

INTERNATIONAL COMPUTERS AND TABULATORS LIMITED

PREPARING A COMPLETE PROGRAM

FOR ATLAS 1

This document replaces both chapter 10 of the Atlas Provisional programming manual (CS348) and the October 1963 Provisional corrected version of chapter 10. (It also replaces an earlier document entitled Preparing a Complete Program dated September 1963)

AUGUST 1964

I.C.T. LIMITED
68 Newman Street

I.C.T. LIMITED
TELEVISION HOUSE
PETER STREET
MANCHESTER 2

CONTENTS

		Page
10.1	Atlas Jobs	1
10.2	Documents	1
10.3	Document Headings and Titles	1
	10.3.1 Headings	2
	10.3.2 Titles	2
	10.3.3 Rules for title preparation	3
10.4	The Input and Output Sections of the Job Description	4
	10.4.1 The Input Section	4
	10.4.2 The Output Section	5
	10.4.3 Output: General Notes	6
5	A Complete Job Description	7
6	The Magnetic Tape Section of the Job Description	7
	10.6.1 Single Tapes	9
	10.6.2 Files	10
	10.6.3 Deck Allotment	11
10.7	Time Estimates for a Job	12
	10.7.1 Computing Time	12
	10.7.2 Execution Time	13
10.8	Store Allocation	13
10.9	Job Description Format	14
	10.9.1 Order of Sections	14
	10.9.2 Case Changes	14
	10.9.3 Backspace	14
10.10	Composite Documents	14
	10.10.1 Job Description combined with Program	14
	10.10.2 Job Description combined with Data	15
	Document	15
	10.10.3 Data Files	17
10.11.	Tape Markers	17
	10.11.1 The Markers ***Z, C, T and A	18
	10.11.2 The Binary Markers ***B, E and F	18
	10.11.3 The Marker ***P	19
10.12	Input and Output using Private Magnetic Tapes	20
	10.12.1 Extensive Input	20
	10.12.2 Job Description References	21
	10.12.3 Re-use of Documents on System Tapes	22
	10.12.4 Extensive Output	22

ERRATA for Preparing A Complete Program
For Atlas 1

<u>PAGE</u>	<u>LINE</u>	<u>CORRECTION</u>
2	11	replace FORTRAN with HARTRAN
3	13	initial tabs are also ignored
19	15	insert 'When' after !document!.

This chapter explains the way in which a program, and the input information it uses are prepared for running on Atlas.

10.1 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.

10.2 Documents

A document is a self-contained section of information presented continuously to the computer through one input channel (but see also 10.10.3). 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 10.3.1 below. The end of a document is indicated by an 'end of document marker' which usually consists of the characters ***Z which follow, on a new line or new card, 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.

10.3 Document Headings and Titles

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

10.3.1 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. The available languages are Atlas Basic Language (ABL), Extended Mercury Autocode (EMA), FORTRAN and ALGOL

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 10.4, 10.5 and 10.6 below.

10.3.2 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 10.3.3 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.

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
---	--	--

Job Description

Program Document

Data Document

10.3.3. 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 organisation 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) Initial commas, spaces and full stops are ignored and titles must not begin with the word END or the word 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

1 TAPE 3

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.

- b) Any number of consecutive spaces and tabs are stored as one space.
- c) Erases do not become part of the title.

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'.

10.4 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 10.4.1, 10.4.2, 10.6, 10.7 and 10.8.

10.4.1. The Input Section This section begins with the word

INPUT

which is followed by the titles of the 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 data 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/62
2 F6479 BEETLE POPULATION 1961
```

The data document "F6479/2 BEETLE SURVEY DATA/62" could then be selected by the programmer by the instruction

```
1050      0      0      1
```

Data may be placed on the same tape as the program, where it becomes a part of the lowest numbered input stream. With an ABL program such data must come between the enter directive and the end of tape marker (usually ***Z) that terminated the program tape. If the 1050 extracode 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 large amounts of input is explained in section 10.12.1.

10.12.2 The Output Section

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

LINE PRINTER
SEVEN HOLE PUNCH
CARDS
FIVE HOLE PUNCH
ANY
TAPE

Here CARDS means the card punch and ANY means the LINE PRINTER. TAPE is used when a private magnetic tape is called for to hold an especially large amount of output (see 10.12.4).

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 the instruction

1060 0 0 1

The type of equipment should normally be followed by a limit on the amount of output thus:-

OUTPUT
0 LINE PRINTER m BLOCKS

where m is the number of blocks, each of 4096 characters, which will be produced by that output stream. If this limit is exceeded the program is stopped, in order to prevent peripherals being tied up for longer than necessary by useless information produced by a faulty program. 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 all told). Furthermore each use of 1065 (end this record) generates 8 characters to be held in the output well.

The maximum amount of output may also be specified as n 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 on 5 or 7 track tape and on the line printer, or one punched card.

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

OUTPUT
0 ANY 1 BLOCK

is understood.

10.4.3 Output : General Notes

- a) Output 0 Output 0 is used by the Supervisor and 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 monitor information if the program goes wrong.

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

0 ANY 1 BLOCK

is assumed.

Similarly if no output stream is selected by the 1060 'select output n' extracode, 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 the extracode

1071 Break output n

is used. 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

the time and date
OUTPUT n
(Title of Job)

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

10.5 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 other documents of the job.

```
JOB  
F64, J. Smith, I.C.T. London, METALS  
  
INPUT  
0 F64, ANALYSIS PROGRAM  
1 F64/A, IRON CONTENT  
2 F64/B, COPPER CONTENT  
  
OUTPUT  
0 LINE PRINTER  
1 SEVEN HOLE PUNCH 3 BLOCKS  
* * * Z
```

Job Title

Program Title
Data Tapes

output
streams
End-of-tape marker

Job Description

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

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

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

Program Document

Data 1

Data 2

10.6 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 10.12.3.)

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 10.12.

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. Normally the tapes that are to be mounted while the program is running are also listed in the job description. However, in programs which require a not-easily-predicted number of tapes it is possible to get tapes mounted which are not listed in the job description. (Sections 10.6.1. and 10.6.2. show how to list tapes in the job description. The extracodes for mounting tapes are described in section 9.6.1. in the chapter on magnetic tapes.)

Full information on magnetic tapes can be found in chapter 9. However, for the sake of completeness some of the relevant facts are repeated here.

- 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 10.3.2 but also, if the tape serial number is not listed in the job description along with the tape title, 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, but up to 80 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.

10.6.1. 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 in the programmer's number (in the range 0 to 99), the tape serial number (preceded by *) and the title of the tape:

21 *F3699 F1000, LONDON SALES, 1963

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. If the tape is not required to be mounted before the job begins, but rather will be called for in the course of execution by a "mount" instruction (see section 9.6.1), the programmer's number must be replaced by a "tape reference letter" consisting of a single letter:

D *F3700 F1000, LIVERPOOL SALES, 1962

The mount instruction will then refer to the tape by its tape reference letter and assign to it a programmer's number. In the example just given the tape title will consist of the 28 characters starting with F1000 and ending with 1962 (i.e., the two double spaces will have been replaced by single space).

The tape serial number, F3699 in the first example, is permanently associated with the tape. If possible, the Supervisor will call for the tape by its serial number, under which the tapes are filed; the tape number should therefore be included when possible even when its inclusion is not made obligatory by a particular Atlas installation. If it is omitted, the * is omitted also.

Thus in the example

19 T9824, WOLVERHAMPTON SALES, 1962

the title is the sequence of 31 characters starting with T and ending with 2.

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 if the tape serial number is absent the Supervisor refers to the tape by the first 30 characters of its title.

The description for TAPE COMMON consists in a programmer's number or tape reference letter only.

TAPE COMMON

36

All the programmer's numbers and tape references in the job description must, of course, be different. A programmer's number must lie in the range 0 to 99 inclusive; a tape reference can be any letter.

The description may refer to logical channel K of the program; the effect is the same as in extracodes 1012 and 1013. This is done by adding .K to the programmer's number.

Thus:

21.3 *F3699 F1000 LONDON SALES

would request that the tape be mounted on a 'channel' which can then be referred to in the program as channel 3. Tapes on different channels can be written to or read from at the same time. The extracodes 1016 (section 9.6.2) and 1012/1013 (section 9.6.1) allow further tapes to be mounted on the same channel.

If a tape in the TAPE category requires file protection, i.e. no write permit ring and/or no write current, an asterisk should be written immediately before the description:

TAPE

*1 *F3002 F1002 PARIS SALES

With file protection block 0 cannot be written to, so it will not be possible to change the title of such a tape. The program will be monitored if the 1014 (change title) extracode is used on a tape with file protection.

10.6.2. Files

A collection of information may extend over several tapes although the programmer may wish to treat it as a single unit. Such a collection of tapes is called a file and only one of the tapes needs to be mounted at any one time. Each tape of a file is specified in the Job Description by a modified tape heading as follows:

TAPE/m

n (tape serial number) (Title of tape)

where m is the number of the tape within the file, counting from 1 upwards. The programmer's number n will be the same for all m of this file. The final tape of the file is entered as

TAPE/m END

n (tape serial number) (Title of tape)

For example, suppose a file of information extends over three magnetic tapes which have respectively as serial numbers and titles in block 0

Q1432 F 1012 BIRMINGHAM SALES

Q1003 F 1012 LONDON SALES

Q1100 F 1012 MANCHESTER SALES

These would be referred to in the Job Description as

TAPE/1

3 *Q1432 F1012 BIRMINGHAM SALES

TAPE/2

3 *Q1003 F1012 LONDON SALES

TAPE/3 END

3 *Q1100 F1012 MANCHESTER SALES

These tapes would then form file 3. The Birmingham tape alone would be mounted and allotted programmer's number 3, i.e. the same as the file number, before the job begins.

The remaining tapes are mounted as required by use of the extracode 1007 (mount next reel of file Ba and allocate number n to it). The tape sections on the second and subsequent reels are not numbered consecutively from the preceding reel, but start again at section 1.

10.6.3. Deck Allotment.

As stated above the first tape of every file and each single tape that is given a programmer's number in the job description, will be mounted on separate decks 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, given a tape reference in the job description, to be mounted in their place.

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, give the other 2 tapes a tape reference and put in his Job Description:-

DECKS 3

10.7 Time Estimates for a Job

10.7.1 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 time is found by allowing two microseconds to each instruction obeyed and adding to this the expected compiling time. The ABL compiler obeys of the order of 1000 instructions per instructions per instruction compiled.

Alternatively the limit on computing time may be specified as

COMPUTING m INSTRUCTIONS

but one 'instruction' in this context 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 256 'interrupts' per second. Furthermore each multiplication instruction is counted as 2 instructions and each division instruction as 4.

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

or COMPUTING 7.5 SECONDS
COMPUTING 1832 INSTRUCTIONS

If the computing time is less than four seconds this entry may be omitted completely from the job description. In this case a standard allowance of four seconds is made.

10.7.2 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 job 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 10.7.1 above was expected to be held up at most 200 times the job description would include:-

COMPUTING 7.5 SECONDS
EXECUTION 20.3 SECONDS

If the EXECUTION section is omitted for the job description the execution time is taken to be the same as the computing time.

10.8 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 s BLOCKS

where s is the maximum number of 512 word blocks used by the program at any one time. No distinction is made between core and drum store. The word "BLOCKS" may be omitted if desired, but this is not recommended.

This section may be omitted, if the store requirement is less than 20 blocks, in which case 20 blocks will be allocated and charged for. 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. The blocks used by the compiler are not counted unless the computer is retained in the main store after the program is entered.

10.9 Job Description Format

1.9.1. Order of Sections

Separate sections of the job description may be listed in any order. For instance the OUTPUT 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

10.9.2 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 equally well.

10.9.3 Backspace

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

10.10. Composite Documents

10.10.1 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 by 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 example given below illustrate these facilities.

Example 1

```
JOB
F196, J. BLOGGS:  NUMBER FREQUENCY
INPUT
1 F196-TABULATED DATA
OUTPUT
0 LINE PRINTER
COMPILER ABL
-----
-----
-----
-----
***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, FERRANTI:  PRIME NUMBERS
OUTPUT
0 LINE PRINTER
1 CARDS 2 BLOCKS
COMPILER ABL
-----
-----
-----
-----
***Z
```

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

10.10.2 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.
identify the data.

No title is needed to

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:-

```
JOB
F900, J. SMITH LTD: WAGES OCTOBER 1964

INPUT
0 F900, WAGE CALCULATION
1 F900, P.A.Y.E. CODES
SELF = 2

OUTPUT
0 LINE PRINTER 50000 LINES
1 CARDS 2000 LINES

DATA
-----
-----
(Data)
-----
***Z
```

10.10.3 Data Files

It may be easier, for the purposes of some programs, to treat several distinct paper tapes or stacks of cards as a single data 'document'. Such a combination is called a data file. Each separate document is given a modified data heading of the form

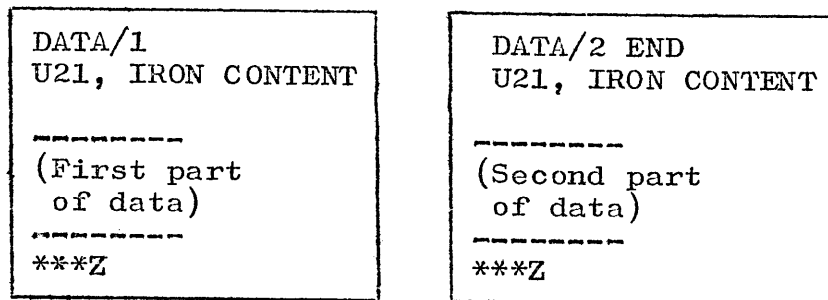
DATA/m

where m is the number of the document within the file. All documents composing the file have the same title and each is ended by a ***Z end-of-tape marker. The last member of such a series must have the heading:-

DATA/m END

These documents may then be fed to the computer in any order and on any peripherals and the computer will combine them as required. In the INPUT section of the job description they will be referred to as one document with the title which each of them bears.

For example, if the data called U21, IRON CONTENT is on two distinct paper tapes these may be headed as follows:-



If the programmer wishes to refer to the file as input two the INPUT section of the job description will contain:-

2 U21, IRON CONTENT

10.11 Tape Markers

So far in this chapter only the marker ***Z has been considered as an end to a document. This is in fact the most common marker 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. One punches 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.

10.11.1 The markers ***Z, C, T and A.

- a) ***Z indicates not only the end of a document but effectively also the physical end of the tape or stack of cards. 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 or stack of cards. 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 or stacks of cards 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.

If the document is data it is better to use the data file system given in 1.10.3 since the parts may then be fed to the computer in any order, on any peripheral.

- 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.

10.11.2 The Binary Markers ***B, E and F

Each of these markers indicates that a binary tape follows.

- 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 or stack of cards. 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 or stack of cards 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 or C. 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. ***A, Z or C is encountered it is itself stored in binary. On cards following 7, 8 a card bearing 7, 8 A, Z or C is also 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.

The marker ***B can also be used to read a fixed number of characters in binary. This is done by prefixing the marker by *n thus:-

*n***B

where n is the number of characters to be read. When the n tape or card characters have been read and stored as a single record, reading continues in internal code.

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 not a standard code the card is assumed to be in binary and is stored as such. Naturally, after a 7, 8 B, E or F has been read all cards are taken as binary regardless of their first column.

10.11.3 The 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.

10.12 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 complete.

Both these facilities are dealt with in this section.

10.12.1 Extensive Input

A document can be copied to a NEW magnetic tape by putting above the document heading the directive:-

COPY TAPE NEW

*tape number tape title

The title specified will be written in section 0 of the new tape and the document following will be copied into section 1 onwards.

If it is required to store the document on a previously used tape (i.e. one which already has a title) the necessary heading is:-

COPY TAPE b

* tape number tape title

Here b is the number of the tape block at which it is intended to begin copying the input.

When the copying process is complete the following information will be printed by the computer:-

*tape serial number/first section where the document is stored/first word title of document.

For example, to copy the first data document of the job given in 10.5 to the NEW tape F1111 it would be fed to the peripheral in the form:-

```
COPY TAPE NEW
*F1111 DATA FOR F64, METALS
DATA
F64/A, IRON CONTENT
- - - - -
(data)
- - - - -
- - - - -
* * * Z
```

The information output by the computer would be:-

```
*F1111/1/0
F64/A, IRON CONTENT
```

The second data document could be sent to section 7 onward of the same tape thus:-

```
COPY TAPE 7
*F1111 DATA FOR F64, METALS
DATA
F64/B, COPPER CONTENT
- - - - -
(data)
- - - - -
- - - - -
* * * Z
```

This could produce the output:-

```
* F1111/7/0
F64/B, COPPER CONTENT
```

Job Description References

10.12.2 In order to use documents copied to private tape, the tape must appear in the tape section of the job description in the usual way. In the INPUT section of the job description the sub-heading:-

i TAPE a/b/c

must appear before the title of the document. Here i is the input stream number, a is the programmer's number for the tape, b the block number, and c the word number within block b at which the document starts. Thus the job description to run the program of 10.5 could begin:


```
JOB
F64, STATISTICAL ANALYSIS OF METALS
INPUT
0 F64, ANALYSIS PROGRAM
1 TAPE 27/1/C
F64/A, IRON CONTENT
2 TAPE 27/7/O
F64/B, COPPER CONTENT
OUTPUT
0 LINE PRINTER
1 FIVE HOLE PUNCH 3 BLOCKS
TAPE
27 *F1111 DATA FOR F64, METALS
```

10.12.3 Re-use of Documents on System Tapes

As already explained, under normal circumstances documents are stored on system tapes before use. Among the information on output 0 will be the location of each document on tape. This will take the form:-

Sa/b/c

where Sa is the system tape number, and b,c are the block and word numbers of the first word of the document. Such tapes will be stored for a fixed period of time and if a document is required again within this time it may be called for direct from system tape, avoiding the use of a slow peripheral. This is done by including in the INPUT section

i TAPE Sa/b/c

Title of document

where i is the programmer's number for the document.

For example, if a program called 'F74 FACTORISATION' has been run and is stored at S7/2/411 it can be re-run with a data document called 'F74 LIST 3A' by a job description as follows:-

```
JOB
F74 FACTORISATION RUN 2
INPUT
0 TAPE S7/2/411
F74 FACTORISATION
1 F74 LIST 3A
OUTPUT
1 LINE PRINTER
***Z
```

Only the job description and the data document would then need to be fed to peripherals. Note that no further reference to the system tape is required.

10.12.4. Extensive Output

Large quantities of output may be written to a NEW magnetic tape by specifying in the OUTPUT section:-

```
OUTPUT
i TAPE a/b/c
type of equipment m BLOCKS
```

Here, i is the output stream number, a the programmer's number for the tape, b the block number, and c the word number where the copying is to start. The last line is as in section 10.4.2.

Thus if output 1 of the job in section 10.5 were 300 blocks instead of 3 the OUTPUT section could be written:-

```
OUTPUT
0 LINE PRINTER
1 TAPE 27/1/0
SEVEN HOLE PUNCH 300 BLOCKS
```

The tape must be specified in the usual way in the tape section of the job description:-

```
TAPE NEW
27 *F1111 F64, OUTPUT METALS
```

This will cause the NEW tape F1111 to be given the title 'F64, OUTPUT METALS' and the output to be sent to this tape beginning at block 1.

If it is required to store the output on a previously used tape, the necessary entry in the OUTPUT section is the same, but the tape must be specified under the heading TAPE in the tape section.

The private tape can be printed by a steering tape consisting of:-

```
PRINT TAPE
*tape serial number/tape title
```

if the whole tape is to be printed

or

```
PRINT TAPE b/c
* tape serial number/tape title
```

if one document is to be printed, from block b word c onwards.

AW
AUGUST, 1964

(C) INTERNATIONAL COMPUTERS AND
TABULATORS LIMITED

1964.