

Volume 2: PROGRAMMING INFORMATION

Part 1 : PROGRAMMING LANGUAGES

Section 6 : FAS OPERATING SYSTEM

Contents

	Page
Chapter 1: INTRODUCTION	
1.1 Purpose	1
1.2 Distribution	1
1.3 Machine Configuration	1
1.4 Summary	1
Chapter 2: FUNCTIONS	
2.1 Start of Run and Operating Procedures	3
2.2 Commands	4
2.2.1 Form of Commands	4
2.2.2 Character Set	5
2.3 &JOB Command	5
2.4 &CHANGE Command	6
2.5 &FORTRAN Command	6
2.6 &ADDLIB Command	7
2.7 &FT2 and &FT3 Commands	8

	Page	
2.7.1	&OPTION Command	8
2.7.2	&LOAD Command	9
2.7.3	&LIBRARY Command	9
2.7.4	&RLB Command	10
2.7.5	&RUN Command (FORTRAN version)	10
2.7.6	&END Command	11
2.8	&ALGOL Command	11
2.8.1	&RUN (ALGOL Version) Command	13
2.8.2	&ADDLIB (ALGOL version) Command	13
2.9	&ALIBS Command	14
2.10	&CALL Command	14
2.11	&SIR Command	15
Chapter 3:	NOT ALLOCATED	17
Chapter 4:	OPERATING INSTRUCTIONS	
4.1	Copying the FAS System	19
4.2	Running FAS Jobs	21
4.3	Use of MTEDIT	23
Chapter 5:	OPERATING SYSTEM ERRORS	
5.1	FAS Systems Errors	25
5.2	Other Error Indications	26
5.2.1	Output from the Punch	26
5.2.2	Output from the Teleprinter	26
5.2.3	Rewind in MANUAL	28
5.3	Action after Output of Errors	28
Chapter 6:	UTILITY PROGRAM CALLPM	
6.1	Purpose	29
6.2	Machine Configuration	29
6.3	Operating Instructions	29
6.4	Error Messages	30
6.5	Creation of Library Tape	31

	Page
Chapter 7: UTILITY PROGRAM MTEDIT	
7.1 Purpose	35
7.2 Machine Configuration	35
7.3 Operating Instructions	35
7.4 Format of Output for LIST	39
7.5 Error Procedures	39
7.6 Example	41
Chapter 8: STORE USED AND TIME TAKEN	
8.1 Store Used	43
8.2 Time Taken	43

Chapter 1: INTRODUCTION

1.1 Purpose

This system provides facilities for compiling and storing programs on magnetic tape; this reduces the amount of paper tape handling required for ALGOL and FORTRAN programs.

1.2 Distribution

The two tapes necessary to implement this system are:

- (i) System Tape (magnetic tape reel)
- (ii) Paper tape program MTBOOT - a bootstrapping program.

1.3 Machine Configuration

A minimum configuration of a 903 computer with a minimum of 8192 words of core store, paper tape reader and punch, control teleprinter and two or more 9kHz magnetic tape handlers is required to implement this system. This system may be expanded to 16384 words of core store and/or 4 magnetic tape handlers.

1.4 Summary

This system provides a basic library tape which holds the system control programs and modified versions of the ALGOL and FORTRAN compilers. It provides facilities for:

- (i) compiling ALGOL and FORTRAN programs from paper tape source code onto magnetic tape and running said programs with a minimum of operator intervention.
- (ii) allowing the user to add programs to the system tape or to a separate library tape. These programs may be in the form of binary dumps, intermediate coded ALGOL programs, or FORTRAN subprograms.

Chapter 2: FUNCTIONS

2.1 Start of Run and Operating Procedures

- (1) A working copy of the system tape is loaded onto any tape handler (normally handler 0).
- (2) A scratch tape should be loaded on another handler (normally handler 1). This scratch tape is then initialised to the name SCRATCH by means of the MTINIT program which is called down from the system tape.
- (3) 900 program MTBOOT is loaded into the paper tape reader and entered by initial instructions at location 8181. The number of the system tape handler is typed in the form:

Hn

- i.e.
- H0 - Tape handler 0
 - H1 - Tape handler 1
 - H2 - Tape handler 2
 - H3 - Tape handler 3

according to the handler used for the system tape.

- (4) To enter the FAS system, type ')' i.e. closed parentheses. If other utility programs are required see description of CALLPM - Chapter 6.
- (5) If the scratch tape is not on handler 1, the operator should type the necessary &JOB and &CHANGE commands (see section 2.4). The operator either switches INPUT SELECT to paper tape reader input, or types in the necessary commands to run the first program. If all commands are on paper tape, it should be sufficient for the operator to

simply load each paper tape in the reader, press the READ button and wait until all the tape is read before repeating the process. Certain errors and/or a failure by a program to reach STOP, END, or 'END' will require operator action.

2.2 Commands

The FAS operating system is controlled by a series of commands which may be either entered from the teleprinter, or read in from the paper tape reader. This system allows semi-automatic operation once the operating system has been entered and the INPUT SELECT switch set to the 'READER' position.

2.2.1 Form of Commands

The general form of a command is:

& symbol, or & symbol, list;

The character (&) is followed by a name (called 'symbol') which can be followed by further items, each separated by a comma; the command is terminated by a semi-colon (the rest of the line after the semi-colon is ignored). The items which follow 'symbol' take one of three formats (depending on the function) are as follows:

(1) Handler number

(2) Symbol

(3) Label.

The handler number must be a single digit in the range 0-3 no other characters, except spaces (which are ignored) are allowed.

The symbol is an alphanumeric string whose first three characters only are used; any further characters are ignored.

Label is an alphanumeric string whose first six characters only are used; any further characters are ignored. Spaces before the first significant characters are ignored, but otherwise are recognised. Labels are used in magnetic tape blocks.

When using commands the character & must be the first character on a new line, i.e. the first character immediately following a line feed. The whole command should be on one line and must not be more than sixty characters long.

2.2.2 The Character Set

Characters are stored in 900 Series internal code and any character in that 64 character set is usable as a valid character. Some characters are specially treated on processing.

The lower case letters a to z are converted to the upper case characters A to Z. The horizontal tab is converted to space, carriage return, blanks and deletes are ignored.

If a halt code is the first character on a line, it will cause a dynamic stop to occur. Otherwise, a halt code will be stored as a left pointing arrow (\leftarrow) as will any other character not in the set mentioned previously including an odd parity character, which also causes output of an error message.

2.3 &JOB

This command is given at the start of each of a sequence of commands in which the commands depend on one another, but do not depend on any preceding sequence of commands. The sequence of commands defined by an &JOB are for example, the translation, loading and running of a FORTRAN program. If translation or assembly fails, the system will ignore all commands until the next &JOB is encountered. The format of &JOB is:

&JOB, STRING;

where STRING represents any sequence of valid characters; this information

is copied to the teleprinter (tape punch if an on-line teleprinter is not fitted). It is to be used for such information as the &JOB name, the programmer's name and/or code number and special information for the operator.

Example:

&JOB, A.SMITH, 16/10/69 : LOAD MASTERFILE ON H3;

2.4 &CHANGE

This command is used to change the allocation of magnetic tape handlers to perform certain tasks in the operating system. The handlers used are given logical numbers 0, 1, 2, 3. The actual hardware handler numbers normally correspond to the logical numbers; this allocation is preset on the system tape.

Logical handler 0 is used for the system file, and the systems tape should normally be loaded on actual handler 0. Logical handler 1 is used for ALGOL and FORTRAN compilation to hold the intermediate code. This it does using a scratch tape (initialised to the file name SCRATCH and mounted on logical handler 1 (usually actual handler 1)).

If it is desired to change the allocation of handlers in use, the command &CHANGE must be used. The full format is:

&CHANGE, n, m;

where n and m are handler numbers (see section 2.2.1), and represent logical and actual handler numbers respectively. Thus if the scratch tape for compilation is mounted on handler 3, use:

&CHANGE, 1, 3;

to re-allocate logical number 1 to actual number 3.

Note that logical handler 0 is allocated to the actual hardware number typed into MTBOOT, when the system is entered.

2.5 &FORTRAN

This command indicates that a source code FORTRAN

program (in 903 Basic FORTRAN) is to be compiled. A scratch tape initialised to SCRATCH must be mounted on logical handler 1. The format of the command is:

&FORTRAN, NAME;

where NAME is in label format (see section 2.2.1) and represents the name of the program, subprogram, or group of subprograms following ; this is the name used within the FAS operating system. It can be the name given to the FORTRAN program by the GLOBAL statement which occurs at the beginning of a program.

The FORTRAN text, including subprograms, must be input following this command. If some of the FORTRAN programs are required to be added to the library by an &ADDLIB command (see section 2.6), this text may consist of a set of subprograms for independent compilation. Otherwise, the main program will be the first text to be input.

The final END of the program or set of programs must be followed by a % character which must be the first character on a new line. This character causes a return to the systems program. It is not necessary to compile in one operation all the subprograms required to run the program; some may be compiled independently and added via the library facility.

2.6 &ADDLIB

This command is used to add to a library tape the FORTRAN program which has just been compiled. It is not advisable to compile a program directly onto a library tape since the translation may fail. Therefore, the &ADDLIB facility is provided to copy the program just compiled from logical handler 1 onto the end of a library tape. The complete format is:

&ADDLIB, n;

where n is the logical number of the library tape (see 2.4). The library tape may be one created by the user or it may be the systems tape on logical

handler 0. In the latter case a copy of the systems tape must be preserved off-line in case of failure during writing.

As the object program in SIR code is being added to the library, certain redundant information is eliminated as follows:

The characters blank, carriage return and erase are ignored; SIR comments beginning on a new line are ignored; these will usually be those generated by the FORTRAN Translator .

The use of &ADDLIB for ALGOL programs is described in section 2.8.2.

2.7 &FT2 and &FT3

These commands are used to activate the FORTRAN loading and running process. The loading routines for FORTRAN contain their own command processing system; hence, &FT2 or &FT3 must be followed by one or more commands (see next paragraph). &FT2 is used for running on an 8192 word core store; &FT3 is used for a 16384 word core store. The formats are:

&FT2; or &FT3;

The commands which may be used following the &FT2 or &FT3 are:

&OPTION, &LOAD, &LIBRARY, &RLB, &RUN, &END.

2.7.1 &OPTION

This controls the loading of FORTRAN programs, giving the option of label lists or relocatable binary tape. The format of &OPTION is:

&OPT,AAA,BBB;

where AAA and BBB represent 'symbols' which provide the following options:

LAB - this gives a label list

NLA - this stops the output of a label list

PRB - punch out a relocatable binary paper tape.

LDG - load-and-go operation (normal)

There must be at least one of these after &OPT.

On entry, an option of &OPT, LDG, NLA; is assumed. Therefore, the command &OPTION need only be used if PRB or LAB are required at some stage.

2.7.2 &LOAD

This causes the FORTRAN program just translated to be loaded from logical handler 1.

The format is:

&LOAD;

2.7.3 &LIBRARY

This command causes a named program (or subprogram) from a library file to be loaded. It may be used in place of, or in addition to, an &LOAD command. The format for this option is:

&LIBRARY, n, AAAAAA, BBBB;

where n is the logical handler number of the library file required, and AAAAAA and BBBB represent labels (see 2.2.1). From one to six labels can be used with an &LIBRARY command. Each label specifies the name of a program, subprogram or group of subprograms which have been added to the file by an &ADDLIB command. The name of the unit or group is the name given in the original &FORTRAN command.

Programs are loaded in the order they are stored on the magnetic tape file, and not necessarily in the sequence they occur in the &LIBRARY command. Thus to ensure that the main program is loaded first, it should be specified singly in an &LIBRARY command (or alternatively loaded by &LOAD). Any further subprograms required are then specified by additional &LIBRARY statements (or by an &LOAD statement).

2.7.4 &RLB

This is used to load relocatable binary tapes via the tape reader. The tapes may be prepared by FT2 using &OPTION, PRB;. This command will only be used as a means of loading into an 8K store, programs which are too large to be loaded by other means.

The command format is:

&RLB;

This is then output onto the teleprinter as follows:

FT2 &RLB;

The computer then enters a dynamic halt, allowing the relocatable binary tape to be loaded into the paper tape reader. By entry at location 12, the tape will be loaded. After loading, a FIRST - LAST - NEXT message is printed out, and the computer again enters a dynamic halt. If further tapes are to be loaded, re-enter at location 12.

After all tapes are loaded, a return is made to &FT2 by entering at location 9. If there are errors, return can be made to the FAS operating system by entering at location 10.

Note when the sequence &FT2 &RLB; is printed out the entire command is in fact printed out. Thus, the programmer can add further information after the semi-colon, e.g. the name of the relocatable binary tape to be loaded, which will be displayed for the operator.

N.B. After &RLB; has been used, the use of &LOAD and &LIBRARY is prohibited.

2.7.5 &RUN

This command terminates the loading process, and if there are no errors, will load the run-time routines and commence running the program. The command format is:

&RUN;

If there are errors during the loading of a program, a return is made to the FAS operating system, and paper tape is ignored until the next &JOB command is read.

During the running of a program, the FORTRAN PAUSE statement will cause a dynamic halt. Continuation is by entry at location 9. Alternatively, the program may either be restarted from the beginning by entry at location 11, or a return can be made to the FAS operating system by using the error entry at location 10 (see preceding paragraph for effects).

A run-time error is treated in the same way as the PAUSE statement.

A FORTRAN STOP statement will cause an automatic return to the FAS operating system and permit the next FAS command to be read. PAUSE must be used if it is desired to retain the FORTRAN program in core store.

2.7.6 &END

This command is used to terminate loading of a program; a return is made to the FAS operating system without actually running the program. The command format is:

&END;

The command is essential to terminate punching of a relocatable binary paper tape when the command &OPTION, PRB; is used. Alternatively, it can be used in place of &RUN; during program development.

If there is an error during loading, the FAS operating system will search through the program until the next &JOB command is read. If no errors occur, the next FAS command will be read.

2.8 &ALGOL

This command indicates that a source code ALGOL program is to be compiled. A file initialised to SCRATCH must be mounted

on logical handler 1.

The format for the command is:

&ALGOL, NAME;

or

&ALGOL, NAME, nn;

where NAME is in label format (see 2.2.1) and represents the name of the program following. This is the name used within the FAS operating system. It may (but not necessarily) be the name given to the ALGOL program by the title placed before the first 'BEGIN'. This title must still be included, starting on a new line.

The symbol nn represents a two digit octal number, which gives the user the option of various modes of translation. If no option is specified, the Translator does not compile checks (e.g. CHECKR) and does not give report information.

The useful options are:

- 20 Translate with reports
- 04 Translate with checks compiled
- 24 Translate with reports and checks
- 22 Do not compile, but check for errors and give reports

Example: &ALGOL, PROG2, 04;

The ALGOL text will be automatically read in from paper tape once the ALGOL Translator has been called down. The program ends in the normal way with:

"END" optional comment;

On a new line following immediately after the last semi-colon, the character '%' must be read. This indicates the end of the ALGOL program and causes a return to the FAS operating system. If '%' is read

elsewhere in the program at the beginning of a new line, the operating system assumes that a breakdown of the block structure has occurred, and reports an error (e.g. due to too many "BEGIN"s not matched by "END"s). If commands are being input from the teleprinter the terminating '%' may also be input by this means.

Once the ALGOL program is input other commands are available, as described below:

2.8.1 &RUN (ALGOL Version)

This command, input after &ALGOL, causes the ALGOL program just translated to be loaded and run. The data for the program, if any, will be input following the &RUN command. If the program is completed normally, a return is made to the FAS system to process the next job. If there are one or more FU errors when loading the program, loading will stop after output of FIRST NEXT. Load any paper tape relocatable binary code procedures which may be needed, and enter at 11. Repeat for as many paper tapes as are required. As soon as all names have been loaded, the program will start to run. Alternatively, the program may be forced to run with code procedures missing, by entering at 14.

The format of the &RUN command is simply:

&RUN;

The command may be used after &ADDLIB as long as logical handler 1 has not been used for writing since the appropriate &ALGOL command. If the program reaches a PAUSE or halt code on data tape it may be continued by entering at 9. Alternatively, the program may be restarted by entering at 14.

2.8.2 &ADDLIB (ALGOL Version)

This command causes the program just loaded to be added to a library tape. It may be used after &RUN provided the tape on logical handler 1 has not been used for writing since the appropriate &ALGOL command.

The format of the command and its use is identical to the FORTRAN version:

```
&ADDLIB, n;
```

where n is the logical handler number of the library tape. This tape must have a write permit ring fitted. It may be either a file made up by the user or the systems tape on logical handler 0. In the latter case a copy of the original systems tape must be preserved (off-line) in case of failure during writing.

2.9 &ALIBS

This command causes an ALGOL program filed on a library tape to be loaded and run. The program must have been filed by the &ALGOL, &ADDLIB sequence. The format of the command is:

```
&ALIBS, n, LABELN;
```

where n represents the logical handler number from which the program is to be loaded, and LABELN represents the (label) name of the program. The command first causes the ALGOL Interpreter to be loaded, followed by the program specified. If FU errors are displayed, load relocatable paper tape as in Section 2.8.1.

2.10 &CALL

This command is used to call down a program dump dumped on tape by MSDUMP. It does this by calling the program CALLPM into store. The format used for the command is:

```
&CALL, n;
```

```
LABEL, entry;
```

where LABEL represents the name of the program to be called down, and entry represents an integer giving the program entry point or zero (see description of CALLPM, Chapter 6). LABEL must be placed on a new line. The logical handler number to be used is represented by n.

2.11 &SIR

This command is used to call down a modified version of the SIR basic assembler and compile a SIR program onto logical handler 1. This can only be used for the assembly of code procedures for ALGOL. These procedures must be added to the systems tape by an &ADDLIB, 0; command.

The format of the command is:

&SIR, PROGNM;

where PROGNM represents the name which will be used by the ALGOL library search, following translation of a program under &ALGOL. One or more code procedures may be translated after an &SIR command, but if they are to be added to a particular ALGOL program, that program must call the procedure named in the &SIR command.

The &SIR command must be followed by input of a SIR program, starting with a *0 (option zero) or *1, on a new line. There must be a global label list following the option, and one or more SIR blocks. The whole must be terminated by a single '%', on a new line by itself. This '%' has the usual SIR effect, and also causes a return to the operating system to read the next command, which will usually be &ADDLIB, 0;

&ADDLIB must be used to copy the program generated by SIR onto the systems tape, if the program is an ALGOL code procedure.

Note that the &SIR command cannot be used for code to be added to a FORTRAN program. Code subroutines to be added to a library tape for use by FORTRAN must be compiled by adding CODE and FORTRAN statements to the beginning and end of the program, and using an ordinary &FORTRAN command.

Chapter 3: NOT ALLOCATED

Chapter 4: OPERATING INSTRUCTIONS

Details concerning the operation of the system are given in Chapter 2, sections 3.5 and 5.3.

This chapter describes the standard form of operation (assuming that programmers provide tapes and operating instructions to the computer operator).

When a version of the FAS system is received, it must be copied onto one or more of the user's tapes. It is essential that each magnetic tape installation always keeps an original copy of the system.

4.1 Copying the FAS System

The original version or subsequent copies of the system's tape (or any other magnetic tape file), may be copied by the following procedure:

- (1) Load the systems tape (without a write permit ring) onto handler n, (suggested handler 0). Load one or two SCRATCH tapes (with write permit rings fitted) onto the other handlers. Make sure the handlers are in 'REMOTE' state. See paragraph(8) below for instructions on making up SCRATCH tapes.
- (2) Load the paper tape '900 MTBOOT' and enter at initial instructions (8181).
- (3) Type in the handler number of the master tape, in the form Hn.

Example:

H0

- (4) The word CALLPM is displayed after 2 or 3 seconds. If it is not, press RESET on the computer control panel and check that the handler

is in REMOTE state, with the correct tape loaded.

- (5) Type:

MTEDIT,

this causes ENT to be displayed.

Type:

32,

which causes MTEDIT to be loaded into store after between 15 to 40 seconds. When MTEDIT is in store it outputs:

MTEDIT

- (6) Type in the following, with a pause of 2 to 3 seconds at the end of the first line:

0,QFAS-SYSTEMS,WRI,1,

KPY,600,600,

This causes the tape on handler 0 to be copied onto the tape on handler 1. Alternatively, in place of KPY,600,600, the form CPY, may be used, but this causes a slower copy, although less tape movement. When the copy is finished EOR is output and both tapes are rewound.

- (7) Another copy of the systems tape may then be made by typing:

0,QFAS-SYSTEMS,WRI,2,

KPY,600,600,

if the systems tape is on handler 0 and a SCRATCH tape on handler 2.

When copying is complete, the original master should be dismounted from the handlers.

An installation master should be kept in reserve, and only used to produce working copies.

- (8) If no SCRATCH tapes are available at step (1) they may be produced by initialising some tapes with MTINIT, as follows:

The tapes to be initialised must be new tapes, and/or have no useful information or programs on them.

Carry out steps (1) to (5), but specify MTINIT in place of MTEDIT. Then type the number of the handler on which the tape to be initialised is loaded, terminating this number with a comma. The header block should be listed, and in that event the operator must check that this tape is one that may be overwritten.

If the magnetic tape moves for more than 10 seconds in one direction, press RESET on the computer control panel (not the magnetic tape controller), and enter at 35.

When INP NEW HDR is output, type in the name:

SCRATCH,

The program should then type:

OK

To initialise further tapes, re-enter at 32. To re-enter the CALLPM program, re-enter at 7328.

(binary 1110010100000). Go to step (5) of the above sequence to call down MTEDIT.

4.2 Running FAS Jobs

- (1) Load a working copy of the systems (QFAS-SYSTEMS) tape file on handler 0. (If handler 0 is not available load on handler n, where n = 1, 2 or 3). The reel

should not have a write permit ring fitted unless the operator is specifically instructed to do so.

- (2) Load a SCRATCH tape on handler 1, (or handler m if convenient, m = 0, 2 or 3). To initialise a SCRATCH file see (8) in paragraph 4.1.
The SCRATCH tape should be loaded unless the operator is specifically instructed not to load it. It is required if any of the commands &ALGOL, &FORTRAN, &SIR or &OVERLAY are used.
- (3) Check that the handlers required are set to the 'REMOTE' state and input and output select switches are set to AUTO.
- (4) Load the paper tape program MTBOOT and enter initial instructions. Type H0 or Hn where n is the handler number on which the systems file is mounted. Characters input, before the letter H, are ignored.
- (5) The word CALLPM should be output. If CALLPM is known to be in store step (4) may be omitted and entry made at 7328 to CALLPM.
- (6) Type close round bracket ')'.
If the SCRATCH tape is mounted on handler number 0, 2 or 3 (i.e. not on 1) then wait until DEMAND light comes on, then type:

&JOB;
&CHANGE, 1, m;

where m is 0, 2 or 3 respectively.

- (8) Then, or otherwise, load the first paper tape (with FAS commands) in the reader. Switch INPUT SELECT to the READER position.
- (9) Providing there are no serious errors, the programs should run automatically. When the end of one paper tape is reached, the next should be loaded and the system restarted by entry at location 9. If an error is output and the program stops, the next paper tape in sequence should be loaded and attempt to enter at location 10 should be made. If the program will not restart, go back to step (3). Press RESET on the computer control panel if necessary. Entry at location 10 should cause the operating system to be re-entered, and commands to be read, skipping to the next &JOB command.

4.3 Use of MTEDIT

The program MTEDIT (see Chapter 7) can be used:

- (i) to manipulate the system tape by selectively copying sections of that tape,
- (ii) to list the contents of the tape, i.e. names of programs filed.

If a particular installation uses predominantly one of the languages ALGOL or FORTRAN, it is advised that a working copy of the tape be made which omits the operating system block for the language not used (i.e. FORTRAN or ALGOL) - Blocks 1 to 11 are essential and must be preserved in the correct order; other blocks may be re-ordered. The system blocks relating to FORTRAN and ALGOL are:

For ALGOL,

QAL1, QAS2, QAPLT1, QAPLT2, DECODE, ADVANCE, BUFFER, QATRIG,
ARCTAN, SQRT, LOWBOU, RANGE, QASTRI, MTOPEN, MTCLOS, MTMRIT,
SETCOM, RAGET,
SDRN/903/FAS/2

For FORTRAN,

QFOR1P, QFORT1, QFT3RT (used only for 16K), QFT2TB, QFORT3, MTFORT
and CHAIN.

Chapter 5: OPERATING SYSTEM ERRORS

5.1 FAS System Errors

The FAS operating system generates its own messages in addition to those generated by the ALGOL and FORTRAN compilers. For details of the latter see Vol. 2A, sections 2.1.4 and 2.1.3 respectively.

The format of a FAS error report is as follows:

FAS En & COMMAND;
or FT2 En & COMMAND;

where FAS indicates that the error has occurred in the main FAS system. En is the actual error number; &COMMAND; represents the last command read in and is printed out in its entirety. For action after output of an error see section 5.3.

Error No.	Cause
1	One or more characters in the command have odd parity, and when the error report is output the character(s) are converted and output as the character ←.
2	Statement overflow. Only the first sixty characters are printed out in the error report.
3	Command out of context.
4	Invalid command.
5	Invalid symbol, handler number, or label in command.
6	Magnetic tape error. Status word is printed on next line.
7	No program compiled, or wrong handler number.
9	Essential program missing from systems tape.

There are a number of variants to the standard error report format. For errors occurring within &FT2; FAS is replaced by FT2.

In error E6 (both in FAS and FT2), the status word is produced as an octal constant commencing with the character '&'. For interpretation of the status word bits see sections 2.7.2 to 2.7.4.

5.2 Other Error Indications

5.2.1 Output from the Punch

If the paper tape punch outputs a continuous stream of characters, the program MTBOOT has detected an error. If the punch outputs a 'binary count' pattern of characters from 0 to 255 (binary value) this indicates an error detected by a version of MTBOOT built into one of the FAS programs. The most common causes of such errors are that either the handler holding the system tape is in a manual state, or a corrupt or wrong tape has been loaded onto the systems tape handler.

This crude error indication is given because the main error routines are not available, or have been overwritten when these errors are detected.

5.2.2 Output on the Teleprinter

*NP:NOT FOUND

This error indicates that the name specified to CALLPM cannot be found. The correct name should be typed in, or MTBOOT should be used to bring down CALLPM from a different tape.

*INPUT ERROR

This error indicates a mistake in input to CALLPM. Type in the correct name and entry point again.

**

Output of a pair of asterisks indicates that the overlay of program dump currently being called down overwrites essential program or data. This indicates program between 8050 and 8191, or between 8110 and 8191 if the last block is being loaded. This error is given by CALLPM.

NO %AFTER END

This error indicates that following an ALGOL Translation the next significant character read (after the semi-colon following the final "END";) is not %. Spaces, newlines, blanks, erase and carriage returns are ignored when searching for %. If the error is given the FAS system will skip to the next &JOB. The error may indicate a break down of the block structure or missing "BEGIN".

%BEFORE END

This error indicates that the ALGOL Translator within the FAS system has detected a % character as the first character on a new line, not following the final "END";. If the % is valid, within a string, a spurious error may be avoided either by inserting a space before the % or the use of extra string quotes.

The error normally indicates a breakdown of the block structure or a missing "END" or closed quote in the program.

5.2.3 Rewind in Manual

If a permanent irrecoverable error is found when attempting to write intermediate code (from a compiler or an overlay dump) onto logical handler 1, the tape on this handler is rewound and set into MANUAL; the remainder of the current job is ignored. An error will usually be given on the next &JOB to be processed.

5.3 Action After Output of Errors

After the errors listed in (Section 5.1) the computer will stop. In most cases it is possible to continue by entry at location 9, to read the next command, or at location 10 to skip to the next &JOB.

Under most error conditions in the FAS system and its associated compilers, run-time systems, etc., a return to the basic system can be made by entry at location 10; this causes a skip to the next &JOB. However, it is sometimes impossible to restart by this means. In this instance, press RESET on the computer control panel (not the magnetic tape controller) and use MTBOOT to call the system back into store. Check that all necessary handlers are in the REMOTE state. If the system still remains inoperative, it must be assumed that either the systems magnetic tape is corrupt, or there is a hardware failure.

Chapter 6: UTILITY PROGRAM CALLPM

6.1 Purpose

This program is used in conjunction with MTBOOT to call down named programs from a library tape. Programs in any part of store except 8100 to 8190 can be called down.

6.2 Configuration

A 903 or 905 computer with a minimum of 8192 words of core store and one or more magnetic tape handlers is required for this program.

6.3 Operating Instructions

CALLPM is used to bring down programs from the QFAS-SYSTEMS tape or a private file of programs created as described in 6.5. Programs can be called down from this file as follows:

- (1) Load the tape, MTBOOT and enter by initial instructions (at location 8181).
- (2) Input the handler number of the library tape by typing the letter H followed immediately by a digit in the range 0-3. Any character input before the letter H is ignored.
- (3) The name CALLPM will be displayed (if the correct tape has been loaded on the specified handler and the handler is available on 'REMOTE'). Type in the name of the program required and terminate with a comma. When ENT is displayed, type the unsigned integer which specifies the entry point required (this should also be terminated by either a comma or a semicolon).

- (4) The program will search along the tape for the given name and on encountering that name will call the program into store. If the program is found and called down successfully, it will automatically be entered at the point specified in the program. If automatic entry is not required, the entry point specified should be 8121; this will cause a dynamic stop.
- (5) To call down further programs, MTBOOT can again be used. This will overwrite locations 8,7300 to 7999, and 8100 to 8179. If locations 7300 to 7999 and 8172 to 8179 have not been overwritten, further programs may be called down by entering at 7328 (binary 1110010100000) without re-using MTBOOT. By alternatively specifying 0 (zero) instead of an entry point, CALLPM will be re-entered as soon as the called program is in store.

6.4 Errors

If the name or the entry point is typed incorrectly, and the terminating comma has not yet been input, that typing can be cancelled by typing in the symbol / (slash).

(1) Input Error

If an illegal character is typed when specifying the program name, entry point, comma, or semi-colon, the message *INPUT ERROR is displayed. CALLPM is displayed and the operator should again type in the program name and entry point (this time correctly).

(2) *NP

If the specified program cannot be found on the file, the message *NP: NOT FOUND is displayed, followed by CALLPM. The operator should either input the correct name and entry point or, if the wrong tape has been used, press RESET; load the correct tape, and use MTBOOT again.

(3) **

If the program specified cannot be input, error ** is displayed. This is caused if any block (except the last one dumped) overwrites any locations between 8050 to 8179; or, if the last block dumped overwrites 8110 to 8179.

(4) Continuous Cycle (Magnetic Tape Error)

If a specified block cannot be found or is misread from the tape, the program backspaces the tape and makes further attempts to read the block, increasing the area of the tape searched. Every attempt at re-reading the correct block is signalled by an erase output on the teleprinter; the teleprinter head moves but nothing is printed. This situation will also arise if the program reads to the end of tape. If the process continues for more than 20 cycles it is never likely to be successful, and the program should be stopped, by pressing RESET. This error may be due to corrupt data on the tape, or a hardware error.

6.5 Creation of a Library Tape

Make up a new library tape by either loading CALLPM into store (e.g. by use of MTBOOT with the QFAS systems tape) and then

dumping with MSDUMP, or by copying programs using MTEDIT.

In either instance the new reel of magnetic tape to hold the library must be initialised with MTINIT to a suitable name (e.g. USERLIBRARY). CALLPM must be written as blocks 2 and 3 of the tape (Label block 2, code block 3).

In the first instance, bring CALLPM into store, and load MSDUMP (MD32 from the QFAS-SYSTEMS tape may be used; entry point at location 32). Dump program with name QCALPM as a single block of code (locations 7300 to 7999). Having made up a tape with CALLPM, programs may be dumped as and when required, by MSDUMP (Versions MD32 and MD6500 on the FAS tape may be used, these are versions of MSDUMP with entry points at 32 and 6500 respectively, occupying 1510 locations each).

Programs dumped may occupy any locations of store except 8100 to 8179 inclusive. If the program dumped is to be used as an overlay, or in conjunction with the FAS operating system, location 8 must not be dumped or used.

If a program occupies any locations in the range 8000 to 8099 these may only be included in the last block to be dumped. If they are not in the last block, CALLPM will give an error on loading the program.

The alternative method of creating a library tape is to copy CALLPM from the FAS systems tape using MTEDIT. CALLPM will be the first program on a new tape reel. MTEDIT is brought down as described in section 4.1. If the systems tape is on handler 0 and the newly initialised library tape on handler 1, with file name USERLIBRARY, the following sequence should be input:

0, QFAS-SYSTEMS, UPD, 1, USERLIBRARY, FND, 3, END,

The tape will then be ready to accept dumped programs, as previously described. Other routines from the QFAS-SYSTEMS tape may be copied by use of DEL and FND commands before END in the previous example.

A library tape as previously described can also hold programs added by the FAS command &ADDLIB. It may also hold data sections written with MTWRITE or QMWRITE. Such sections must be preceded with User Label Blocks and the programs which read them, must ignore the program label blocks and the code following.

Chapter 7: UTILITY PROGRAM MTEDIT

7.1 Purpose

MTEDIT is used to:

- (i) selectively copy data or program blocks from one magnetic tape to another,
- (ii) copy a complete tape,
- (iii) print out a list of the programs and data labels on a file.

7.2 Machine Configuration

As stated for FAS operating system, see section 1.3.

7.3 Operating Instructions

The form of the data input to the utility program MTEDIT consists of numbers and alphanumeric strings, which must be input according to the following rules.

(a) Input of Numbers

Only positive integers can be input to this routine, therefore a sign is unnecessary. The number is terminated by a separator or terminator. Input of characters will continue until a terminator or separator is found. Any character other than digits, separators or terminators will cause an error message. A separator placed within a number or a number input which is >131071, will similarly cause an error message. Separators are the characters: space, tab, newline (line feed). Terminators are the characters comma and colon. Carriage return, blank and erase are always ignored.

Example:

(L) (S) 131072 } both unacceptable
1 (S) 2 } numbers

(L) (S) 10432 acceptable number
where (L) = newline
 (S) = space

(b) Input of Alphanumeric Strings

Letters, digits and other characters in the internal code set other than the terminators may be included in a name or alphanumeric string. The input will only be terminated by a comma or colon, even when a defined limit to the length of a string exists; in which event the excess will be ignored. Spaces and newlines before the string will be ignored, otherwise they will become significant.

Example:

(L) (S) JACKAS,
(L) (S) AB47CD,
(L) (S) JAZZ30,

All names and numbers should be terminated by commas, unless otherwise stated. If AUTO is set, input is normally from the tele-printer, but if entry is made at location 33, this causes input from the tape reader.

Any numbers or identifiers input incorrectly may be cancelled by typing / (slash) before the terminating comma or colon.

The MTEDIT program is called from the QFAS systems tape as described in section 4.1. If initialising is involved, it should be performed before the loading of the MTEDIT program. If MTEDIT is to be

used simply to copy a systems tape use the instructions in section 4.1. If the WRI option is to be used the tape to be written to must be initialised to the name SCRATCH, or to the same name as the tape which has just been read. The created tape is called the object tape and the tape from which it is created is called the source tape.

The operating instructions for this program are as follows:

- (1) Call the program MTEDIT into store.
- (2) Enter MTEDIT at location 32, unless paper tape input is required. In which event entry is made at location 33.
- (3) Input one of the following two options:
 - (i) Input the handler number and file name of the tape to be copied; each terminated by a comma.
 - (ii)) (closed parenthesis). This will cause a return to CALLPM if MTEDIT was called down from magnetic tape by CALLPM.
- (4) Input one of the following three options:
 - (i) WRI, followed by the handler number of the object tape, for the creation of a new tape. Go to step (5).
 - (ii) UPD, followed by the handler number and file name of the new tape. The object tape will be opened for updating. Go to step (5).
 - (iii) LIST, this will list the file name, generation number and all the label blocks (programs) on the source tape. Return to step (3).

(5) Input one of the following six Edit Command options:

- (i) FND, n, This will copy the source tape up to and including block n (where n is an unsigned integer).
- (ii) DEL, n, will read and ignore all blocks of the source tape up to and including block n (where n is an unsigned integer).
- (iii) CPY, will copy the remainder of the source tape on to the object tape. The effect is then as for the option END.
- (iv) END, will close both tape files (source and object) and commence rewinding the tapes; on completion will output EOF. Return to step (3).
- (v) REM, will copy programs from the source tape, with the exception of programs or data sections starting with specified labels. The names of the programs or sections to be removed must be specified in a list after REM, each separated by a comma. The list is terminated by closed parenthesis()) which causes the remainder of the tape to be copied. These names must be specified in the order they occur on the source tape. The specified label and the following blocks up to (but not including the next label block) are omitted.
- (vi) KPY, n, l, is an alternative command to FND, n, The command copies all blocks on

the source tape up to and including block n. By specifying the expected average block length(1) the speed at which the blocks are copied may be increased. Blocks that exceed the expected block length are copied as before, but with more tape movement. 'n' and 'l' are unsigned integers. When copying a complete tape, specify n larger than the highest numbered block.

If there are a large number of blocks of let us say, 500 words and a number of label blocks of about 20 words or less, then l would be set to 500. The tape will then be copied rapidly, even if there are a few blocks longer than 500 words let us say, 2000 words.

To copy the FAS system tape the following format should be used:

KPY,600,600,

7.4 Format of Output for LIST

The format of the listing is as follows:

BLOCK	LABEL	TYPE
1	QFAS-SYSTEMS	4
2	QCALPM	1
3	MNRGPRG	5
10	EOF	

7.5 Error Procedures

- (1) Error Messages output are in a standard format:

*MTL n m

where n represents the status word of the magnetic

tape handler in octal form (unless otherwise specified)

m represents an error number unless otherwise specified (printed in 5 digit decimal form).

- (2) The status bits have the following meaning:

BIT	MEANING
1	Handler busy
2	Handler in MANUAL state
3	Missed data transfer
4	Parity error
5	Short block
6	Long block
7	Write permit ring fitted
8	Beginning of tape marker under read head.
9	End of tape marker detected
10	Zero character (false end of block).
11	An instruction has been rejected as 'done nothing' due to operator or hardware error

If bit 2 is set to 1, the other bits of the control word have no meaning. Bits 12-18 are undefined.

- (3) The error numbers are:

NUMBER	MEANING
1	Error in opening file
2	Error in reading block
3	Error in writing block
4	Error in closing block
5	Incorrect handler number input

NUMBER	MEANING
6	Handler not available
7	Incorrect file name (instead of 1)
8	Incorrect numbers input
9	No write permit ring
10	Indicates an unrecognised input at step (4) or (5)

Note one exception, error output is in the form:

*CHN n m

An invalid character has been input; n is the external code value of this character in octal and m refers to the entry to subroutine QCHIN.

All programs halt after an error signal, however, re-entry is made at location 9.

7.6 Example

This example demonstrates the way a library tape may be formed from a systems tape. All underlined text is that input by the operator.

```
H0
CALLPM MTEDIT,
ENT 32,
MTEDIT
0,QFAS-SYSTEMS,LIS,
BLOCK LABEL TYPE
 1 QFAS-SYSTEMS
 2 QCALPM    1
 4 QMTCAL    1
 6 QFT2R1    1
 9 QMTSYS    1
12 QFT2TB    1
14 QFORT2    1
17 QFORIP    1
19 QFORT1    1
24 MTEDIT    1
26 DBG       1
28 MNEPRG    5
54 EOF
```

EOR

1, HATCH-FILE, LIST,

BLOCK	LABEL	TYPE
1	HATCH-FILE	0
2	EOF	

EOR

0, QFAS-SYSTEMS, UPD, 1, HATCH-FILE, FND, 3, DEL, 23, REM, DBG,)

EOR

1, HATCH-FILE, LIS,

BLOCK	LABEL	TYPE
1	HATCH-FILE	0
2	QCALPM	1
4	MTEDIT	1
6	MNEPRG	5
32	EOF	

EOR

)
CALLPM

Chapter 8: STORE USED AND TIME TAKEN

8.1 Store Used

During compilation and processing of commands, the whole of the first 8192 words of store is utilised. Location 8 holds an operating system control word which is not overwritten by the user or operating system overlays (apart from locations 0 to 8; all other locations in the first 8192 words of store may be overwritten by operating system overlays). The main routines to process commands are overlayed by the compiler, loader, edit program, etc.

For an ALGOL program at run-time, the store from 4000 upwards is available for program and data. The maximum possible size of ALGOL code is 3,500 words, in fact little difference from the paper tape orientated system.

For a FORTRAN program likewise, the store available for running programs is about 80 locations less than that for the equivalent paper tape orientated system using relocatable mode of operation.

The store above 8191 is only used for loading and running 16K programs. If the store is larger than 16384, no use is made of the excess store.

8.2 Time Taken

The FAS Operating system is almost entirely paper tape orientated and/or magnetic tape speed dependent. ALGOL programs compile at around 60 statements per minute for a typical program, however, FORTRAN programs compile at around 50 statements per minute.