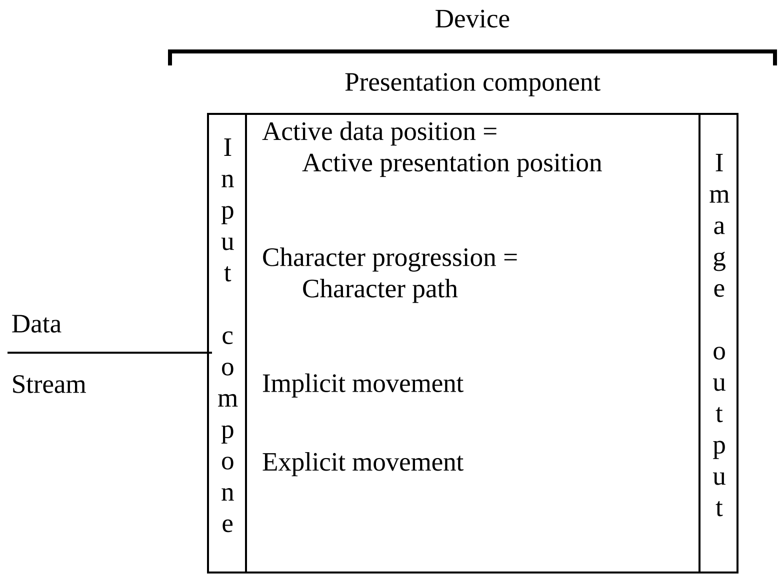


**Figure 3 - Bi-directional device with presentation component only**

# 7 The changes and additions made to Standard ECMA-48

## 7.1 New modes for handling bi-directionality

Standard ECMA-48 did not provide for the necessary control functions for handling bi-directional text applications. This situation could have been amended in two ways:

– by specifying a significant number of new control functions and by making substantial modifications to existing control functions;

– by introducing new modes, thus keeping the number of new control functions and modified control functions to a minimum.

Although the use of modes had been deprecated in earlier editions of Standard ECMA-48 it was decided that the more viable of the potential solutions was the introduction of two new modes.

**The BI-DIRECTIONAL SUPPORT MODE (BDSM)**

A bi-directional device according to figure 2 in 6.6 can support two kinds of data streams:

– a data stream coming from an application designed to specifically handle bi-directional data on bi-directional devices ("bi-directionality-aware" application);

– a data stream coming from an application designed to handle uni-directional data on bi-directional devices ("bi-directionality-unaware" application).

BDSM allows to select between the support for "bi-directionality-aware" applications (the EXPLICIT state) and the support for "bi-birectionality-unaware" applications (the IMPLICIT state).

In the case of "bi-directionality-unaware" applications, the method of handling the bi-directional data depends on the receiving device because the data stream does not contain the control functions necessary to achieve the desired result. The following information will help to determine the appropriate actions to be taken in the device: An interpretation of the data stream structure and of the actual content (semantics) and the implied inherent directionality of the graphic characters in the data stream (given the context). This information cannot always be obtained in a unique and unambiguous manner from the data stream alone.

**The DEVICE COMPONENT SELECT MODE (DCSM)**

In the case of a "bi-directionality-aware" application (BDSM set to EXPLICIT), the data stream contains the appropriate control functions to achieve the desired result. However, there must be a means of specifying whether the control functions apply to the presentation component or to the data component.

DCSM allows to specify whether the relevant control functions are to be performed in the presentation component (the PRESENTATION state) or in the data component (the DATA state).

In the case of a "bi-directionality-unaware" application (BDSM set to IMPLICIT), the setting of DCSM has no effect. It is considered to be set to DATA (the reset state).

The new modes are specified in the fifth edition of Standard ECMA-48 (for BDSM the reset state should be EXPLICIT and the set state IMPLICIT). To preserve a uni-directional left-to-right behaviour on a bi-directional device, BDSM should be set to EXPLICIT and DCSM to PRESENTATION.

## 7.2 Control functions added for handling bi-directionality

Standard ECMA-48 contained control functions to determine, for example, the presentation direction to be used by a character-imaging device. On the other hand, it was not possible to specify the line orientation relative to the character path and the line progression or the character path relative to the line orientation. Apart from making changes to existing control functions, three new control functions were deemed necessary to complement the existing ones in Standard ECMA-48.

**SELECT CHARACTER PATH (SCP)**

SCP allows in the presentation component the selection of the character path relative to the line orientation set by SELECT PRESENTATION DIRECTIONS (SPD). The overall directions are established by SPD. SCP can be used to select the character path for a certain number of lines without affecting the other established directions and orientations.

**START DIRECTED STRING (SDS)**

SDS allows to establish a specific direction for a string of characters in the data component. Standard ECMA-48 contained already the control function START REVERSED STRING (SRS) permitting to establish the opposite direction <u>relative</u> to that currently established. SDS now also allows to determine the <u>absolute</u> direction of a string it indicates the beginning of.

**SELECT IMPLICIT MOVEMENT DIRECTION (SIMD)**

SIMD allows to determine the direction of the implicit movement in the data component without affecting the established character progression. The direction of the implicit movement can be either the same as the direction of the character progression or the opposite. This new control function influences the effects of the control functions BACKSPACE (BS), CARRIAGE RETURN (CR), and NEXT LINE (NEL).

## 7.3 Control functions modified for handling bi-directionality

A number of the existing control functions in Standard ECMA-48 had their definitions changed or extended to permit their use in bi-directional applications. Others were changed to include, for example, the new term "active presentation position" in place of the former term "active position". The following list includes only those control functions with substantial changes.

**BACKSPACE (BS)**

BS moves the active data position one character position in the direction opposite to that of the implicit movement. Now that the direction of the implicit movement can be changed by the parameter value of SELECT IMPLICIT MOVEMENT DIRECTION (SIMD) the definition of BS was changed to cover the new dependency.

**ACTIVE POSITION REPORT (CPR)**

Whether CPR reports the active data position or the active presentation position depends on the setting of the DEVICE COMPONENT SELECT MODE (DCSM).

**CARRIAGE RETURN (CR)**

CR moves either the active data position or the active presentation position depending on the setting of the DEVICE COMPONENT SELECT MODE (DCSM). Also, depending on the parameter value of SELECT IMPLICIT MOVEMENT DIRECTION (SIMD) the active position is moved to the line home position or the line limit position.

**DELETE CHARACTER (DCH)**

Whether characters are deleted in the data component or in the presentation component depends now on the setting of the DEVICE COMPONENT SELECT MODE (DCSM).

**DELETE LINE (DL)**

Whether lines are deleted in the data component or in the presentation component depends now on the setting of the DEVICE COMPONENT SELECT MODE (DCSM).

**ERASE IN AREA (EA)**

Whether characters are erased in an area in the data component or in an area in the presentation component depends now on the setting of the DEVICE COMPONENT SELECT MODE (DCSM).

**ERASE CHARACTER (ECH)**

Whether characters are erased in the data component or in the presentation component depends now on the setting of the DEVICE COMPONENT SELECT MODE (DCSM).

**ERASE IN PAGE (ED)**

Whether characters are erased in a page in the data component or in a page in the presentation component depends now on the setting of the DEVICE COMPONENT SELECT MODE (DCSM).

**ERASE IN FILE (EF)**

Whether characters are erased in a file in the data component or in a file in the presentation component depends now on the setting of the DEVICE COMPONENT SELECT MODE (DCSM).

**ERASE IN LINE (EL)**

Whether characters are erased in a line in the data component or in a line in the presentation component depends now on the setting of the DEVICE COMPONENT SELECT MODE (DCSM).

**INSERT CHARACTER (ICH)**

Whether characters are inserted in the data component or in the presentation component depends now on the setting of the DEVICE COMPONENT SELECT MODE (DCSM).

**INSERT LINE (IL)**

Whether lines are inserted in the data component or in the presentation component depends now on the setting of the DEVICE COMPONENT SELECT MODE (DCSM).

**LINE FEED (LF)**

Depending on the setting of the DEVICE COMPONENT SELECT MODE (DCSM) this control function affects either the active data position or the active presentation position.

**NEXT LINE (NEL)**

Depending on the setting of the DEVICE COMPONENT SELECT MODE (DCSM) either the active data position or the active presentation position is moved. Furthermore, depending on the parameter value of SELECT IMPLICIT MOVEMENT DIRECTION (SIMD) the active position is moved to either the line home position or the line limit position.

**REVERSE LINE FEED (RI)**

Equivalent change as to LINE FEED (LF) above.

**SELECT ALTERNATIVE PRESENTATION VARIANTS (SAPV)**

For some of the parameter values the effect depends on the direction of the character path. Also, a number of new parameter values were added.

**SET LINE HOME (SLH)**

The setting of the DEVICE COMPONENT SELECT MODE (DCSM) determines whether the line home position is set in the data component or in the presentation component.

**SET LINE LIMIT (SLL)**

The setting of the DEVICE COMPONENT SELECT MODE (DCSM) determines whether the line limit position is set in the data component or in the presentation component.

**SELECT PRESENTATION DIRECTIONS (SPD)**

New parameter values were added to permit selecting the line orientation in addition to the character path and the line progression in the presentation component.

**SET PAGE HOME (SPH)**

The setting of the DEVICE COMPONENT SELECT MODE (DCSM) determines whether the page home position is set in the data component or in the presentation component.

**SET PAGE LIMIT (SPL)**

The setting of the DEVICE COMPONENT SELECT MODE (DCSM) determines whether the page limit position is set in the data component or in the presentation component.

**START REVERSED STRING (SRS)**

This control function is effective in the data component. Information on the interdependency with START DIRECTED STRING (SDS) was included in the definition.

# 8 Levels of device support for bi-directionality

Because of the diversity of application and device requirements, different levels of bi-directional support may be considered. This ECMA Technical Report proposes the following two:

## 8.1 Bi-directional devices with presentation component and data component

(see figure 2)

This level addresses presentation devices with full bi-directional functionality, allowing the support of either "bi-directionality-aware" or "bi-directionality-unaware" applications. This is achieved by supporting both explicit and implicit handling of data. On these devices all bi-directional control functions and modes can be supported without limitation.

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 at least a limited bi-directional support.

## 8.2 Bi-directional devices with presentation component only

(see figure 3)

This level addresses presentation devices with limited bi-directional functionality and high dependency on application software. On these devices the following control functions can be supported with some possible limitations:

- SELECT CHARACTER PATH (SCP);
- SELECT PRESENTATION DIRECTIONS (SPD);
- SELECT IMPLICIT MOVEMENT DIRECTION (SIMD);
- SELECT ALTERNATIVE PRESENTATION VARIANTS (SAPV);
- also BACKSPACE (BS), CARRIAGE RETURN (CR) and NEXT LINE (NEL) the effects of which are influenced by SIMD.

The bi-directional modes BDSM and DCSM are not supported at this level.

# Annex A

## Worked examples

This annex provides examples of the use of the control functions and modes defined in the fifth edition of Standard ECMA-48 and described in this ECMA Technical Report. These examples cover:

– the combined use of control functions in bi-directional texts,

– the nesting of strings,

– the definition of fields,

– the use of the two new modes.

The device used in the following examples has both a data and a presentation component. It is a device as shown in figure 2 of this ECMA Technical Report.

Each worked example comprises the following sections:

**Description**

This section provides 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 a space. Each control function is shown with its acronym and parameter values. Control functions are shown in **bold** text. The graphic symbol representing the graphic character SPACE is a low line shown as **_** .

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

**Data Component**

This section shows the data as stored in the data component, prior to being submitted to the presentation process for subsequent presentation by the presentation component in the graphic image output. For ease of reference, the lines as well as the character positions in each line of the data component are numbered.

In order to avoid a 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 the data component (8 lines of 60 character positions each) is shown.

**Presentation Component**

The presentation component shows the content of the graphic image output that should occur as the result of the presentation of the information in the data stream.

The lines and the character positions in the presentation component are not numbered as the numbering scheme can change depending on the 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 the presentation of parentheses.

**Data stream:**

**BDSM(EXPLICIT) SAPV(3)**

**SPD(0;1) 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-
       1
       2       abcd_efgh_ij
    L  3       kl_mnop_qrst_uvw_xyz
    i  4       (abc{def]ghi}
    n  5       abcd_efgh_ij
    e  6       kl_mnop_qrst_uvw_xyz
    s  7       (abc{def]ghi}
       8
```

**Presentation component:**

```
  abcd_hgfe_ij
  kl_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 the editing of the bi-directional graphic image output starting from the results of WORKED EXAMPLE 1. It incorporates movements of the active presentation position and the active data position, also the insertion, deletion and replacement of graphic characters.

Character insertion is always performed according to the presentation directions and the 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 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 the erased state (by ICH and ECH), are acquired from the active 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 the erased state (by ICH and ECH), are acquired from the active presentation position.*

  *If the direction of the character path is the same as the direction of the character progression, the nesting level for the erased character positions is set to that of the active 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 active 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-
   1
   2        abcd_efgh_ij
L  3        kl_mnop_qrst_uvw_xyz
i  4        (abc{def]ghi}
n  5        abcd_efgh_ij
e  6        abcd_efgh_ij
s  7        kl_mnop_qrst_uvw_xyz
   8        (abc{def]ghi}
```

**Data component after editing**:

```
                  Character positions
                 1         2         3         4         5         6
      123456789-123456789-123456789-123456789-123456789-123456789-
   1
   2        abcd_efABCgh_ij
L  3        kl_mABC123DEFnop_qt_GHI_xyz
i  4        (abc{ABCdef]ghi}
n  5        abcd_efABCgh_ij
e  6        kl_m123DEF456nop_qt_uGHIxyz
s  7        (abc{deABCf]ghi}
   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 how tabulation works 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-
     1
     2         123    abcd          efg
  L  3         abc    defg          hij
  i  4         abc    defg          hij
  n  5         123    abcd          efg
  e  6
  s  7
     8
```

**Presentation component:**

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

## BI-DIRECTIONAL TEXT - WORKED EXAMPLE 4

**Description:**

This example demonstrates the implicit handling of bi-directionality by means of appropriate features of the device itself. 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 in the data stream;
- all graphic characters with implied left-to-right presentation direction are shown in capital letters; all graphic characters with implied right-to-left presentation direction are shown in small letters; any graphic character, not falling into the two categories mentioned above, has a presentation direction established by the device algorithm.

Two possible graphic image outputs can be constructed in 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 the only one that could be applied.

**Data stream:**

**CUP(2;7) SLH(7) SLL(54)** A B C D _ e f g h _ I J **NEL** K L _ m n o p _ Q R S T _ u v w _ X Y Z **NEL** ( A B C { d e f ] G H I } **NEL**
**CHA(7) SLH(7) SLL(54)** a b c d _ E F G H _ i j **NEL** k l _ M N O P _ q r s t _ U V W _ x y z **NEL** ( 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-
        1
        2       ABCD_efgh_IJ
      L 3       KL_mnop_QRST_uvw_XYZ
      i 4       (ABC{def]GHI}
      n 5       abcd_EFGH_ij
      e 6       kl_MNOP_qrst_UVW_xyz
      s 7       (abc{DEF]ghi}
        8
```

**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__ponmKL_
                GHI}]fedABC{(
                 jiEFGH__dcba
         zyxUVW__tsr_MNOP__lk
                }ihgDEF]{cba(
```

*NOTE A.2:*

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

Printed copies can be ordered from:

**ECMA**
114 Rue du Rhône
CH-1204 Geneva
Switzerland

Fax: +41 22 849.60.01
Internet: documents@ecma.ch

Files can be downloaded from our FTP site, **ftp.ecma.ch,** logging in as **anonymous** and giving your E-mail address as **password**. This Technical Report is available from library **ECMA-TR** as a compacted, self-expanding file in MSWord 6.0 format (file T053-DOC.EXE) and as an Acrobat PDF file (file T053-PDF.PDF). File T053-EXP.TXT gives a short presentation of the Technical Report.

Our web site, http://www.ecma.ch, gives full information on ECMA, ECMA activities, ECMA Standards and Technical Reports.

**ECMA**

**114 Rue du Rhône**
**CH-1204 Geneva**
**Switzerland**

**This Technical Report ECMA TR/53 is available free of charge in printed form and as a file.**

**See inside cover page for instructions.**