## 6.1 DOCUMENTS

### 6.1.0 General

A document is a set of data (instructions and/or numbers etc.) This data can be required or be produced by a program. Each document has a name which is considered part of the document and it may have some form of terminating mark.

The name of a document consists of not fewer than 2 and not more than 8 components. If the document is on paper tape, cards or a line printer each component is separated from the next by solidus and if on magnetic tape each component is stored in a word. A component can have up to 8 characters; the characters allowed are, letters, digits and . (point). + (plus) as a component (it can't be the first or second) is allowed and indicates that the document is composite. VS as a component is allowed. It is advisable to have as a component the date on which the document was created.

The purpose of each set of data having a name is to ensure that a program uses the correct set of data and it does this by requesting a specific document by its name. Each new document created is given a name.

In a 'document request name' when requesting a document it is allowed to use - (minus) in place of a component (it can't be the first or second) which means "don't care what this component is" and if no further components follow the - (minus) then this means "don't care what further components are". In a "document name" which is to be stored or output it is allowed to use * (asterisk) in place of a component (it can't be the first or the second) which causes Basic Input Routine to store or output today's date as this component. This is the date as stored by the 150/12 instruction. Character VS is allowed but when read by Basic Input Routine it is ignored and each component is stored in a word with the characters right-justified.

A document can be simple or composite. A composite document is a collection of documents. The name of a composite docuement has + (plus) as one of its components and each document on a composite document has its own name. A simple document contains that document only.

Unless Basic Input Routine, a semi-built-in routine or an Orion Library routine is to read a document, there is no restriction on the layout of a document, otherwise the-layout of the document must obey certain rules.

If Basic Input Routine is to search a composite document for a requested document then the layout of the composite document must be such that the names of the documents have in common the components which precede the component + (plus).

For example, if the composite document's name is B/A/+ then the first two components of the documents' names are B/A

### 6.1.1 Magnetic Tape Documents

The name of the document is held in Block O. It is advisable to follow the last block of information on any tape by the non-sequentially addressed block.

#### 6.1.1.1 Composite Magnetic Tape Documents to be read by Basic Input Routine a semi-built-in routine or an Orion Library Utility routine.

(a) Each document of the composite document is made up of a number of blocks. Each document must start with a new block, i.e. no block can contain parts of two different documents.

If pre-addressed tape is used the single-word block addressed markers will intervene.

(b) The first document is a list of the names of the documents (including itself) on the tape. This document is read to find out whether a specific document is on the tape. The name of this document must have the component DOCULIST and it occupies the position of the + (plus). For example, if the composite document's name is B/A/+ then the name of the list document is B/A/DOCULIST. If the component following the + (plus) is the date on which the composite document was created then the component following DOCULIST will be this date.

The order of the names in the list document must be the same as that of the documents on the composite document.

(c) The first block of every document is a 9 word block which contains in the last 8 words, the name of the document. This block is known as the name block and will be read by Basic Input Routine when searching for a specific document.

(d) Length of Blocks

(i) If the mentioned routines are to read the document itself, then the length of each block must be less than 130 words; otherwise the block length is unrestricted.

(ii) One word blocks are only allowed on pre-addressed tapes. These are the block address markers.

(iii) The list document which will be read by the mentioned routines contains the names of the documents (including itself) set out in consecutive groups of 8 words. A block of this document must have length (8n + 1) words where $0 < n \leq 16$. Thus no block may have part of a name at the end of it and the rest of the name at the beginning of the next. The names must be in the order as that of the documents on the tape.

(e) The first word of each block must have

(i) D0 set to 1 if the block is a name block, otherwise D0 is 0.

(ii) D1 set to 1 if the block is the last block of the document, otherwise D1 to 0.

(iii) The l.s. 15 bits set to the binary integer > 1 which is the length of the block. This length is only needed for blocks which are to be read by the mentioned routines.

(f) If the document is to be read by a semi-built-in routine the second word of the first block of information (i.e. the first block after the name block) must contain, in the modifier half, the length of the drum required by the semi-built-in routine.

(g) If the document is to be read by Basic Input Routine then the document must end with a Basic Input Directive (this directive is part of the document) e.g. END or ENTER etc. (Note: USE and PROCESS directives must be the last information of a block since Basic Input Routine after carrying out the required action returns to read the next block.)

(h) Some Orion Library Utility routines use the non-sequentially addressed block and so if these routines are to read the composite document then this block must follow the last block of information.

### 6.1.1.2 Simple Magnetic Tape Documents to be read by the mentioned routines

(i) The blocks of information follow Block 0, i.e. there is no name block.

(ii) Each block must be less than 13O words long.

(iii) The first word of each block will have D0=0 and D1=0 unless it is the last block in which case D1 will be 1.

The l.s. 15 bits will contain the length of the block as a binary integer.

(iv) The second word of the first block will contain the drum length in its modifier half if a serai-built-in routine is to read the document.

(v) If to be read by Basic Input Routine then it must end with a Basic Input directive.

(vi) Some Orion Utility routines use the non-sequentially addressed block and so this block must follow the last block of information.

### 6.1.2 Documents on paper tape and cards

Most documents on these media will be simple documents. The first line i.e. card for cards (excluding run out etc.) will be the DOCUMENT directive with the name of the document. It is advisable to terminate a document with some terminating mark.

All documents to be read by Basic Input Routine or a semi—built-in or an Orion Utility Routine must finish with a terminating directive, for example END, ENTER, READ etc. The format of a document to be read by Basic Input Routine is defined in 7.1. For a document to be read by Symbolic Input Routine the format is defined in 7.2.

Basic and Symbolic Routines do not recognise composite documents on paper tape or cards.

### 6.1.3 Conventions for distinguishing various versions and of documents on the System tape

The document request name given in the specification gives only the invariant components. The next component gives the form (i.e. language)

either BIN means binary and map (see 7.3)

or CHA means characters (i.e. Basic Input Language - see 7.1.)

The next component gives the date.

If there is a semi-built-in form of the document available then the third component must be different from the third component in other forms and this must be given in the specification,

the fourth component will be SEMI, and

the fifth component is the date.

Example

Document request name.

ORION/SYSTEM/COMPILER/SYMBOLIC/-

Name of Character form

ORION/SYSTEM/COMPILER/SYMBOLIC/CHA/18NOV63

Name of binary and map form

ORION/SYSTEM/COMPILER/SYMBOLIC/BIN/20NOV63

Semi-built-in name

SYMBOLIC

Name of semi-built-in form

ORION/SYSTEM/SYMBOLIC/SEMI/1JUL64

## 6.3 Semi-Built-In Programs

### 6.3.1 General

The semi-building-in technique is employed when a routine is used sufficiently frequently by different programs to warrant its retention on the drum in a general form for as long as possible. A semi -built-in program (SBIP) can be kept on the drum outside any program's reserved region until the space it occupies is required by some other program. Not more than eight SBIPs can be held in the machine at any one time, but all SBIPs are also permanently stored on the system tape.

### 6.3.2 Form of SBIPs

Since any SBIP may be used by a number of different jobs, all SBIPs are stored on the drum in a mapped form, and are not relativised until they are brought down into the core store. In order that SBIPs may be read onto the drum as rapidly as possible, the permanent program on the system tape is held in the same form as the program on the drum, both of these programs being in binary. The form used is similar to the binary-and-map form described in section 7.3, except that in the case of SBIPs the whole of the map follows the binary program, the latter being preceded by the chapter change word pairs. A library routine, MAPPER, is provided (q.v.) to convert Basic Input language documents into semi-built-in form. The rules given there must be adhered to for every SBIP.

### 6.3.3 Operation and Use

An SBIP may be called in by a USE (or COM) or READ directive, or by a 150/51 instruction. The specification of an SBIP will indicate which should be used. A list of SBIPs currently in the machine is kept by OMP, and if the called SBIP is on the drum, the directive or instruction will be implemented. If the SBIP has been called by a directive the first chapter will be relativised according to the datum point of the job, brought down into the core store and entered. If a 150/51 is used the chapter specified is relativised, brought down and entered or not, as instructed (see section 5.3.51)

If the SBIP is not on the drum OMP calls a chapter of Basic Input to read the required SBIP from the system tape. This chapter will use registers A64 to A255 of the main program's reserved region and the previous contents of these registers will be lost. If these registers are outside reservations, RES.VIOL will result.

Basic Input may not be able to read the SBIP immediately because there is insufficient drum available to store it, or because there are already eight SBIPs in the machine, or because another (or the same) SBIP is being read in. In any of these cases a message will be printed on the Flexowriter and the program halted. (See section 5.8.2) The program will continue when the reason for the stoppage has been removed. A list of SBIPs on the drum can be obtained by using the NAMES directive (section 5.7.4.5) and an SBIP not in use may be removed from the drum by using a REMOVE directive (section 5.7.4.10). The amount of drum store occupied by SBIPs not in use can be found by using a SPACE directive (section 5.7.4.4)

As soon as the chapter of Basic Input has successfully read in the SBIP, whether it has been halted or not, a message will be output on the Flexowriter (see section 5.8.2) and the required chapter of the SBIP is then transferred to the core store as described above.

### 6.3.4 Names of Semi-Built-In Programs

The conventions followed in naming any document on the system tape are given in section 6.1.3. SBIPs follow the rules given there, so the document name of the semi-built-in form of a program might be

ORION/SYSTEM/SYMBOLIC/SEMI/1JUL64

The first, second and fourth components are defined by convention; the fifth component is the date. The third component cannot contain more than eight characters which must be chosen from the letters, digits and . (point).

This third component, in this example SYMBOLIC, is also known as the semi-built-in name, and is used by itself when an SBIP is called in by a directive, or as the content of the Y address when a 150/51 instruction is used.

### 6.3.5 Overwriting

An SBIP will be overwritten if the space it occupies is required by another program (except another SBIP) unless it is in use, that is unless it has been called in by a job in the machine. This protection is automatically removed when the job concerned is abolished, or it may be removed before that by a 150/51 instruction (see section 5.3.51)

### 6.3.6 Library Routines in Semi-Built-in Form

The following routines are in semi-built-in form

PREADD

PRINT

PUNCH

## 6.4 Standard Restarts

### 6.4.1 7-track paper tape reader

#### 6.4.1.1 Parity Fail

There are 3 standard restart procedures for parity fail.

When a parity failure occurs a message is printed on the Flexowriter and the reader is disengaged. The messages are of the form

job-name geo. name RESTART n.

where n= 2, 1 or 0

The failed character which is on the output side of and adjacent to the character over the reading position should be examined. If it is of even parity or there is any obvious reason for the failure then the character should be marked and, in general the job abolished. If it is of odd parity then the appropriate restart procedure should be carried out.

Procedure for RESTART 2

This ensures that the tape is synchronised with what has already been read.

(i) Mark the failed character

then (ii) Move the tape back one character (this is to allow for the case where the failed character is NL).

then (iii) Move the tape back to between the 2nd and 3rd NL characters beyond (ii) above, but in any case not more than 10 characters beyond the 2nd NL character.

then (iv) Finally, press the Engage button.

Procedure for RESTART 1.

This procedure is as that for RESTART 2 except that (iii) should read:

Move the tape back to between the 1st and 2nd NL characters beyond (ii) above, but in any case not more than 10 characters beyond the first NL character.

Procedure for RESTART 0.

This allows for a parity fail within the procedure just attempted. This is to repeat the procedure just attempted so the operator will repeat the procedure from (ii) onwards for the appropriate restart.

### 6.4.2 Card Reader

There is one standard restart for card failures.

The message printed on the Flexowriter is of the form.

job-name geo. name RESTART 1.

and the reader is disengaged.

The procedure is to take and mark the failed card from the reject pocket and to place it at the bottom of the feed-hopper and engage the reader.