P D P 1 5 U S E R ' S G U I D E

Department of Computer Science

Edinburgh University

The first part of this guide is a general introduction to the Digital Equipment Corporation's PDP-15 family of computers and a description of the PDP-15 and PDP-9 installations in the Department of Computer Science.

The second part describes an implementation of a restricted version of the IMP programming language for these machines. The basic system runs on a PDP-15 or PDP-9 with Dectape as the system device. Modified versions of the system have been produced for the PDP-15 with disk and for the PDP-7. A knowledge of the IMP programming language is assumed.

revised October 1977

INTRODUCTION TO THE PDP15

GENERAL

The PDP15 model computer belongs to a family of 18-bit machines manufactured by the Digital Equipment Corporation (DEC). Its predecessor in the family was the PDP9 which was itself preceded by the PDP7. Generally speaking programs developed for the earlier machines can be run on the later model. The Department of Computer Science has both a PDP15 and a PDP9. There are unfortunately quite a number of minor differences between the control panels on the two machines but the functions of the keys and switches are broadly similar. Identical software systems are run on the two machines and the information given in the rest of this document applies to both machines unless otherwise stated.

The PDP9 is a basic machine with 8K of 18-bit ferrite core store and a standard Central Processor without options; the PDP15 has 24K of core store and is fitted with the Extended Arithmetic option. The core cycle time is one microsecond for the PDP9 and 800 nanoseconds for the PDP15 and the execution time for most instructions depends only on the number of core accesses (one, two or three) that they require. Instruction and data formats are described in the DEC User Handbook.

The principal Operating System in general use on the machines is the Department's IMP15 System. This system is designed for single-user on-line operation: it accepts control commands from the keyboard and uses Dectape as the system storage device.

The machines are run entirely on an 'open shop' basis: that is, there is no job reception and users operate the machines themselves. The software system does not provide for job identification or for dating and timing jobs. It is, therefore, the responsibility of the individual user to assemble his data, present it to the computer correctly, load the programs he wants and label his output for identification. It is particularly important that all useful paper-tapes and listings are taken away at the end of a session, since paper-ware left in the machine-room will be discarded.

Operating instructions for the peripheral devices are displayed in the machine-room. The safety precautions appropriate to all electrical and electro-mechanical equipment should be observed. In particular no attempt should be made to interfere with or forcibly stop a peripheral device when it is operating. In the unlikely event of serious malfunction, or signs of overheating or burning, the machine should be switched off at the power supply and a member of staff informed at once. Fire extinguishers are located on the side of each power supply.

SWITCHING ON

To switch on the machine, operate the key at the power supply box if the MAINS ON light is out, and turn the ON/OFF switch on the front panel of the computer to the ON position. The computer should then spring to life, with the paper-tape punch motor sounding off for a few seconds (PDP-9 only).

SWITCHING OFF

It is recommended that the PROGRAM STOP key on the front panel be pressed and the Dectape controls set to the OFF position before switching off. Turn the ON/OFF switch on the front panel to OFF and press STOP at the power supply box. The machine is normally left on throughout the day, but it should be switched off by the last user of the day, evening or night.

In an emergency simply press the STOP button on the power supply box.

KEYS AND INDICATORS

The function of all keys, switches and indicators on the Control Panel is described in Chapter 10 of the DEC User Handbook. Most of the operations which can be carried out are relevant only when self-contained machine-code programs are being run and it is a serious mis-operation to use them (particularly RESET or DEPOSIT) while one of the Operating Systems is running. An exception is the use of the STOP key to halt execution of instructions and CONTINUE to resume. Even this should be avoided while Dectape transfers are being carried out.

Sometimes when a program runs wild the machine may enter a micro-loop. In this condition the STOP key will have no effect and the RESET key must be used; on the PDP15 the STOP key and the RESET key must be operated simultaneously to clear the condition.

Note that the PDP15 has a RUN light which goes out when the computer is halted, whereas the PDP9 has a STOP light which comes on when it is halted.

INITIAL LOADING

Initial loading is carried out by means of a feature known as Hardware Read-In, which enables instructions represented in a special format on paper-tape to be read into store and executed.

The start address for the instruction sequence will usually be marked on the leader of the tape in octal; a simple mental conversion to true binary is required to set the required pattern on the switches. For the PDP15 a switch is set (=1) when the stripe on the leading edge is showing; for the PDP9 a switch is set when it is in the up (horizontal) position.

The Hardware Read-In procedure can be used to load any instruction sequence prepared in the required format, but is most used to feed in a so-called Bootstrap loader for one of the Operating Systems.

1. If the machine is off, carry out the switch-on procedure.

2. Check that the appropriate System tape is mounted on one of the Dectape drives, that the unit selector is switched to 0 (8 on the PDP-9), and that the REMOTE-OFF-LOCAL switch is in the REMOTE position. Ensure that no other drive is also switched to REMOTE on unit 0.

3. Position the paper-tape bootstrap in the reader, with leader at the read station. Check that the reader is ON (PDP-15 only).

4. Check that the address switches on the control panel are set to the correct address -- 20 (octal) for the short bootstrap.

5. Check that the user console is on-line.

6. For Dectape operation ensure that data switch <0> is set; for Filestore operation it should be in the clear position (see above for description of data and address switches, see also Filestore).

7. Depress RESET and then READIN.

The paper-tape bootstrap should immediately pass through the reader. Then the System Dectape should wind for a few seconds. As soon as it stops, the system identification message should be printed on the user console.

If the paper tape bootstrap has become torn or stained, it should be replaced.

THE FILESTORE

During the summer of 1976, the Department has acquired a high density disk drive (capable of holding about 67 megabytes). To reduce the high access times associated with linear storage media such as Dectape, the IMP15 System has been modified for operation off the filestore, which replaces Dectape to a certain extent.

As far as the user is concerned this means that logical units 0 and 2 (i.e.the system and work tapes) reside on the Filestore's disk instead of on real Dectapes. Odd numbered units remain unchanged, such that each user may still use up to four actual tapes. Units 4 and 6 are discontinued, and if used will be identical to unit 2. As the decision whether to go to Dectape or Filestore is made at the device handler rather than file processor level, "virtual" and "physical" tapes appear identical to the user and to most system components (with one exception, see below). thus the only obvious difference is that in access time and transfer rate.

Restrictions and Exceptions

- Not all even numbered units are mapped onto unique filestore tapes: DT0 is mapped onto filestore "tape" 0; DT2, DT4, DT6 are all mapped onto filestore "tape" 2.

- Users requiring "direct acess" facilities are advised that although transfers of block lengths which are not a multiple of 256 words are supported, data transfers can only take place in the forward direction, unlike real Dectape, which can be read and written backwards if desired.

- As the filestore is connected to the PDP9/15 via the inter-processor links, the device mnemonic 'LK0'' must not be used when the filestore is being used.

- Should there be a breakdown in communications between the PDP9/15 and the filestore, the operating system can be made to revert to Dectape-only operation at the flick of a switch.

NOTE: FOR DECTAPE OPERATION DATA SWITCH <0> MUST BE SET.
FOR FILESTORE OPERATION DATA SWITCH <0> MUST BE CLEAR.

Recovery Procedure

Should the filestore appear to hang up, i.e. ESC+CR and ESC+ESC have no apparent effect, the following course of action is recommended:

Set Data Switch <0> (i.e.revert to Dectape-only operation)
Set Address Switches to 00001
Press STOP, and then press START.
Type ESC+CR. If nothing happens, type ESC+CR again (up to three times). If still nothing happens, type ESC+ESC up to three times. If still nothing happens, reload the system using the paper-tape bootstrap.

Do not attempt to get back on the filestore, as normally some action will be required at the other end to clear the condition.

KEYBOARD INPUT CONVENTIONS

1. The Carriage Return key (by itself) is used to terminate a line of input.
2. The DEL key may be used to cancel the last (extant) character typed on the current line.
3. The CAN key may be used to cancel the whole of the line.
4. The EOT key is used to signify logical end-of-file.

On old-fashioned Teletype keyboards the relevant keys are:
DEL: RUBOUT
CAN: CONTROL+X
EOT: CONTROL+D

PAPER-TAPE INPUT CONVENTIONS

1. The normal paper-tape code is ASCII(ISO).
2. Physical end-of-tape is taken to signify end-of-file.
3. There is no delete character. However, the delete (or rubout) character (i.e. all holes punched) is ignored, so it may be used to ocerpunch wrong characters.
4. A carriage-return is ignored if it is immediately followed by a line-feed. Hence both LF and CR-LF designate newline.

DECTAPE FILES

Dectapes have no labels recorded on them, and a particular tape is identified to the operating system by the (logical) unit number of the deck on which it is mounted. Any physical deck may be switched to any logical unit by means of a rotary switch mounted on the deck. Thus the abbreviation DT1, for example, means the Dectape mounted on whichever deck is currently switched to unit number one. But see also Filestore section

The basic information about the way in which Dectape is used for the storage of files is given in the DEC publication 'MONITORS' under the heading 'File-Oriented DECtape'. The chief points are that, since Dectapes can be read and written on a block-by-block basis, each tape can hold a number of files. A file is recorded as an integral number of blocks, each block except the last including a pointer to the next block. Every tape carries a Directory (recorded in a fixed position on the tape) which stores a list of the files on that tape and details of block-occupancy.

An important point to note is that when the user specifies a file for output which has the same name as an existing file on the same tape, the new file does not literally over-write the existing file. It is recorded in free blocks and only when it is closed does it 'replace' the existing file - the replacement involving only Directory operations. This arrangement has the following consequences:

1. It is not illegal to specify for output a file-name which already exists in the Directory. This is, in fact, the standard method of performing updating operations.
2. The existing file is not altered until the output file is closed, so that if a job is abandoned before this stage the old file is intact and the partially created new file is lost.

These points do not apply to the special type of file known as a scratch file.

THE IMP15 SYSTEM

The IMP15 Operating System provides facilities for the running of IMP programs (including the IMP15 Compiler and other utility programs), and for the creation and manipulation of user files. These facilities are requested by means of commands provided on a control stream (usually the on-line console). The Operating System has two major components: the Supervisor, (which is permanently resident) and the Command Language Executive (which is displaced when a program is run. Both components are brought into store from the IMP system tape when the system is initialised, and control is transferred to the Executive. Provided the system is running, it can be reloaded at any time by striking the ESC (Escape) key twice in succession. Any current activity is irrecoverably abandoned.

The Executive indicates that it is ready to accept a command by typing the compound symbol ' ' on the console. It then reads a command terminated by a newline or a semi-colon. If the command is valid, the operation requested is carried out and then control returns to the Executive. If the command is faulty, an error report is output and a new command is requested.

The leading element in a command (the command 'verb') indicates which operation is to be performed and the remainder of the command specifies parameters for that operation. Some operations are built-in system functions; the remainder are implemented by loading and executing IMP programs, which may be system utilities or user programs.

COMMAND SUMMARY

Transfer:	T input / output
Background Transfer:	BT input / output
Delete:	D file!
File-list:	F unit
Newname:	N old-name / new-name
Compile:	C source / object , map
Library-select:	L object-library
Run:	R input-streams / output-streams
General commands:	object-file input-streams / output-streams
Edit:	E old-file / new-file
Scratch:	S scratch-file number
Assign:	A control-input / control-output
Set default units:	UU unit
	WU unit

STREAM IDENTIFIERS

The parameters for most commands denote input/output data streams. These parameters will be called stream identifiers. A stream identifier belongs to one of the following classes.

1. A device mnemonic:
   PR --- paper-tape reader
   PP --- paper-tape punch
   LK --- local printer
   LP --- filestore printer
   TT --- on-line console (Teletype)
   LKn --- inter-processor link
   CS --- control stream

2. The device mnemonic DTu (where 'u' indicates unit number in the range 0 to 7) followed by a file identifier, denoting an internal (Dectape) file.
   Examples: DT1 TEST1; DT7 SPEC SRC; DT0 PAL8; DT1 CS2; DT2 2
   For convenience, 'DT1' may be replaced by 'U' (for User), or omitted altogether provided that the file identifier starts with three letters; 'DT2' may be replaced by 'W' (for Work); and 'DT0' (the system unit) may be represented by a full stop.
   Examples: TEST1; .PAL8; U CS2; W2

3. An identifier belonging to class 1 or class 2 followed by a colon and a letter in brackets indicating a special mode of use.
   Examples: PP:B - binary; W2:D - Dec format

4. An equals-sign followed by a digit in the range 0 - 6. This indicates equivalencing of one stream to another; the significance is explained below (Run command).
   Examples: =0; =3

5. An identifier belonging to class 1, 2, or 3 preceeded by a special pre/post-processor mnemonic and a dash (minus). See section on pre/post-processors.
   Examples: L-LP; C-LK

6. For input only, two or more identifiers belonging to one of the above classes (except 4) connected by plus signs. This indicates a single input stream consisting of all the individual streams concatenated in the order specified.
   Examples: PR+PROG+TT; TEST1+TEST2

7. N (for None). This denotes a null file for input, and a 'sink' for output.

SPACES

The general rules for the placing of space-symbols in command strings are given below. By a 'word' is meant a command verb, a device mnemonic (including the abbreviations 'U' and 'W', but not the full stop), a file-name, a file-name-extension, or a number.

1. Spaces are illegal within a word.

IMP15 RUN TIME ERROR REPORTS

FAULT	MEANING	ADDITIONAL INFORMATION
1	multiplication overflow	multiplicand,multiplier
2	division overflow	dividend,divisor
4	storage exceeded	
7	switch label not set	index value
9	input ended	
10	non-integral quotient for '/'	dividend,divisor
13	operator intervention (non-trappable)	
14	non-numeric symbol in READ	
15	checksum error on input (the faulty data can be read after trapping this fault)	
19	illegal use of input/output stream	
24	illegal exponent	number,exponent
28	array inside-out	lower bound, upper bound
29	output exceeded	error flag
32	array bounds exceeded (also produced for switch or fault number out-of-range)	index value

In the fault report, information is also provided identifying which input and output streams were selected at the point of failure, the next symbol on the input stream (if accessed), the size of the program, and the line number at which it stopped. This is all the monitoring information that is automatically provided by the system.

If a program which terminates with a fault has a named output file, the extension of this file is altered to TMP.

For a program which terminates by executing an explicit stop statement, the stopping line number only is printed out.

For a program which stops at endofprogram, nothing at all is printed.

IMP15 COMPILER ERROR REPORTS

When the Compiler detects an error in a program, it types out the erroneous line with an indication of the nature of the fault on the following line. For the first three faults listed, it also prints an up-arrow to identify the point within the statement at which the error was detected.

*FORM	- incorrect statement format (or excessively long statement)
*ATOM	- incorrect statement component (eg keyword)
*NAME	- name not declared
*DUPLICATE	- name or label declared twice
*BOUNDS	- illegal bounds in switch or own array declaration
*INDEX	- index value of switch label outside stated bounds
*CONTEXT	- formally correct statement in wrong context (eg return not within a routine)
*MATCH	- parameter mismatch
*%BEGIN MISSING	- unmatched end
*%START MISSING	- unmatched finish
*%CYCLE MISSING	- unmatched repeat
*%END MISSING	- premature endofprogram
*%FINISH MISSING	- unmatched start at end of procedure
*%REPEAT MISSING	- unmatched cycle at end of procedure
*%RESULT MISSING	- result not specified in function or map
*'...' MISSING	- label or procedure not found before end
*ACCESS	- instruction cannot be executed (not illegal, but may indicate another error)
*ORDER	- formally correct statement in wrong sequence

Certain kinds of error cause compilation to be abandoned as a result of a run-time fault in the Compiler. An example is Fault 9 -- indicating that the endofprogram statement has been omitted from the input file; or Fault 32 -- indicating that the capacity of one of the compiler's tables has been exceeded.

IMP15 SYSTEM ERROR REPORTS

Supervisor report

dev NOT READY - the device specified is not operable. This report may indicate a device mal-function or faulty medium, a mis-operation, or a user error. To re-try the transfer, the user should ready the device and respond with Carriage Return. The conditions which can cause the report to be produced for a particular device are described in the relevant operating instructions.

Executive reports

COMMAND l ?	- unknown command letter
DEVICE d ?	- unknown or illegal device mnemonic
?	- faulty command format
NO	- attempt to carry out a privileged operation, for example to delete files on the system tape.
f NOT ON DTu	- file specified does not exist.
f ALREADY ON DTu	- file specified in a Scratch command already exists.
NO NEW DTu f	- named output file not entered in directory -- usually because nothing has been written to it.
DATA ERROR	- parity or checksum failure on transferring input file; the transfer is abandoned.
FILE TOO BIG	- (a) after Transfer command -- no free blocks for named file or capacity of scratch file exceeded (b) after Scratch command -- not enough free blocks at fixed intervals on the tape to accommodate the size of file requested.
I.O. AREA FULL	- store available for input/output routines and buffers not sufficient for files requested.
DTu DIRECTORY FULL	- no room in directory to enter new file name.
DTu CORRUPT	- inconsistency in directory information; a salvage operation should be mounted immediately.

Loader Reports

SYSTEM FAULT 1	- faulty format in object program
SYSTEM FAULT 2	- [compiler error]
SYSTEM FAULT 3	- error in program which prevents loading
SYSTEM FAULT 4	- not enough space in core for program
SYSTEM FAULT 5	- switch label set twice (not detected by compiler)
SYSTEM FAULT 6	- external reference not satisfied (or other library fault)
SYSTEM FAULT 9	- compiler output file too big

PERMANENT PROCEDURES

The following standard procedures are implemented:

read ch, print ch, read symbol, print symbol, next symbol, skip symbol, space, spaces, newline, newlines, read, write, select input, select output and nl.

In addition the following non-standard procedures are provided:

routine close input
closes the currently selected input stream.

routine close output
closes the currently selected output stream.

integerfn instream
the stream number of the currently selected input stream.

integerfn outstream
the stream number of the currently selected output stream.

integerfn indev
the type of the device associated with the currently selected input stream, coded as follows: 0 None; 1 Console; 2 Dectape file; 3 Other.

integerfn outdev
the type of the device associated with the currently selected output stream, coded as above.

integerfn outpos
the number of symbols on the current output line

integerfn input
the input-available status of the current input stream. The value is zero if and only if the stream is associated with the on-line console and no data is available or pending.

routine prompt(integer k)
causes the symbol k to be printed on the on-line console as a prompt to the user provided that the currently selected input stream is associated with the console and no input is available or pending. Otherwise it has no effect.

integerfn tint
the (ASCII) value of any interrupt character that has been typed in following ESC. A zero value is returned if no interrupt has been given since the start of the program or the last call of 'tint'.

integerfn free store
the number of words of store currently unoccupied.

IMP LANGUAGE RESTRICTIONS

The major restrictions are severe but simply stated:

1. The only data-type permitted is integer.

2. The only data-structure permitted is the one-dimensional array.

This means that reals, strings, records and multi-dimensional arrays may not be used. The interpretation of integer is in relation to an 18 bit operand length (permitting a range of -131072 to +131071). byteinteger is implemented, and shortinteger may also be specified but is not treated differently from integer.

In addition there are these lesser restrictions:

3. The control variable in a cycle statement must be a scalar (not an array element).

4. The implementation of hexadecimal, octal and binary constants is non-standard as indicated by the following examples (standard form in parentheses):

```
16_FFFF      (X'FFFF')
8_177777     (K'177777')
2_10101      (B'10101')
```

Otherwise the version of the language implemented includes all the features taught in first year with the addition of for clauses, and predicates.

INPUT/OUTPUT PROGRAMMING

A user program may access data on up to 8 input and 8 output streams (numbered 0 to 7). The way in which actual data streams are associated with the program's logical streams has been described above (Run command). The only effect of calling the 'select input' ('select output') routines is to establish which stream is to be used by the next input (output) instruction. Streams are implicity opened by the first input or output instruction affecting them; they may be explicitly closed by means of the 'close input' ('close output') instruction. If a reprocessable stream is re-opened after being closed, transfer starts again from the beginning of the stream.

Programs are entered with input 1 and output 1 selected.

It is strongly recommended that, in programs making use of multiple input or output streams, the stream numbers should be referred to symbolically by means of constinteger variables declared at the head of the program.

File Processors

A File Processor is a routine which reads or writes files in a particular format on a particular medium or device. All File Processors transmit information in the form of character-streams and the basic File Processors convert, where necessary, between device-dependent representations (like the IBM 29 card code) and the ASCII(ISO) code, which is the standard internal code. Normally IMP programs operate within the information subset of this code using the 'symbol' input/output routines. If the full set (including control characters) requires to be handled, the character-handling input/output routines 'read ch' and 'print ch' should be used instead of the symbol-handling routines.

A number of special File Processors are available which enable non-standard character sets to be handled. The 'character' routines should also be used in IMP programs for input and output of non-standard character sets.

The requirement for a file processor other than the default is indicated by the presence of a mode letter in parentheses in a device mnemonic. The following list shows which modes are available for particular devices.

file:D - the seven-bit ASCII format used by the DEC system. This format can be used for information interchange between the DEC system and the IMP system. D mode is assumed by default for a file with extension SRC.
Examples: DT2 PROG MAC :D, NOTES:D, W2:D, FORMAT SRC

file:B - binary -- arbitrary eight-bit values. B mode is assumed by default for a file with extension ABS.
Actually, arbitrary 9-bit values may be used, provided the value 511 is not used, which is interpreted as logical end-of-file.

file:F - full-word -- arbitrary eighteen-bit values. On output full-word files are padded out with zeros to a multiple of 255 characters (ie words).

PR:B - binary -- arbitrary eight-bit values.

PP:B - binary -- arbitrary eight-bit values.

PP:V - visual -- legible characters.

LKn:B - binary -- arbitrary eight-bit values.

SYSTEM COMPONENTS

The Supervisor

The Supervisor is permanently resident in store while the system is in use. It consists of a set of routines which provide services for the Executive and for user programs. Its functions include interrupt-processing, handling transfers on the system device (Dectape), handling input and output on the on-line console, reloading the other system components and reporting error conditions affecting peripheral devices.

The Executive

The Executive accepts and acts on commands input from the control stream. It is the custodian of a collection of File Processors from which it assembles input/output packages for use by the Compiler or a user program. After processing a command which calls for the running of an IMP program, the Executive passes control to the Loader. All file directory operations are carried out by the Executive. As some of these are not completed until a program terminates, it is important not to abandon a job before control is returned to the Executive.

The Loader

The Loader is a routine which loads into store object programs produced by the Compiler. It is designed as a separate system component, rather than as part of the Executive, chiefly because the Executive is overwritten as a program is loaded. The space occupied by the Loader itself puts a limit on both the (static) length of a user program and the size of its input/output package. However, when loading is complete and control is transferred to the program, this space is available for dynamic allocation for program work-space and input/output buffers.

INTERNAL FILES

Internal text files are recorded on Dectape (but see also Filestore section). A number of system files are held on the system tape (mounted on unit 0) and the general practice is that each user has a private Dectape (conventionally mounted on unit 1) on which user files are recorded. A file is identified to the Executive by specifying a Dectape unit number and a file identifier. There are two kinds of file identifiers corresponding to two kinds of internal files: named files and scratch files.

Named files are used for storing all information of a permanent or semi-permanent nature. The file identifier for a named file consists of a file-name (up to six letters or digits starting with a letter) and an optional file-name-extension (up to three letters or digits). The file-name-extension may be employed at the user's discretion to distinguish, for example, between different versions of a file. It should be noted that the file-name-extension is an integral part of the file identifier and that there is no necessary connection between files with the same file-name but different file-name-extensions. A named file is created when it is used as an output stream, although its identifier is entered in the Directory for the tape to which it is written only when output is complete. At this point any existing file on that tape with the same identifier is automatically deleted.

Scratch files are used for temporary storage within a single run of a program or between runs of the same or different programs. The use of scratch files may also be dictated by the technical considerations noted below. The file identifier for a scratch file consists of a number (more precisely, up to six letters or digits starting with a digit) rather than a name plus extension. A scratch file is created (but not initialised) by the Scratch command which reserves a number of blocks at fixed intervals on the tape specified for the file. The size of the file is at the discretion of the user. Scratch files are in effect fixed storage sites, with the consequence that as soon as any output is written to a scratch file, the previous contents (if any) are lost. Scratch files have the following advantages over named files:

1. Because of an operating system restriction, a program cannot use more than one named file for output, but may use any number of output scratch files.
2. Output written to a named file is not re-processable within a single run of a program, while output written to scratch files may be read back if required.
3. The File Processor for an output scratch file occupies less space than the File Processor for an output named file.

The scratch file facility is peculiar to the IMP15 system, but compatible with the DEC system in respect of file directories.

IMP CONSOLE CONVENTIONS

1. DEL: cancels the last (extant) character typed on the current line (printback: '\').

2. CAN: cancels the complete line (re-prompt '_').

3. EOT: represents end-of-file.

4. SO - Shift Out: causes subsequently typed lower case letters to be converted to upper-case. (set initially)

5. SI - Shift In: restores availability of full character set including lower-case letters.

6. DLE: causes immediately following character to be treated as data(that is, without control significance).

7. Carriage Return: finishes a line of input (the system supplies the Line Feed). The data transmitted to the program is terminated by Newline. The Line Feed is not issued until subsequent input (or output) is required. Thus it acts as an automatic prompt. IMP programs may also issue an explicit prompt character to request input from the keyboard.

8. A line of input may be finished by typing Line Feed instead of Carriage Return. Again the data transmitted to the program is terminated by Newline but keyboard input remains enabled so that the user may continue typing while the earlier data is being processed.

9. The user may also type ahead after finishing a line with Carriage Return, or after a line of output, by issuing a Line Feed instead of waiting for the system to supply it.

10. The Escape character (ESC) acts as an interrupt character and causes a question-mark to be printed out. The next character typed is treated as an interrupt message. The interrupt message can be accessed in IMP programs using the function TINT (see Permanent Procedures).
    Escape followed by Carriage Return causes the current activity to be terminated and control to be returned to the Executive; if given when an IMP program is running, it causes Fault 13 to be signalled.
    Escape followed by Escape causes the complete system to be reloaded.
    On Teletypes fitted with an ALT MODE key, the Escape code can be generated by CONTROL+SHIFT+K.

11. If a key on the keyboard is struck while console output is in progress, output sent to the console is discarded until the next input request or a key on the keyboard is struck again, or if a special character (STX) is contained in the output.

Scratch command: S scratch-file number

Examples: S U3 25 :create '3' on DT1 (25 blocks)
S DT2 999 50 :create '999' on DT2 (50 blocks)

Effect: A scratch file bearing the indicated file identifier is created on the Dectape specified. The file identifier must begin with a digit. The number specifies the number of blocks required in decimal. The normal interval between blocks is 5. A non-standard interval may be specified by supplying an additional (decimal) number following the number of blocks. See also INTERNAL FILES.

Assign command: A control-input / control-output

Examples: A PR/LP
A W1/TT
A TT/LP

Effect: This command permits the assignment of alternative devices to the control input and output streams (normally both the on-line console). The main use of the command is to allow for a limited form of batch-processing, with the Executive reading pre-prepared commands from a batch input file. When a device other than the on-line console is specified for control input, the input from the device is treated as a batch control stream. In a batch control stream all Executive commands must be preceded by one or two greater-than symbols ('>'). In the event of an error, the Executive skips to the next command starting with two such symbols.

The altered assignments continue until another Assign command is encountered or end-of-file is reached on the control stream.

Set default Dectape unit: UU unit or WU unit

Examples: UU DT3
WU DT1

Effect: These commands set the default user (UU) or work (WU) tapes to the specified unit.

Edit command: E old-file / new-file

Examples: E TEST1
E COPIER V4/COPIER V5
E TEST2/N
E N/PROG5

Effect: This command calls the standard system Editor (the PDP-15 version of the Edinburgh Compatible Context Editor). The input stream identifier indicates the file to be edited and the output stream identifier the name to be given to the edited file. If no output identifier is specified (as in the first example), the new file will have the same name as the old file (and the old file will be lost when editing is complete). The third example is a form which can be useful when it is desired just to inspect a file without modifying it. The fourth example illustrates the normal way of creating a new file from the console (edit command G*); it has the advantage over the more straightforward T TT/PROG5 that any mistakes noticed before the end of the operation can be corrected immediately.

A general account of the context editor and the form of the editing commands is given in a separate document. The following paragraphs describe features of the PDP-15 implementation.

Editor Command Input

The prompt for a command is the symbol '>'; the prompt for a line of input (using Get) is the symbol ':'.

A command may extend over more than one physical line provided that it is broken immediately after a comma.

Editor Special Commands

%C Close - the edited file is closed and the editing operation terminates.

%M Monitor - normal monitoring is switched on; that is, the current line is typed out after execution of each command line, unless the line has already been typed. (set initially)

%F Full - full monitoring is switched on; that is, the current line is typed out after execution of each command line unless the last command executed was a Print command.

%Q Quiet - monitoring is switched off.

Run command: R input-streams / output-streams

Examples:
```
R
R PR/LP
R TESTA,TESTB/PP,W1,W2
R DATA4/LK,=1
```

Effect: This command causes the last program compiled to scratch file 9 on the work tape to be loaded and entered with the data streams specified set up as assignments to the logical streams (numbered 0 to 7) used in the program. The correspondence is established by the order in which the stream identifiers are given. In the third example above this would be:

input		output	
CS	0	0	CS
TESTA	1	1	PP
TESTB	2	2	W1
		3	W2

Note that the stream identifiers typed in are matched off starting with stream 1 rather than stream 0, since it is rarely necessary to alter the default assignments for input 0 and output 0.

The default assignment for input 1 is to equivalence it to input 0, and similarly for output 1. Other streams may be equivalenced by explicit assignment, as in the fourth example; the reference must always be to a lower-numbered stream. The effect of equivalencing is to associate two or more logical streams with the same actual stream. The default assignment for stream 2 and upwards is None.

When a program is being loaded for running, any external procedure specs are referenced with respect to two libraries -- the user library (if defined) and the system library. The user library takes precedence.

General commands: object-file input-streams / output-streams

Examples:
```
.LIST TEST1
.SOAP PROG/NUPROG
TEST2 PR/LP,TT
```

Effect: The effect of a general command is the same as for a Run command, except that the program to be run is specified as the command verb, instead of being implicit. For a named file the extension OBJ is assumed -- it must not be included in the command. Thus the first example calls for file LIST OBJ on the system tape to be run, and the third example calls for file TEST2 OBJ on DT1 to be run.

Compile command: C source / object , map

Examples:
```
C PR
C TEST1
C TEST1/,LK            :map to local printer
C TEST2/TEST2 OBJ      :object code to named file
C LIB4/LIB4            :ditto
```

Effect: This command causes the source program specified to be compiled by the IMP15 Compiler. The Compiler implements a restricted version of the IMP programming language. It translates programs into a private text representation of machine-code instructions which is subsequently read in by the IMP15 Loader. The object code produced is written to the first output stream (default assignment: scratch file 9 on the work tape) and the program map is produced on the second output stream (default assignment: None). Error reports are sent to the report stream. Note the odd-looking comma in the third example, which has the effect of leaving the default assignment for the first output stream, while explicitly specifying the second.

The default assignment (W9) for the object code is appropriate when it is not required to retain the compiled program for any length of time. The program remains available for running (by the Run command) until another program is compiled. When compiling to this file the work tape must be on WRITE ENABLE.

When it is required to keep a copy of the compiled program for subsequent use or when compiling a library, a file identifier may be explicitly specified for the object code, as in the fourth and fifth examples. Named object files are automatically given the extension OBJ. If an extension is given explicitly in the command it will be overwritten or, if no extension is specified, it will be added. In the fifth example, for instance, the object file created will be LIB4 OBJ not LIB4. In this case the compiled program is available for running (in a general command by citing the object file name) as long as the file remains in existence. This avoids the need for repeated re-compilation of a frequently used program. Note that scratch file 9 on the work tape is used for a temporary version of the Compiler's output whenever the final destination is a named output file.

Library-select command: L object-library

Examples:
```
L LIB4 OBJ
L U1
```

Effect: This command defines the user library to be used in subsequent program-loading operations. The library consists of a set of precompiled external routines. The selection remains in force until another Library-select command is given, or the system is re-loaded.

Transfer command: T input / output

Examples:
```
T PR/PP
T PR:B/PP:B                    :binary
T W2/LP
T TEST1+TT+TEST2/TEST3         :concatenated input
```

Effect: The input stream is transferred character by character to the output stream. The default assumption for input is the control input stream and for output is the control output stream (normally the on-line console).

Background Transfer command: BT input / output

Example: BT PROG/LP

Effect: This command has the same effect as the Transfer command except that the user may carry out other operations (editing for example) while the transfer is going on. Concatenated input files are not permitted for this command nor is a named output file.

Delete command: D file!

Examples:
```
D DT2 PROG MAC!
D U3!
```

Effect: The file-identifier specified is deleted from the file directory on the Dectape specified. Note the requirement to type an exclamation mark at the end of the command, as a safeguard against error.

File-list command: F unit

Examples:
```
F DT6
F U
```

Effect: All the file identifiers recorded in the file directory of the Dectape specified are listed on the control output stream.

Newname command: N old-name / new-name

Examples:
```
N DT1 PRIG1/DT1 PROG1
N DT3 TEST2/DT3 PROG1
```

Effect: The file identifier specified by 'old-name' is changed to 'new-name'. Note that the (same) Dectape unit must be indicated for both names.

Pre- and post-processors

These may be thought of as filters interposed between a file processor and the user program. A pre-processor sits between an input file processor and the user program; a post-processor between the user program and an output file processor.

It is perhaps worth noting that some of the implicit file processors are implemented in terms of pre/post-processors and some more basic file-processors.

For example:
```
PR = C-PR:B
PP = P-PP:B
LK = E-LK:B
```

Here is a list of the pre/post-processors implemented:

Name	Type	Description
E	in	Generates an end-of-file upon recognising the EOT character.
E	out	Inserts an EOT character on closing output.
C	in	Strips parity bits and converts CR-LF to NL.
P	out	Generates even parity.
C	out	Inserts CR before every LF.
S	out	Strips newlines.
L	out	Inserts line numbers and paging newlines. This is useful for obtaining line-numbered listings without the need to run a program to do the job. Examples: "T file/L-LP" or "BT file/L-LP"
M	both	Causes each symbol read (or written) to be echoesd on the current control output stream.
0	both	This pseudo-processor causes the following stream list to be assigned starting with stream 0 instead of stream 1. For example: "C prog/0-LP,N" sends fault messages to the line printer.

2. At least one space is obligatory between two adjacent words.

3. Spaces are optional in all other positions.