

INTERNATIONAL COMPUTERS AND TABULATORS LIMITED

PREPARING A COMPLETE
PROGRAM FOR ATLAS I.

TL 1254

Atlas Support Group,
21 Portland Place,
London, W.1.

CS 460

March 1966

Contents

	<u>Page</u>
Notes	1
1. Atlas Jobs	2
2. Documents	3
3. Document Headings and Titles	4
3.1 Headings	4
3.2 Titles	4
3.3 Rules for Title Preparation	5
4. The Input and Output Sections of the Job Description	7
4.1 The Input Section	7
4.2 The Output Section	7
4.3 Output: General Notes	9
5. A Complete Job Description	10
6. The Magnetic Tape Section of the Job Description	11
6.1 Single Tapes	12
6.2 Deck Allocation	12
6.3 Half-Inch Tapes	13
6.4 Comments in Job Description	13
7. Time Estimates for a Job	15
7.1 Computing Time	15
7.2 Execution Time	16
8. Store Allocation	17
9. Job Description Format	18
9.1 Order of Sections	18
9.2 Case Changes	18
9.3 Backspace	18
10. Composite Documents	19
10.1 Job Description combined with Program	19
10.2 Job Description combined with Data Document	20

	<u>Page</u>
11. Tape and Card Markers	22
11.1 The Tape Markers ***Z,C,T,A and I	22
11.2 The Binary Tape Markers ***B,E and F	23
11.3 The Tape Marker ***P	24
11.4 Card Markers	24
12. Input and Output using private magnetic tapes	25
12.1 Input Peripheral to Tape	25
12.2 Reading a Tape by Program	26
12.3 Output to Tape from Object Program	27
12.4 Output from Tape to Peripheral	27
12.5 Format of Magnetic Tapes	28
13. Job Description Parameter	30

Notes

1. Any changes in the information given here will be reported in the Atlas 1 System Programs Bulletins.
2. Some details of operation vary between London, Manchester and Chilton; programmers are advised to contact the installation concerned for information on the current system.

1. Atlas Jobs

Each run of a program on Atlas is known as a job; it may range from a small job, for which there is no data outside the program itself, to a large job requiring several batches of data, possibly arriving on different media, e.g. punched cards and paper tape.

The various parts of a job may be submitted separately to the computer each on one deck of cards or length of paper tape, or two or more parts may be combined on a single deck or tape. In any case each part must be properly identified for the computer and for this purpose the concept of a 'document' has been introduced.

2. Documents

A document is a self-contained section of information presented continuously to the computer through one input channel. Typical examples of a document are a collection of data on a length of paper tape or the program itself.

Each document carries at its head suitable identifying information as detailed in 3.1. The end of a document is indicated by an 'end of document marker' which usually consists of the characters ***Z on a new line, or 7, 8 punching in the first column of a new card which follows the last record of information of the document.

By means of the identifying information the Supervisor prepares a list of documents as they are accepted in the store, and it also keeps a list of jobs for which further documents are awaited. A job may require several documents and only when all these have been supplied can execution begin. The Supervisor therefore checks the appearance of each document; when all have been entered work on the job may commence. Documents for a job may thus be fed to the computer in any order.

3. Document Heading and Titles

Every document is preceded by the identifying information mentioned above. This consists of a heading and the title of the document.

3.1 Headings

The heading indicates which type of document follows and must be one of a standard list. The most common types of heading are as follows:-

- a) COMPILER (which is followed on the same line or card, after one or more spaces, by the name of a program language). The document following this heading is a program in the stated language. Available languages include Atlas Basic Language (ABL), Extended Mercury Autocode (EMA), ALGOL and HARTRAN (for Fortran).

For an Atlas Basic Language Program the heading will be

COMPILER ABL

- b) DATA

The document following consists of data required by a program.

- c) JOB

The following document is a request to the computer to execute a job and gives relevant facts about it.

The last type of document is called a 'job description'. It gives, for example, a list of all other documents required for the job and a list of output streams the program will produce. It is described in detail in sections 4 to 9 below.

3.2 Titles

The title of a document consists of one line (or one punched card) immediately following the heading. It may be composed of any combination of characters obeying the rules of section 3.3 below.

The prime consideration is, however, that it should be unique among all the documents stored in the computer at the same time. This is obviously made essential by the time-sharing facilities of Atlas, to avoid confusion between documents intended for different jobs.

3.2 Titles (Continued)

A document will thus usually take one of three forms exemplified by the following, the second line of each document being its title:-

JOB F6479,SMITH;I.C.T. ----- ----- (Job Description) ----- ----- ***Z	COMPILER ABL F1, SURVEY PROGRAM ----- ----- (Program) ----- ----- ***Z	DATA F6479 BEETLE SURVEY ----- ----- (Data) ----- ----- ***Z
<u>Job Description</u>	<u>Program Document</u>	<u>Data Document</u>

3.3 Rules for Title Preparation

Besides being unique, document titles must obey the following rules:-

- a) The title must begin with an identification of the person or organization which originated the document: normally this will take the form of an account number or name. For example, documents prepared for the I.C.T. Atlas Computing Service are identified by a letter F followed by an account number.
- b) The number of characters must not exceed 80.
- c) 'Backspace' must not be used.
- d) A title must not contain three successive asterisks (***) .
- e) Titles must not begin with the work 'TAPE'.

Furthermore:

- a) The characters of a title are read in and stored in Supervisor records in Atlas internal code in the normal way, but the shift characters, 04, 05, 06, 07 are subsequently removed.

This means for example that on the Flexowriter the titles

[Tape]

and

1TAPE3

are identical. For the same reason a length of run-out appearing on tape in the midst of a title would not become part of the title.

3.3 Rules for Title Preparation (Continued)

- b) Any number of consecutive spaces and tabs are stored as one space.
- c) Erases do not become part of the title.
- d) Initial commas, spaces and full stops are ignored.

Documents used in the same job need not have related titles.

The job title itself normally contains the name and abbreviated address for the return of the results, but this is not necessary in the titles of data and program documents.

If necessary the title of a data document or job description (but not a program document) may be on the same line, or card, as the heading, provided sufficient room remains to accommodate all the title on that line or card. In this case the heading and title must be separated by 'comma', 'space' (one or more), or 'tab'.

4. The Input and Output Sections of the Job Description

After the heading and title, a job description is divided into various sections, each one describing a particular aspect of the job, e.g. input documents, store used and so on. These sections may be assembled in any order and are dealt with individually below in sections 4.1, 4.2, 6, 7 and 8.

4.1 The Input Section

This section begins with the word

INPUT

which is followed by the titles of data documents used in the job, each preceded by the number by which the program refers to them. These numbers must be in the range 0 to 15. The program document itself usually is given number 0, but is in fact always taken to be the lowest numbered input document. Thus, if a program operates on two data documents which it refers to as inputs one and two respectively, the job description would contain

```
INPUT
0 (Title of Program)
1 (Title of Data 1)
2 (Title of Data 2)
```

To take a concrete example:-

```
INPUT
0 F1, SURVEY PROGRAM
1 F6479/2 BEETLE SURVEY DATA/66
2 F6479 BEETLE POPULATION 1965
```

The data document "F6479/2 BEETLE SURVEY DATA/66" could then be selected by the programmer by using a "Select Input" instruction (the 1050 Extracode in ABL)

Data may be placed on the same tape as the program, where it becomes a part of the lowest numbered input stream. Such data must come between the end of the program and the end-of-document marker that terminated the program stream. If an instruction is not used to select a given input stream, the lowest numbered stream is assumed and one obtains the data which followed the program.

The input section may also contain a reference to a magnetic tape on which an especially large document has previously been stored. This way of handling a large amount of input is explained in section 12.1.

4.2 The Output Section

This section of the job description specifies the type of peripherals to be used for output. Possible types of equipment are:-

4.2 The Output Section (Continued)

LINE PRINTER
 SEVEN HOLE PUNCH
 CARDS
 FIVE HOLE PUNCH
 ANY
 TAPE

Here CARDS means the card punch. Each Atlas Installation will specify which types of equipment may be used for output ANY e.g. ANY may produce output on LINE PRINTER or SEVEN HOLE PUNCH. TAPE is used when a private magnetic tape is called for to hold an especially large amount of output (see 12.4).

The output section begins with the word

OUTPUT

which is followed by a list of output mechanisms, each preceded by a programmer's number in the range 0 to 15. For example one might have

OUTPUT
 2 ANY
 1 CARDS
 0 LINE PRINTER

In this case, in order to send output to the card punch, the programmer would first have to select this form of output by a "Select Output" instruction. (The 1060 extracode in ABL.)

A request will be made to the operator to mount special stationery for a given output stream if an asterisk is placed in front of the word LINE PRINTER. Thus if output stream 3 is to be printed on special stationery, the output section should contain

*LINE PRINTER m LINES

The type of equipment should normally be followed by a limit on the amount of output, specified as so many lines. One line is the output produced by one use of the 1065 (end current record) or the 1067 (output one record) extracodes. Thus one line means one printed or blank line on 5 or 7 track tape and on the line printer, or one punched card. One writes

OUTPUT
 0 LINE PRINTER m LINES

The maximum amount of output may also be specified as n BLOCKS. A block contains 4096 characters. The number of characters allowed for must in general be larger than the number actually printed or punched. On the average each line output to punched paper tape or line printer requires an additional six characters (the maximum possible is 8) to be allowed for. Exactly 8 additional characters must be allowed for to punch one card (making 88 in all). Furthermore each use of the 1065 extracode (end this record) to produce a blank line generates 8 characters to be held in the output well.

4.2 The Output Section (Continued)

If the number of blocks is omitted, one block only is allotted, and if the whole output section is omitted

```
OUTPUT
O ANY 1 BLOCK
```

is understood.

4.3 Output: General Notes

- a) Output 0 Output 0 is used by the Supervisor and some compilers, but is still available for normal output from the program. It is on this output that such information as the number of instructions done in compiling and executing the program, the number of store blocks in use when the program ended, the number of blocks accumulated for each 'stream' and other such items are printed.

It is also used for fault information if the program goes wrong.

If no output 0 is mentioned in the output section of the job description

```
O ANY 1 BLOCK
```

is assumed.

Similarly, if no output stream is selected by a "select output" instruction, any subsequent output will go to output 0.

- b) Atlas can readily accept two or more streams of output from a program for the same type of equipment, even though only one such equipment may exist. The streams are accumulated independently within the computer and eventually output one after another.

In fact all output is accumulated and none will be printed until all computing has ceased unless a "Break Output" instruction (the 1071 extracode in ABL) is used to break output stream n. In this case all of the output stream n accumulated so far will be sent to the peripheral.

In either case the output information with programmer's number n will always be preceded by

```
Supervisory Number / Date. Time
OUTPUT n
(Title of Job)
```

The last line of output gives the number of blocks sent to that output.

5. A Complete Job Description

We are now in a position to give an example of a complete job description and for the sake of illustration we include the documents of the job.

JOB F64, J. Smith, I.C.T. London, METALS	Job Title
INPUT 0 F64, ANALYSIS PROGRAM 1 F64/A, IRON CONTENT 2 F64/B, COPPER CONTENT	Program Title } Data Tapes
OUTPUT 0 LINE PRINTER 1 SEVEN HOLE PUNCH 3 BLOCKS ***Z	} output } streams End-of-tape marker

Job Description

```

COMPILER ABL
F64, ANALYSIS PROGRAM
-----
(Program)
-----
***Z
    
```

Program Document

```

DATA
F64/A, IRON CONTENT
-----
(Data)
-----
***Z
    
```

Data 1

```

DATA
F64/B, COPPER CONTENT
-----
(Data)
-----
***Z
    
```

Data 2

6. The Magnetic Tape Section of the Job Description

Magnetic tapes are used with Atlas in two ways. Firstly, they are used by the Supervisor for such purposes as storing input and output. These are called System Tapes, and under normal circumstances need not concern the programmer at all. Their operation is quite automatic (see also section 12.3).

Secondly, the programmer may use magnetic tapes in his program either:

a) For private input and output purposes.

or

b) By magnetic tape extracodes.

The use of magnetic tapes for private input and output purposes will normally only be necessary if there is a very large amount of input or output. Full details of the way in which such tapes are employed are given in section 12.

The most common use of magnetic tapes is by extracodes within the program. The tapes required may be mounted on a tape mechanism (a 'deck') before running the program or during the actual execution.

The tapes mounted before the job begins must be listed in the Job Description. However, in programs for which the number of tapes is not easily predictable, it is possible to get tapes mounted which are not listed in the Job Description.

Sections 6.1 and 6.2 show how to list tapes in the Job Description. The extracodes for mounting tapes are described in Chapter 9 of the ABL Programming Manual (CS 348A). This chapter also gives full information on the magnetic tape system; here we will mention only the following:

- a) Information is stored on magnetic tapes in blocks of 512 48-bit words.
- b) The first block is known as block 0, and is not available to the programmer. Block 0 contains the serial number of the tape, and the title if the tape has one. The title of the tape must obey the rules of document titles given in 3.2; the first 30 characters of the title must identify the tape uniquely. Only those 30 characters are printed by the Supervisor when calling for the tape, and only thirty characters are stored and checked.
- c) In preparation for use each tape is mounted on a tape deck by an operator who receives instructions for the purpose from the Supervisor. When mounted, each tape is positioned by the Supervisor at the beginning of block 1, the first block available for storing information.

6.1 Single Tapes

Each tape required for a job is specified in the job description by 2 lines of printing:- a heading, and a description. The heading is one of three:-

- | | |
|-------------|---|
| TAPE | - a tape belonging to the user and already having a title; |
| TAPE NEW | - an untitled tape, not previously belonging to the user, to be titled and kept when the job is over; |
| TAPE COMMON | - an untitled working tape to be retained by the system when the job is over. |

The description consists basically of the programmer's number (in the range 0 to 99), and the title of the tape:

```
21      F1000,  LONDON SALES,  1965
```

This format applies with headings TAPE and TAPE NEW. That for TAPE COMMON is described later in this section.

The programmer's number, 21 in the above example, is the number by which the tape will be referred to in the program.

The title in the description is the complete title stored in block 0 of the tape. Under a TAPE heading, this will be used to check that the correct tape has been mounted; under TAPE NEW, the title will be written to block 0 when the tape is mounted. In either case the Supervisor refers to the tape by the first 30 characters of its title.

The description for TAPE COMMON consists of a programmer's number only, e.g.

```
TAPE COMMON
36
```

All the programmer's numbers in the Job Description must, of course, be different. A programmer's number must lie in the range 0 to 99.

6.2 Deck Allocation

As stated above, each single tape that is given a programmer's number in the job description will be mounted on a separate deck before the start of the job. These decks will then be available to the programmer throughout the course of the job. He can by extracodes cause any of the original tapes to be unloaded and new ones to be mounted in their place.

6.2 Deck Allocation (Continued)

If however, at some stage in the program he requires more decks than will be allocated to him in this fashion, he must mention the number he will need in the job description. This is done by writing:-

DECKS d

where d is the maximum number of mechanisms that will be in use at any one time. Thus a programmer requiring 3 tapes to be mounted at some stage in the program but only 1 at the beginning, would give the single tape a programmer's number and put in his Job Descriptor:-

DECKS 3

6.3 Half-Inch Tapes

At those installations capable of handling half-inch (IBM-compatible) magnetic tape, each tape required must be listed in the Job Description by means of a TAPE IBM heading, followed by a new line by a description, consisting of the programmer's number (in the range 0 to 126) and the title of the tape, e.g.

TAPE IBM
42 WINCHESTER SALES

The deck allocation is treated in the same way as for one inch tape; each tape used should have a different programmer's number.

6.4 * in Job Description

If the character * appears in the specification of a tape title in a job description or object program, the following characters on the line will be printed as usual but not checked when comparing the title with that on tape. Hence comments can follow the character *.

For example:

TAPE 1 SYSTEM TAPE 5* USE RBP/10

requires a tape of title SYSTEM TAPE 5.

TAPE 2 SPECIAL COMPILERS* WRITE PERMIT

requires a tape of title SPECIAL COMPILERS.

The following restrictions apply:

1. The character * must follow immediately after the title proper - intervening spaces are not permitted.
2. If used with the facility TAPE NEW in order to title a tape, the entire line including asterisk and comments will be written to tape.

6.4 * in Job Description (Continued)

3. If a tape called by such a sequence is pre-mounted it will not be accepted as required tape. The deck on which it is mounted should be disengaged by the operator and re-engaged.
4. No comment is permissible in the TAPE COMMON statement, since this specifies no tape title. It must be assumed by the operators that common and new tapes are to be mounted with writing permitted.
5. A maximum of 31 characters (including the asterisk) will be printed on the tape operators teleprinter.
6. Although not checked, the characters following * form part of the title written by the 1014 extracode, or read by the 1015 extracode.

7. Time Estimates for a Job

7.1 Computing Time

Since each program on Atlas will normally be time-sharing with others there can be no direct control of an individual program by an operator. Thus, there is no means by which an operator may tell if a faulty program has entered an infinite loop and is thereby wasting machine time.

To overcome this problem a time limit for a program is placed in the job description and if the program exceeds this limit it is stopped by the computer.

The limit appears in the job description as:-

COMPUTING p.q SECONDS (or MINUTES, or HOURS)

where p.q is a decimal number. This is found by allowing about 3 microseconds to each instruction obeyed in the program, and adding to this the expected compiling time.

Alternatively the limit on computing time may be specified as

COMPUTING m INSTRUCTIONS

In this context, one 'instruction' means one instruction interrupt, equal to 2048 basic instructions obeyed. In fact the Supervisor actually times the program in terms of these units of 2048 instructions. Conversion from estimates given in terms of seconds, minutes or hours is made on the basis of 10,000 'interrupts' per minute. For the purpose of this calculation, each multiplication instruction is counted as 2 instructions, and each division instruction as 4.

The table below will help programmers to estimate the compiling time required for programs written in various languages.

<u>Language</u>	<u>Instruction interrupts during Compiling</u>
ABL	about 1 per line of program
Algol	$3\frac{1}{2}$ to 7 " " " "
EMA	$2\frac{1}{2}$ to 5 " " " "
Hartran	about 40 " " " "

For example, a program requiring at most three million instructions, and having a compiling time of one and a half seconds would have the entry

COMPUTING 10.5 SECONDS

or

COMPUTING 1750 INSTRUCTIONS

If the COMPUTING section is omitted from the Job Description a standard allowance is made as follows:-

at London : 15 secs.
 at MUSE : 60 secs.
 at S.R.C. : 90 secs.

7.2 Execution Time

If a program uses magnetic tapes it may be held up at some stage while a block of information is brought from tape to store, a time of 64 milliseconds. This wait can be eliminated in many cases by calling for a block 64 m.s. before it is needed, or by using a sufficiently large variable length transfer buffer or by resorting to branching. Some tape waiting time, however, may be inevitable and if it is likely to occur it must be shown in the description.

This is done by the heading:-

EXECUTION p.q SECONDS (or MINUTES or HOURS)

where the time estimate in this case is found by adding an upper limit for the tape waiting time to the COMPUTING time.

For example, if the program quoted in 7.1 above was expected to be held up at most 200 times the job description would include:-

COMPUTING 10.5 SECONDS
EXECUTION 20.3 SECONDS

If the EXECUTION section is omitted from the job description the execution time is taken to be one minute more than the computing time.

8. Store Allocation

An estimate of the amount of store needed by the program is also required by Atlas to prevent a faulty program from monopolising the store by producing a large amount of useless information. This is done by the Job Description entry:-

STORE S1/S2 BLOCKS

where S1 is the maximum amount of store used by the object program, and S2 is the store used during compiling. S1 and S2 are measured in blocks of 512 words; no distinction is made between core and drum store. The word 'BLOCKS' may be omitted if desired.

An alternative entry is:-

STORE S BLOCKS

where S is the store used by the object program. In this case a standard allowance is made for S2; the allowance varies among the installations, and also depends upon the compiler specified. The blocks used by the compiler are not counted against the object program unless the compiler is retained in the main store after the program is entered.

If the STORE entry is omitted entirely from the Job Description, a standard allowance for object programs is made as follows:-

at London : 20 blocks
 at MUSE : 32 blocks
 at S.R.C. : 32 blocks

If the estimate for store is exceeded at any time, the program is stopped by the computer. One extra block should be allowed for each input and output stream.

9. Job Description Format

9.1 Order of Sections

Separate sections of the job description may be listed in any order. For instance the OUPUT section could precede the INPUT section. STORE could be followed by COMPUTING. However, to make it easier to read it is perhaps advisable to keep to a fixed order, for example the order in which the sections have been introduced. That is:-

INPUT
OUTPUT
MAGNETIC TAPES (this is not an actual section heading)
DECKS
COMPUTING
EXECUTION
STORE

9.2 Case Changes

Throughout this chapter the sections of the job description have been written in capital or upper case letters since these are common to all forms of input. Changes of case are ignored however, and if the job description is on seven-hole tape, lower case letters could be used.

9.3 Backspace

Throughout the job description 'backspace' is an illegal character.

10. Composite Documents10.1 Job Description Combined with Program

A job description may be combined with a program to form one composite document. In this case the last item of the job description will be followed, not by an end of tape marker (**Z) but the program heading:-

COMPILER (Program Language)

and then by the program itself. No further title will be necessary since the composite document takes the title of the job description. This will be the usual form taken by small programs which are only run once.

If this procedure is adopted, no input zero is mentioned in the INPUT section of the job description and the computer will compile and execute the program immediately following the job description. If there are no separate data documents the INPUT section may be omitted completely.

The examples given below illustrate these facilities.

Example 1

```
JOB
F196, J.BLOGGS:  NUMBER FREQUENCY
INPUT
1 F196--TABULATED DATA
OUTPUT
O LINE PRINTER
COMPILER  EMA
-----
-----
-----
-----
***Z
```

Job Description/Program

```
DATA
F196--TABULATED DATA
-----
-----
-----
-----
-----
-----
-----
***Z
```

Data Document

In this example the job required one other data document so the INPUT section must be included.

Example 2

```
JOB
F324/1, W. BROWN, ICT LONDON, PRIME NUMBERS
OUTPUT
O LINE PRINTER
1 CARDS 2 BLOCKS
COMPILER ALGOL
-----
-----
-----
-----
***Z
```

10.1 Job Description Combined with Program (Continued)

Here no further information is required and the INPUT section has been omitted.

10.2 Job Description Combined with Data Document

It is also possible to combine the job description with a data document. This is particularly useful when the same program is to be run more than once, using different data each time. In this case the INPUT section of the job description must include:-

```
SELF = n
```

where *n* is the programmer's number for the data which follows on the same document. The program itself is specified as the lowest numbered input stream, in the same way as when the job description is a separate document, and the last item of the job description is followed by the heading

```
DATA
```

and then the data itself. No title is needed to identify the data.

For example, consider a wage calculation carried out each month using one fixed set of data, say a list of P.A.Y.E. codes and a second set of data consisting of a list of hours worked by each member of the staff. The second set of data would, of course, vary from month to month and could be combined with the job description while using the same program and P.A.Y.E. code tapes.

The program and data documents would be:-

COMPILER ABL
F900, WAGE CALCULATION

(Program)

***Z

DATA
F900, P.A.Y.E. CODES

(Data)

***Z

The job description could then be combined with the second data document thus:-

10.2 Job Description Combined with Data Document (Continued)

JOB

F900, J. SMITH LTD: WAGES OCTOBER 1965

INPUT

0 F900, WAGE CALCULATION

1 F900, P.A.Y.E. CODES

SELF = 2

OUTPUT

0 LINE PRINTER 5000 LINES

1 CARDS 2000 LINES

DATA

(Data)

***Z

11. Tape and Card Markers

So far only the marker ***Z and the 7, 8, Z punching have been considered as an end to a document. These are in fact the most common markers but there are others which are dealt with below.

On punched tape all the markers consist of *** followed by a single letter. On cards *** is not acceptable, and is replaced by punching the 7 and 8 positions in the first column of the card and the letter in the last column. The other 78 columns can contain anything at all. If the last column contains anything but a terminator letter, or is blank, the punching 7, 8, Z is assumed.

11.1 The Tape Markers ***Z, C, T, A and I

- a) ***Z indicates not only the end of a document but effectively also the physical end of the tape. The peripheral equipment concerned is disengaged by the computer and when re-engaged by the operator a new document will be read.
- b) ***C indicates that the end of the current document has been reached but that another follows on the same paper tape. The end of the document is noted by the computer and reading is continued for the next document without interruption.
- c) ***T indicates a temporary stop. When this marker is encountered the peripheral equipment is disengaged by the computer and when next engaged by the operator a continuation of the same document is read. Thus, if a document consists of two tapes the first part can be ended with a ***T. When this has been fed to the computer the second part is read by the same peripheral with no document heading and the computer will treat the two parts as one document.
- d) ***A is used only by a machine operator and is an instruction to the computer to abandon the previous incomplete document and disengage the equipment. It is required if part of a document is damaged before input is complete and the operator requires the computer to disregard the information it has already received.

On re-engaging a tape-reader after ***A, T or Z, the next character is lost. On 5-track tape at least two figure shift characters should be punched following the marker, and before the next carriage return and line feed; on a 7-track tape, the marker should be terminated by either upper case and newline, or two newlines.

11.1 The Tape Markers ***Z, C, T, A and I (Continued)

- e) ***I is used only at the end of binary data following ***E. When ***I is recognised, binary reading ceases and the rest of the data undergoes normal conversion to internal code.

11.2 The Binary Tape Markers ***B, E and F

- a) ***B. When this marker is encountered the computer reads the information following on the same document in binary, instead of internal code, to the physical end of the tape. There is no test for end of tape markers. The last 2 or 4 characters from a paper tape read in this fashion will be overwritten in the store by the 12 bit character 0707 (octal), which replaces any spurious characters generated as the end of the tape passes through the reader (this does not apply to punched cards).
- b) ***F causes the mechanism to be disengaged. When it is next engaged by the operator the new paper tape is read to its physical end in binary. Thus ***F combines the effects of ***T and ***B.
- c) ***E causes the input following on the same tape to be read in binary, but a check is made for ***A, Z, C, I, P and T. When one of these is encountered, binary reading ceases and the appropriate action is taken.

If the end of a reel of tape is encountered after ***E and before any of these *** sequences, the last 2 or 4 characters on the tape will be denoted by the 12-bit character 0707 (octal); the next tape will then be read as a continuation of the same binary document. When ***A, Z, C, I, P or T is encountered, it is itself stored in binary. During card input, following 7, 8 E no card bearing an end-of-document marker is stored in binary.

Note that if it is required to read to the physical end of a 7-track tape it is necessary to precede the ***B, E or F by ***P, as described below. If this is not done the tape may be rejected because of a spurious tape parity fault when the end of tape passes through the tape reader.

If code conversion is being applied to paper tape input, any marker sequence in the input stream will be overwritten with a string of internal code 07; during binary input, marker sequences are stored (in binary) before the appropriate action is taken.

11.2 The Binary Tape Markers ***B, E and F (Continued)

When reading punched cards an alternative way of causing the card to be read in binary is available. If the first column of a card is a standard code the contents of the card are converted to internal code. If the first column is punched in rows 7 and 9 (irrespective of any other punchings), the card is assumed to be in binary and is stored as such. Naturally, after a 7, 8B, E or F has been read all cards are taken as binary regardless of their first column.

11.3 The Tape Marker ***P

When reading 7-track paper tape in internal code or in binary the Supervisor normally checks the parity of each character (an odd number of holes for correct parity) and rejects the document if a character with wrong parity (blank tape for example) is encountered. This parity checking may be suppressed by punching ***P, but it will be restored again at the end of the given document. If the input is binary and ***P has been punched, wrong parity characters will be recorded as punched, but if the input is internal code they will be replaced by the fault character 7.7 (inner set).

Note that to suppress parity checking on binary input it is necessary for ***P to appear before ***B, E or F; ***P after ***B, E or F will not be recognised.

11.4 Card Markers

The same markers are available for use with cards, but with the *** replaced by 7, 8 punching in the first column of the card and with the letter in the last column.

Marker cards are stored in the input stream only after B or F, when no terminators are recognised and the reader proceeds until its input hopper is empty.

12. Input and Output using Private Magnetic Tapes

All the documents fed to peripherals for input are stored by the computer on a magnetic tape known as the system input tape. They are then brought into the core store as required. If a program has a large amount of input it may be in the programmer's interest, and in the interest of the efficiency of the computer, to transfer this input to a private tape before starting his program, instead of using the system input tape.

Output is normally accumulated on the system output tape before being sent to the required peripherals. If a program involves extensive output this may be written to a private magnetic tape instead, and later output upon requesting the Supervisor to do so by means of a special document fed in through one of the peripherals after the job is completed.

Both these facilities are dealt with in this section.

12.1 Input Peripherals to Tape

A document can be copied from an input peripheral to a private magnetic tape, by heading the document

```
COPY TAPE b                PRIVATE INPUT b
  [Title of tape]         or         [Title of tape]
```

where b is the starting section on tape.

If a new tape is required, the heading is:-

```
COPY TAPE FREE
  [Title to be written on tape]
```

or

```
PRIVATE INPUT FREE
  [Title to be written on tape]
```

In this case recording will commence at section 1.

The document following will be copied to the named tape; the above two lines of heading will be overwritten with run-out code, and the line following the heading will be ignored on cards or 7-track tape. Hence information read subsequently will start after the heading (from 5-track tape) or two lines after the heading (7-track tape and cards). Note that this heading replaces the conventional headings DATA, COMPILER etc. followed by document title. The document as recorded on tape has no title recorded with it.

Recording will be terminated at the end of the document i.e. at warning codes Z, C or A or at end of tape following warning codes B, or F. Normal use may be made of other warning symbols. When the end of the document is reached, the tape is rewound and disengaged, and a message is printed on the chief operators teleprinter as follows

12.1 Input Peripherals of Tape (Continued)

[Title of tape]

COPYING COMPLETED TO BLOCK n

where n is the block label of the next available section on the tape, in octal.

If subsequently another document is recorded starting at section n on the same tape, it may be treated as an addition to the original document, and the whole tape can be read by a program as a single input document. Hence long documents can be broken down for input into any convenient size by insertion of end of document sequences; the subdivision can be decided by operators bearing in mind the expected machine load and the current mean time between failures. A failure in the course of recording a document requires restarting at the beginning of the document; the restart is under operator control, no record of the operation being retained in the restart block.

12.2 Reading a tape by program

A document recorded on tape by the above method may be used subsequently in a program as an input stream by including the following statements in the job description:-

```
INPUT TAPE a/b/O
n [Any convenient title]
TAPE
a [Title of tape]
```

a is the logical tape number, quoted both in the input statement and the tape statement.

b is the address of the section on tape where reading is to commence.

n is the logical input stream number. It is followed by any title the user may wish; this title is not checked on tape, and is effectively ignored by the Supervisor. It must, however, be present in the job description for the sake of uniformity with direct input streams.

The input stream so defined may be read by means of input extracodes. Reading will commence at the first record on section b; it is thus possible to start reading part way through an input document, which may be useful in the event of a long input stream. Note that any heading is not decoded or spaced over; in particular no notice will be taken of any compiler statement occurring in this input document. Normally a compiler is called to compile the lowest numbered input stream, which is always regarded as the program document. If the lowest numbered stream specified in the Job Description is the Private Tape, the compiler must be specified in the last line of the Job Description. If no compiler statement appears, the ABL compiler will be called to start the job.

12.2 Reading a tape by program (Continued)

The private tape used in the above way may also be used by the program as a conventional tape using tape extracodes. This would not normally be done until the input stream has been completely read, for the input extracodes will reposition the tape whenever a new block of input is required.

12.3 Output to Tape from an Object Program

An output stream may be written to a magnetic tape rather than output immediately by use of the output statement:

```
OUTPUT n TAPE b / [Final destination] [amount]
[Title of tape]
```

or

```
OUTPUT n TAPE FREE / [Final destination] [amount]
[Title of tape]
```

In the former case, output stream n will be written to the named tape starting at section b. In the latter case, a new tape is used and the title specified is written to the tape; output is recorded from block 1 onwards.

The final destination and amount of output are specified as for normal output streams.

The tape used for output stream n may also be used as private tape 110+n in the object program; normally such use will precede use of the tape for output. The output extracodes assume the tape to be positioned correctly whenever a new block of output is generated, and hence, if used privately, the tape must be repositioned by the user before output. Only one output stream can be directed to any one tape. The tape so generated may be used as input data for a subsequent program, using the facility for using a tape as an input stream. In this case, the first records on the tape will comprise:

```
OUTPUT n
[Title of job which created this output]
```

and these will be the first records read by a program.

If an output stream destined for magnetic tape is broken into sections by use of the break output extracode, a new section on tape will be started after each break output. Each section of output will be printed subsequently as a separate document.

12.4 Output from tape to peripheral

A private output tape can be output to any peripheral by use of a standard program. The heading is supplied by the user. Alternative sections at the end of the program select the particular output device to be used. Details of this program may be obtained from the installation concerned.

12.4 Output from tape to peripheral (Continued)

This job will halt until the output device is idle and will, then initiate output from the named tape. The device will remain booked until the end of the current document, when any other normal output which has been accumulated will be output.

The output is terminated by the line

ENDS AT BLOCK B

where B is the next block address on tape in octal. If the output has been broken, each portion will be output independently. The next portion may be output subsequently by printing another document starting at section B from the same tape.

No record of this output is recorded in the restart block, and hence, in the event of a machine breakdown, the job should be read in again to restart printing. Printing may be resumed at any block label B.

The *LINE PRINTER facility is available for printing output from private tapes on special line-printer stationery.

A magnetic tape produced by direct input from a peripheral may be output directly onto any desired equipment. The formats of input and output tapes are identical.

12.5 Format of Magnetic Tapes

Input or output tapes can, if required, be created by an object program using tape extracodes. The format of each block of 512 words is as follows:-

Half-word 0:	digits 0->23 next block address on tape. If zero, indicates end of document (not necessarily conversely)
Half-word 0.4:	digit 7 = 1 if last block of document, 0 otherwise Remaining digits irrelevant.
Half-words 1 to 7.4:	Irrelevant
Half-words 8 to 511.4:	Records in internal code or binary. Each record comprises a half-word record separator followed by the record.

12.5 Format of Magnetic Tapes (Continued)

The record separator contains:

Digit 0:	1 if binary, 0 if internal code.
Digit 1:	1 if continuation of incomplete record at start of block, 0 otherwise.
Digit 2:	1 if start of incomplete record at end of block, 0 otherwise.
Digits 12->23	Number of 6-bit characters in record (including carriage control character).

The information comprises binary characters (2 per half-word) or internal code characters (4 per half-word) terminated by a 6-bit internal code carriage control character, indicating the following:-

Digit 0:	1 if paper throw, 0 otherwise.
Digit 1:	1 if carriage return, 0 otherwise.
Digits 2->5	Number of line feeds <u>or</u> channel for paper throw.

At the end of a block, any incomplete record is ended without any carriage control character but with digit 1 set to 1 in the record separator. There follows at least one zero half-word indicating end of information. The record is resumed at word 8 of the next block; word 8 contains a count of the remaining characters with digit 2 set to 1.

Internal code records always start in inner set internal code.

13. Job Description Parameter

In a job where different parts of a standard program are required to process different types of data, it may be more convenient if the data type is indicated in the job description rather than in the data or program. The parameter section in the job description provides for this, and is written in the form:

```
PARAMETER
  *number
```

The number consists of up to eight octal digits, and is left justified. PARAMETER and the number may be on the same line, separated by one or more spaces.

Examples:

```
PARAMETER *00061247 )
PARAMETER          )
*1060              ) Alternative
PARAMETER *1060000 ) forms
```

When the parameter section appears with other sections, the order of sections is immaterial; if omitted, the parameter is taken to be zero.

The value of the parameter may be read by program using the code instruction

```
1140 4 0 s:
```

which will set the half-word s' = parameter number.

DEC

International Computers and
Tabulators Limited

CS 460

March 1966

(C)