ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

A SET OF I/O PROCEDURES for ECMALGOL

January 1967

BRIEF HISTORY

ECMA TC5, after having issued the Standard ECMA-2 for ECMALGOL and a proposal for Hardware Representation of ALGOL basic symbols, devoted their activity at their last meetings in 1965 and 1966 to following item of their program of work:

> "To propose methods for the description of Input and Output facilities and to report on the extent to which it is possible to define common I/O facilities."

Following Companies took an active part in this work:

Compagnie des Machines Bull
N.V. Electrologica
Elliott Bros. (London) Ltd
English Electric Leo Marconi Computers Ltd
IBM-WTEC
I.C.T. International Computers and Tabulators Ltd
I.T.T. Europe Inc.
N.C.R. The National Cash Register Company Ltd
UNIVAC Computers (Europe) Ltd

It turned out difficult to define a common I/O system, and it did not appear appropriate to recommend the adoption of an ECMA Standard for the following reasons:

i) The original ALGOL Report had no I/O provisions.

ii) Standardization efforts came too late in the field of I/O for ALGOL as most implementors have by now devised their own I/O facilities.

iii) Existing ALGOL implementations are not likely to be reorganized in order to cope with an I/O standard.

iv) Several approaches have been considered also valid and implementors needs vary considerably depending on particular configurations.

Nevertheless, it was agreed that a rather basic proposal would be very useful inasmuch as most features of such a proposal could also be used for more sophisticated implementations. The present proposal is therefore submitted to ECMALGOL implementors as a valuable option for I/O procedures applicable to a majority of configurations.

ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

A SET OF I/O PROCEDURES FOR ECMALGOL

For the purpose of input/output, certain conventions are proposed which take the form of procedures which define input/output. It is expected that these procedures will be implemented in code. Input/output is intimately connected with hardware and consequently variations must be expected both between different channels on one machine and different implementations. This document leaves undefined these areas and concentrates on specifying a basic input/output facility that can be expanded firstly,by the individual implementor who will define actions undefined in this document and secondly, by the individual user who will build up a set of more complex procedures to satisfy his normal needs.

The term channel is used in this document to describe the destination or source of data occurring in an Algol program.

Mechanism

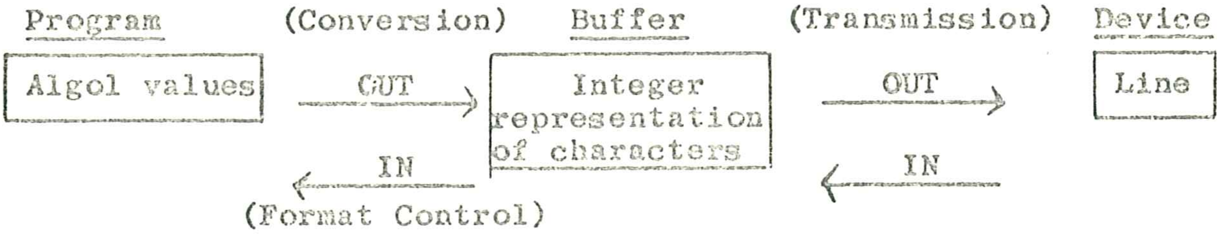
Input and output is conceived in terms of records or lines which in most cases precisely correspond to printed lines on a page. For each channel there will be defined a set of allowable characters each of which is required to correspond uniquely to one and only one positive integer. The set of characters comprises distinguishable marks each of which occupies one printing position.

The set of characters will vary with the implementation but the following is considered the minimum set:

letters	A to Z
digits	0 to 9
symbols	+ - . $_{10}$, ␣ (standing for space)

I/O takes place through the medium of a fictitious buffer which can accommodate the ordered set of integers corresponding to the characters comprising a line. The correspondence between an integer value and the corresponding character can be determined by the integer procedure EQUIV.

The action of I/O is shown diagrammatically below:



It will be seen that I/O comprises three concepts:

conversion — the changing between Algol values (of type Boolean, integer, real) in a program and integer representations of characters within a buffer.

transmission — the movement between characters represented as integers within a buffer and characters recorded on some external device.

format control — the process under which conversion is controlled.

These words (and associated verbs) are used with these meanings throughout this paper. They are modified by IN or OUT when a particular direction is implied.

Conversion and format control are effected by procedure calls.

In practice there are a number of buffers each one associated with a particular channel Q. A channel is associated with a physical device but the exact way in which devices are attached to channels is not defined in this paper. A particular channel may not be used at the same time for transmission both IN and OUT.

Each buffer has associated with it a pointer P which normally takes values between 0 and B-1 where B is the size of the buffer. The size of the buffer may be altered dynamically by means of the procedure LINELENGTH and is also dependent on hardware restrictions which determine an upper limit to the size of each buffer. At any time the size, B, of the buffer is to be considered the smaller of these two numbers. For an OUT buffer the pointer P determines the next position in which a character will be placed: a call of an OUT conversion procedure will attempt to fill locations, P, P+1, P+2, ... of the buffer and will automatically update the pointer. Conversely for an IN buffer the pointer P determines the position of the character at which IN conversion will next commence: the pointer will be automatically incremented by the number of characters used to determine the next IN value as defined by the IN format. Access to the pointer P is provided through special procedure calls.

Each line of information is transmitted as a complete buffer containing representations of the characters comprising the line. The line itself will normally contain several different values and the separation of the line into values is determined by the format control.

The format conventions are such that a set of values converted OUT can subsequently be converted IN and will reproduce the same values, provided that the identical format has been specified.

Each line is treated as a self-contained unit and it is not possible for one number to overlap the boundary between two lines.

Each channel has associated with it a line pointer R, which is incremented by 1 each time a line of information is transmitted. Access to the pointer R is provided through special procedure calls.

Format Control

Conversion is controlled by format. On entry to an Algol program a standard format is automatically invoked; methods are described later by which this format may be changed. Standard format is as follows:-

for IN conversion — numbers are of unlimited length, terminated only by syntax, buffer size or 2 successive space characters. Boolean characters are converted from 0 or 1.

for OUT conversion — Integer values are converted to the form N digits (of which up to N-1 leading zeros are replaced by spaces); the first digit printed is preceded by sign. Real values are converted to the form sign followed by decimal point and M digits followed by base ten, sign and 2 digits.

Boolean values are converted to the form standing for true or false respectively.

In the above, sign stands for space if positive and for - if negative while N, M are integers defined locally for the implementation.

Conversion Procedures

Conversion is invoked by a call of one of the following procedures.

IN Conversion

In the case of IN conversion, the process is as follows:

step 1 If the buffer pointer P exceeds the buffer size, INNEWLINE (Q) is invoked.

step 2 The next characters from the buffer are examined and ignored until either

(a) a permissible character (i.e. one out of the set of characters of which the value being converted is made up) is found

or (b) the format size is exhausted

or (c) the buffer is exhausted

whichever is first. In case (b) the procedure is exited and the result undefined. In case (c) the procedure INNEWLINE (Q) is invoked and step 2 repeated. In case (a) step 3 is entered.

step 3 Successive characters are converted and formed into a value until the conversion is terminated.

A termination is any of

(a) a character that does not belong to the set of characters of which the value being converted is made up.

(b) exhaustion of the maximum number of characters specified by format control to contribute towards one value, if applicable.

(c) exhaustion of the buffer.

(d) violation of the syntax of the value being assembled: e.g. in the case of an integer, the sequence + 23 - 14 would cause termination after the digit 3.

(e) the occurrence of K successive spaces; the value of K can be modified by the procedure SPACETERMINATE.

If termination is through case (d) the result is undefined.

In any event the buffer pointer (Q) is incremented for each character converted and on conclusion of the process it points to the next character position, which in case of termination (c) will be a fictitious position outside the buffer.

The permissible character of which ALGOL values are comprised are:

integer	:	+ - 0 to 9
real	:	+ - 0 to 9 . $_{10}$
boolean	:	0 1 (unless modified by BOOLEAN)

B1 **procedure** INREAL (Q,X); **value** Q; **integer** Q; **real** X;

comment the next characters are converted from the buffer in an attempt to form a real number whose value is assigned to X;

B2 **procedure** ININTEGER (Q,X); **value** Q; **integer** Q,X;

comment the next characters are converted from the buffer in an attempt to form an integer whose value is assigned to X;

B3 **procedure** INBOOLEAN (Q,X); **value** Q; **integer** Q; **boolean** X;

comment the next characters are input from the buffer in order to form a boolean value. Strings which represent true and false, respectively, are used according to the current format.

OUTconversion

The process of OUTconversion is the inverse of INconversion. Values are converted to characters which are placed in the appropriate buffer. The conversion is under the control of a format. If the value cannot be properly expressed in the current format alarm conversion takes place: this is defined locally for each implementation.

The process is further described as follows:-

step 1 compute the number of characters N resulting from convert sign. If there is room in the buffer proceed to step 2 otherwise invoke procedure OUTNEWLINE (Q) and then go to step 2.

step 2 place the integer representations of the value, under format control as successive characters in the buffer, increment the buffer pointer P by N, and exit. Note that if N > B the result of output is undefined.

B4 procedure OUTINTEGER (Q,X); value Q,X; integer Q,X;

comment This procedure converts the integer value X into its character forms under format control which are placed in the buffer;

B5 procedure OUTREAL (Q,X); value Q,X; integer Q; real X;

comment This procedure converts the real value X into its character forms, under format control, which are placed in the buffer;

B6 procedure OUTBOOLEAN (Q,X); value Q,X; integer Q; Boolean X;

comment X is converted and placed in the buffer under format control;

B7 procedure OUTSTRING (Q,X); value Q,X; integer Q; string X;

comment The string X is converted and placed in the buffer under format control;

Control Procedures

B8 procedure INNEWLINE (Q); value Q; integer Q;

comment Purpose: transmission.

The buffer is emptied, the next line on channel Q is transmitted IN to the buffer, P is reset to zero and the line pointer Q is incremented by 1,. The effect of this procedure when Q determines an OUT buffer is undefined. The method of terminating the end of a line for IN transmission will depend on hardware;

B9 procedure OUTNEWLINE (Q); value Q; integer Q;

comment Purpose: transmission

This procedure causes the buffer of channel Q to be transmitted OUT, the appropriate linepointer to be incremented by 1, P to be reset to zero and the buffer to be emptied. An empty buffer is defined as one containing all spaces and with P=0.

The definition of empty buffer is channel dependent, but is such that the IN transmission of characters previously transmitted OUT is possible. The effect of OUTNEWLINE when Q defines an IN channel is undefined. The mechanism of defining a physical new line will depend on the device allocated to channel Q: for example, if Q is attached to a paper tape punch the symbol NL may be transmitted while a lineprinter would result in physical movement of paper. Successive calls of OUTNEWLINE are accepted. If the optional page layout procedures are provided the linepointer value is checked against page size and if equal to or larger than this, the procedure NEWPAGE is invoked;

B10 integer procedure POINTER (Q); value Q;
integer Q;

comment Purpose: linelayout

The value is that of the current position of buffer pointer P of channel Q;

B11. procedure SETPOINTER(Q,P); value Q,P;
integer Q,P;

comment Purpose: line layout

The buffer pointer for channel Q is set to the value P. The effect of $P < 0$ or $P \geq B$ is undefined;

B12 procedure LINELENGTH (Q,L); value Q,L;
integer Q,L;

comment Purpose: line layout

B, the size of the buffer of channel Q is set to the smaller of L and the physical maximum size;

E E3 integer procedure LINEPOINTER (R); value R;
integer R;

comment Purpose: page layout

This gives the value of the LINEPOINTER R of channel Q which may be used to implement page layout procedures;

E14 procedure PAGESIZE (Q,L); value Q,L; integer Q,L;

comment Purpose: page layout

The page size of the buffer of channel Q is set to the smaller of L, and any physical upper limit;

B15 procedure NEWPAGE (Q); value Q; integer Q;

comment Purpose: outpage

Instructions are given in channel Q to start a new page and to reset P and R to zero. The string defined by the procedure TITLE is output;

B16 integer procedure CHARACTER (Q,i); value Q,i; integer Q,i;

comment Purpose: buffer access

This procedure enables access to be gained to individual characters in the buffer, thus providing for special format effects. The procedure gives as value the integer corresponding (that is defined by procedure EQUIV) to the i^{th} $(0 \leqslant i \leqslant B -1)$ character of the buffer of channel Q;

B17 procedure SETCHARACTER (Q,i,j); value Q,i,j; integer Q,i,j;

comment Purpose: buffer access

The i^{th} value of the buffer of channel Q is set to the value j. This will normally have been obtained by use of the procedure CHARACTER;

B18 integer procedure EQUIV (A); string (A);

comment Purpose: buffer access

The value of the procedure is a non-negative integer which is in one to one correspondence with the single character given as parameter A. A possible set of integer values is that defined in the ECMA standard 6-bit code table;

Format Control

The standard format can be modified by a call of one or more format setting procedures. On entry to an Algol program a standard format set is automatically invoked: the format procedure has the effect of changing one format parameter and leaving the remainder unaltered. A format call determines the format for a particular channel for all future use of that channel until it is modified in the dynamic sense by a further format procedure call referring to the same channel.

The user is provided with facilities to store and restore the format currently existent.

Entry to Algol Program

On entry to an Algol program standard format is automatically invoked. In addition all buffers are emptied, pointers set to zero and procedure store format is called for all channels.

The procedures INNEWLINE and OUTNEWLINE are automatically invoked at the last possible moment.

Format Procedures

The three procedures that follow each determine the conversion of a value to a total of W characters.

B19 procedure FLOAT (Q,W); value Q,W; integer Q,W;

comment This procedure determines that real values are converted with W-5 decimal digits preceded by decimal point and sign and followed by exponent base 10 symbol, sign and two digits;

B20 procedure ALIGNED (Q,W,J); value Q,W,J; integer Q,W,J;

comment This procedure determines that real numbers are converted with J digits before and W-J-2 digits after the decimal point J. Up to J-1 leading consecutive zeros will be replaced by space. The sign immediately precedes the first digit printed;

```
B21     procedure INTEGER (Q,W); value Q,W; integer Q,W;

        comment   This procedure determines that
                  integers are converted with W-1 digits
                  (including up to W-2 leading zeros
                  replaced by space) and with the first
                  digit printed preceded by sign;
```

Optional Format Procedures

```
E22     procedure SIGN (Q,plus,minus); value Q;
                  integer Q; string plus, minus;

        comment   SIGN affects the symbol transmitted
                  by all OUT procedures and causes the
                  string plus to be transmitted for
                  positive numbers and minus for
                  negative numbers.  Note that the
                  standard format for sign is given
                  by the procedure call

                  SIGN (Q,‘␣’,‘-’);

E23     procedure PRECEDE (Q,string); value Q, integer Q;
                  string string;

        comment   This procedure determines that the
                  named string precedes all OUT;
                  conversion.  Note that the standard
                  format is given by PRECEDE (‘’);

E24     procedure BOOLEAN (Q,TRUE,FALSE); value Q;
                  integer Q; string TRUE, FALSE;

        comment   The strings determine the strings
                  for OUT conversion of Boolean
                  values.  The standard format is given
                  by BOOLEAN (Q,‘1’,‘0’);

E25     procedure SPACETERMINATE (Q,K); value Q,K;
                  integer Q,K;

        comment   This procedure determines that K
                  (K ≥ 1) successive space characters
                  are treated as a terminator provided
                  a valid character has first been
                  converted.  The standard convention
                  is K=2;
```

E26 **procedure** TITLE (Q,string); **value** Q; **integer** Q; **string** string;

comment This procedure specifies a string which is subsequently available as the page title on that channel. The standard format is given by TITLE (Q,´`);

E27 **procedure** STOREFORMAT (Q); **value** Q; **integer** Q;

comment This procedure causes a copy of the format set for channel Q to be preserved for use by a subsequent call of RESTOREFORMAT. A further call of STOREFORMAT for the same channel will replace the value stored previously;

E28 **procedure** RESTOREFORMAT (Q); **value** Q; **integer** Q;

comment This procedure causes all formats for channel Q to be reset to the values preserved by a previous call of storeformat.

The procedure does not affect the value stored by STOREFORMAT.

N.B. If this procedure is called without a previous call for store format, the effect will be undefined;

NOTES

1. The procedures are marked with either a B or an E standing for Basic and Extended respectively. The I/O determined by the Basic set is considered the minimum feasible set while the extended set is considered of wider use and application.

2. This report deals with symbols termed characters and not with basic symbols as defined in the Algol report. A character is used to describe any distinguishable mark that is associated with a channel. The set of characters for any implementation is defined in the implementers manual together with the corresponding table of integers. It is possible that some channels may be limited to a subset of the set of characters. Each character represents one printing position. In procedure B19 it is assumed that the symbol chosen to represent the Algol basic symbol, base 10, is a single character. The effect of using a representation which results in hardware capacity being exceeded is undefined.

3. A table for use in representing Algol texts in terms of the ECMA 6-bit standard is given in ECMA/TC5/65/6.

4. The period of existence, in the dynamic sense, of the buffer is not defined in this document. Manifestly each implementation may define buffers which may not remain in existence throughout the activation of an Algol program and may choose to permit the programmer to specify specifically the duration of their existence. Such processes have been deliberately left out of this document because it is thought impossible to define such a process without introducing new concepts to ECMALGOL.