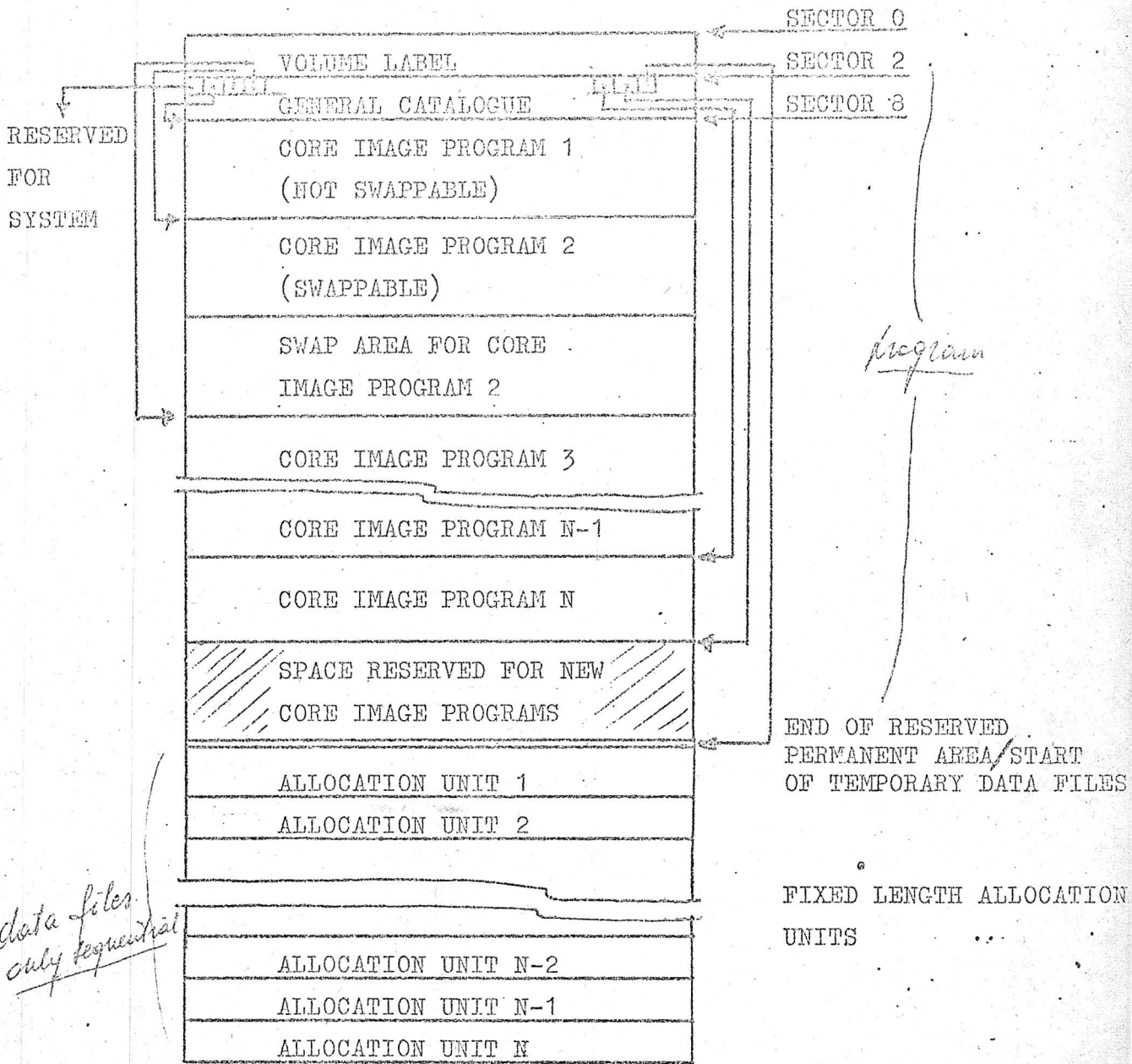


Disc Organization

The disc contains two types of files: core image foreground programs and temporary data files.

At disc initialization time, the user can define how much disc space is to be reserved for each of the two types of programs.

The layout of the disc is as follows:

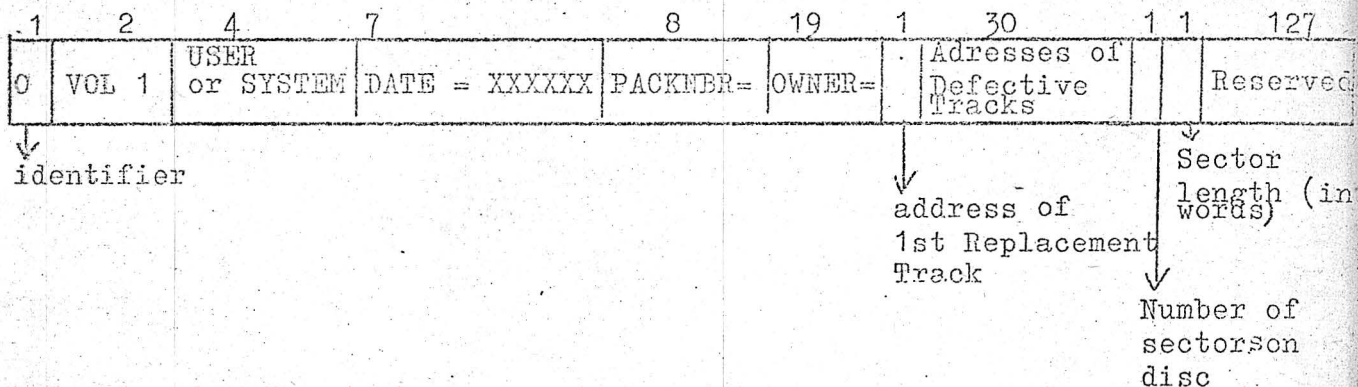


Note: The General Catalogue contains entries related to the core image programs. These entries are described on the following page.

Monitor is also on disk preferred in data file so it is early read by IPL but IPL cannot be on disk.

Volume Label

The volume label contains the following information:



This information is transmitted to the disc at disc initialization time. The defective track table is kept by the monitor.

General Catalogue

The General Catalogue consists of a table of entries related to the core image programs saved on the disc.

Each entry is 8 words long and has the following format (except for the first entry which contains internal monitor information):

PROGR. NAME	T P E	-	BSA	ESA	- -	ACA
0	1	2	3	4	5	6

where

PROG. NAME is the name of the core image program name, consisting of up to six ASCII characters.

TYPE indicates in the left character of this word, whether this program is swappable: S

not swappable: B

BSA is the absolute beginning sector address on disc

ESA is the absolute ending sector address on disc.

ACA is the absolute memory address for loading.

Core Image Programs

Core image programs can be stored on the disc by means of the operator control message SA (Save).

These programs occupy the first part of the disc, up to an address defined at system generation time. Each core image program has an entry in the General Catalogue.

These programs can be swappable, once they have been loaded into the dynamic memory partition. When the user stores a core image program on the disc (SAve), he must declare whether this program is swappable or not. If it is, it may not contain any internal buffers or ECB areas, but must have reserved space for them in the buffer pool ('Get Buffer' monitor request).

Note: These core image programs are always loaded at an absolute memory address, as indicated in the last word of the General Catalogue entry for the program. Therefore, if the user defines his memory partitions, he must take care that the beginning address of the dynamic partition is the same, or at least a lower address, as the address in the ACA word of this program's General Catalogue entry, and that the dynamic partition is long enough to contain his programs. If this is not possible, the user must rebuild his core image programs on disc, by reloading them and doing another SAve operation.

Temporary Data Files.

The disc area reserved for storing temporary data files is divided into allocation units. At system generation time the user must define the size and number of these allocation units. To the temporary data files which are to be stored in this area he must assign file codes in the range from FO to FF. These file codes have been reserved especially for temporary data files.

At the first write operation for a file, the system will assign an allocation unit to the file code specified in the ECB.

If more than one allocation unit is needed, the system will extend the file with further units. The system keeps a table of the allocation units assigned to particular file codes. When the user deletes a temporary data file (monitor request 'Delete Temporary File on Disc') the system releases the allocation unit(s) belonging to its file code and thus they become free again to be reassigned.

When a temporary data file is written onto disc its loading address is always the first address of the first allocation unit assigned to it, i.e. if an allocation unit is not filled up to the end by a file, the remaining area of that unit is not used for another file.

OPERATION

Loading a Program

By means of the operator control messages LD and LL it is possible at any time to load programs into memory. In the message the operator can specify the partition into which the program must be loaded. Then the program is read from the input device in relocatable binary format and placed in the partition specified. For each program loaded, the monitor builds a Program Control Table (PCT) in memory.

When program loading is completed, the program name, beginning loading address and the address of the next free location are printed on the operator's typewriter.

The loader computes the beginning loading address according to the following rules:

- the base address is either the next free location in the partition or the beginning address of the partition. The latter is always the case when the program is loaded into the Dynamic Area (P6) or into the Background Area (P7).
- before loading the first word of a fore ground program, the loader automatically reserves a 16-word save area, of which the address is placed in the Program Control Table. This area is used by the monitor to save the PSW and the contents of registers A0 to A14 when the program is interrupted.

Both the ST operator control message and the 'Activate' monitor request cause loading and entering of disc-resident core-image programs into the Dynamic Area. This may result in a swapping procedure in case the activated program is the disc-resident task activated with the highest priority, i.e. then the running program is swapped out.

The monitor request 'Activate' can be used to load segmented disc-resident programs into the dynamic area. The program segments must then be self-contained, i.e. not make any external references, except to a common area in memory (acquired by 'Get Buffer' monitor request). The first program segment reserves this area, activates the next segment (disc-resident) and transmits the address of the common area. At the end of this segment a return is made to the first one, and another segment may be activated, etc.

Starting a Program

As described above, programs are started by the dispatcher according to priority, so they must first have been connected to a level and be activated. A program can be activated in one of the following three ways:

- for interrupt level programs: by an interrupt.
- for software level programs: - by operator control message
 - by 'Activate' monitor request.

OPERATOR CONTROL MESSAGES

There are a number of control messages which the operator can enter via the typewriter. To do so, he must first press the interrupt button on the control panel of the CPU (INT). The system then types out M: and the operator can type a control message.

Such a message consists of two characters identifying the message, followed by a space, possibly followed by one or more parameters, followed by CR -- LF. The parameters are separated by commas. Some of these parameters are optional. When a parameter at the end of a message is not used, the last parameter may not be followed by a comma, e.g.

CC PAR1

When a parameter which is not located at the end of a message remains unused, it must always be replaced by a comma, e.g.

CC PAR1,(PAR2),PAR3 or CC PAR1,,PAR2

Note:

- Every numerical value in an operator control message must be a hexadecimal value, for example X'52' or /34.
- If a message contains an error, the system types out a message ER. The operator may then press the INT button and type in a correct message.

In the following paragraphs, the syntax and use of the available messages are given. In the descriptions of the syntax, Backus Normal Form is used for the notation, i.e.

| means: or

[] means: optional component; any or all items within these brackets may be omitted: [+ | -] <integer> can mean +<integer>, -<integer> or <integer>.

[] means: alternative components; one of the items within these brackets must be selected: [+ | -] 426 can mean +426 or -426.

< > means: these brackets contain a syntactic item.

␣ means: space.

Some operator control messages can be used only under the Disc Real Time Monitor, others under the DRTM as well as under the DOS monitor. In the following paragraphs, the messages are listed in alphabetical order according to their mnemonics and for each message an indication is given under which monitor it can be used.

MESSAGE	MEANING	USABLE IN		PAGE
		DOS	DRTM	
AB	Abort a Program	X	X	
AS	Assign a File Code	X	X	
CL	Connect a Program to a Level		X	
CT	Connect a Program to a Timer		X	
DL	Disconnect a Program from a Level		X	
DM	Dump Memory	X	X	
DP	Define Partitions		X	
DT	Disconnect a Program from a Timer		X	
HD	Halt Dump	X	X	
LD	Load a Program	X	X	
LL	Load a Program from a Library		X	
MC	Manual Device Control	X	X	
PS	Pause	X	X	
RD	Release Device	X	X	
RP	Release Partition		X	
RS	Restart	X	X	
RY	Retry an I/O operation	X	X	
SA	Save a Core Image Program		X	<i>= not in BRTM</i>
SC	Set Clock		X	
SD	Set Date		X	
SS	System Status		X	<i>= " " "</i>
ST	Start	X	X	
WM	Write into Memory	X	X	

TABLE OF OPERATOR CONTROL MESSAGES

Abort a Program (DOS and DRPM)

syntax: AB<name>

use: This message definitely stops a program which is running.
<name> is the name of the program which is to be aborted.
Under the DOS monitor or in the case of the background program,
under DRPM, no name need be specified.

example: AB PROG

The foreground program named PROG is aborted.

Assign a File Code (DOS and DRPM)

For DOS:

syntax: (a) AS<file code>,<device address>
(b) AS<file code>,<disc unit file code>,<file name>

use: By means of this message a new file code can be created or the
assignment of a previously created file code can be modified.
It can be used to assign a file code to a physical non-disc unit
(a), or to a permanent area on disc (b).
<file code> is the file code to be assigned, consisting of two
hexadecimal characters between 01 and 7F.
<device address> is the physical device address of a non-disc unit
to which the assignment is made.
<disc unit file code> is a reserved system file code, with the
value X'0A', X'0B', X'0C' or X'0D'.

For DRPM:

syntax: AS<file code>,<device address> [F]

use: By means of this message a new file code can be created or the
assignment of a previously created file code can be modified.
<file code> is the file code to be assigned, consisting of two
hexadecimal characters between 01 and 7F.
<device address> is the physical address of the device to which the
assignment is made and consists of two ASCII characters for the
device name and two hexadecimal characters for the device address,
as defined at system generation time.
F if specified, indicates that this file code is reserved for the

foreground.

The following device names can be used:

TR: ASR tape reader
TP: ASR tape punch names can be used:
TY: operator's typewriter
PR: punched tape reader
PP: tape punch
LP: line printer
CR: card reader
CP: card punch
MT: magnetic tape
TC: magnetic tape cassette
NO: no device; an operation on this file will have no effect.

example: AS 08,LP02,F

File code 08 is assigned to the line printer with physical address 2 and it will be used only by foreground programs.

Connect a Program to a Software Level (DRTM only)

syntax: CL, <name>, <level>

use: This message is used to connect a program to a software level. <name> is the name of the program which is to be connected. It consists of up to 6 characters, the first one of which must be a letter.

<level> specifies one of the software priority levels 50 to 62, to which the program is to be connected. It must be a hexadecimal number from 32 to 3E.

A background program is automatically connected to level 63 (default value).

The software level will start with

- the operator control message ST, or
- the monitor request Activate.

example: CL PROG,34

The program named PROG is connected to level 52 (hexadecimal 34).

Connect a Program to a Timer (DRIM only)

syntax: CT L<name>,<number1>,<number2>,<number3>

use: By means of this message the program <name> can be connected to
- the timer indicated by <number1> , to be reactivated every number
of cycles of that timer, as indicated by <number3>; or
- the absolute time HH MM SS, to be reactivated every time the
timer specified by <number1> has made the number of cycles
specified by <number3>.

<name> is the name of the program which is to be connected.

<number1> is the number of the timer (0 is assumed to be the
real time clock).

<number2> is the number of cycles of the timer before the first
activation (default value is 0) or, if the time of activation
is to be given in absolute form, this has to be specified here
as follows: HHLMMLSS

<number3> is the pulse rate, i. e. the number of cycles of the
timer between two activations. Default value is 1.

A program can be activated only if it has been connected to a
level first.

examples: CT PROG,1,3,4

The program named PROG is connected to timer number 1. The number
of cycles before it will be activated for the first time is 3
and it will be reactivated every 4 cycles of that timer.

CT PROG,04 12 15,3

The program named PROG is connected to timer number 1. It is
to be activated at absolute time 4 hours, 12 minutes, 15 seconds
and will be reactivated every 3 cycles of timer number 1.

Disconnect a Program from a Software Level (DRIM only)

syntax: DL L<name>,<level>

use: This message causes the program specified to be disconnected
from its software priority level.

<name> is the name of the program which must be disconnected.

<level> is the software level number (2 hexadecimal charac-
ters) from which the program must be disconnected.

If the program was also connected to a timer, it is disconnected as well.

This request for disconnection is accepted only if the program specified has made its exit first (in particular I/O operations).

example: DL PROG,34

The program named PROG is disconnected from software level 52, to which it had previously been connected.

Dump Memory (DOS and DRTM)

syntax: DM <address1>,<address2>

use: This message causes a memory dump on the print file, which usually is the line printer.

<address1> is the beginning address of the memory dump (up to 4 hexadecimal characters).

<address2> is the ending address of the memory dump (up to 4 hexadecimal characters).

example: DM 4F,3FE

The contents of memory will be dumped from hexadecimal address 4F to 3FE.

Define Partitions (DRTM only)

syntax: DP <p1> / <p2> / <p3> / <p4> / <p5> / <p6> / <p7>

use: This message is used to define the number of memory partitions and their lengths. Note that programs in one partition cannot modify programs in another partition.

Up to 7 partitions can be defined:

<p1> number of pages of lk words for the buffer pool partition.

<p2> to <p5> specify the number of pages of lk words for the resident foreground partitions.

<p6> is the number of pages of lk words for the dynamic foreground partition.

<p7> is the number of pages of lk words for the background partition.

example: DP 3/2,1/4/2

The memory will be partitioned as follows:

P1	P2	P3	P6	P7
3k	2k	1k	4k	2k

DP 3/2/6

This results in the following partitioning:

P1	P2	P7
3k	2k	6k

(no dynamic foreground partition)

DP 3/3,2/4/

This results in the following partitioning:

P1	P2	P3	P6
3k	3k	2k	4k

(no background partition)

DP ///5.

This results in:

P7
5k

(background partition only)

Disconnect a Program from a Timer (DRTM only)

syntax: DTL <name>,<number>

use: This message is used to disconnect a program from a timer.
<name> is the name of the program which is to be disconnected.
<number> is the number of the timer from which the program must be disconnected.

example: DT PROG,3

The program named PROG is disconnected from timer number 3.

Halt Dump (DOS and DRTM)

syntax: HD

use: If this message is given the output of a memory dump can be stopped. This can be very useful if the dump is taking place on the operator's typewriter, as this is a very slow device.

Load a Program (DOS and DRTM)

syntax: LD, [<value>] , [<number>] , [<level>] , [M]

use: This message causes a program to be loaded into the partition specified. It is loaded from the standard object input device and connected to a level (if specified). <number> is the number of the partition into which the program is to be loaded. If not specified, the program will be loaded into the background area. In this case, the parameters <level> and M are irrelevant and need not be specified. <value> is the displacement value, relative to the beginning address of the partition, in hexadecimal. If <number> is specified, the program is loaded as many words after the beginning address of the partition as specified here in <value>. If <value> is not specified, the program is loaded behind the last program in the partition specified.

<level> specifies the software level to which the program is connected (two hexadecimal characters from 32 to 3E).

M if specified, indicates that the program to be loaded is written in master mode. If not specified, it is in user mode.

Every time an object program is loaded from a non-disc device, the program identification, its loading address and its length are printed out on the typewriter.

If a program has not previously been connected to a level, a message (NC) is output on the typewriter.

example: LD

The program is loaded in the background area.

LD , 2

The program is loaded into partition 2 behind the program loaded last.

Load a Program from a Library (DRTM only)

syntax: LL, <name> , [<file code>] , [<value>] , [<number>] , [<level>] , [M]

use: By means of this message a program can be loaded from a library, e.g. a cassette tape.

<name> is the name of the program to be loaded.

<file code> is a number of 2 hexadecimal digits specifying the file code of the file which contains the library.

The other parameters have the same meaning as ~~is~~ the operator

Manual Device Control (DOS and DRTM)

syntax: MC␣<device address>,<order>

use: This message can be given if the operator wants to do a manual operation on a magnetic tape device.

<device address> is the physical address of the tape unit, consisting of two hexadecimal characters.

<order> must be one of the following hexadecimal numbers, each one of which indicates a specific operation:

X'31': rewind to load point

X'32': rewind and unload

X'33': backspace one block

X'34': skip forward one block

X'35': search tape mark forward

X'36': search tape mark backwards.

Pause (DOS and DRTM)

syntax: PS␣<name>

use: This message causes a program to be stopped temporarily.

<name> is the name of the program which is to be put in pause state and need be specified only under DRTM for foreground programs. To restart the program, the operator must give an RS message.

Note: Under the DRTM, for swappable disc-resident programs, this message may result in a swapping operation.

example: PS PROG

The foreground program named PROG is put in pause state.

Release Device (DOS and DRTM)

syntax: RD␣<device address>

use: When the monitor has typed out a PU error message (peripheral unit failure) after an I/O operation and thus requests operator intervention, the operator may type in this message if he wants to release the operation on the device which resulted in the error message (see Error Handling).

<device address> gives the address of the device which the operator

wants to release and consists of two hexadecimal characters, as specified at system generation time.

After release of the device, control is returned to the user with the error status in word 4 of the Event Control Block.

example: RD 02

Release the last I/O operation on the device with physical address 02.

Release Programs (DRTM only)

syntax: RP<number>[,D]

use: This message causes all programs loaded in the partition specified to be released.

<number> is a number from 1 to 7 indicating the partition number.

D if specified, indicates that the preceding partition is to be extended, i.e. partition <number> is added onto partition <number> - 1.

If a program, connected to a certain level, belongs to the partition released, it is automatically disconnected from that level. This message must not be used while a program is running in that partition.

example: RP 3,D

This message specifies that all programs in partition 3 are to be released; the partition is destroyed and its memory space is added onto partition 2. The partition numbering stays the same, i.e. the remaining partitions are 1, 2 and 4 (if there were 4 partitions originally).

Restart a Program (DOS and DRTM)

syntax: RSL<name>,<value>

use: This message causes a program, which has been stopped temporarily by a Pause Monitor Request or PS operator message, to be restarted. <name> is the name of the program which is to be restarted and need

only be specified under the DRTM for foreground programs.

<value> is a hexadecimal value of up to four hexadecimal characters which will be loaded into the A7 register by the monitor. The operator may specify this value as an answer. This new value for the A7 register may be specified only if the program has been stopped by the Monitor Request Pause.

For DRTM only: If this message is given in relation to a disc-resident program which had been swapped out of memory following a previous PS message, it will be reloaded into the dynamic partition when its priority is the highest of all disc-resident programs waiting for activation.

example: RS PROG,81E

The program PROG which has previously been stopped by a Pause Monitor Request, is restarted and will find the value X'081E' in the A7 register.

Retry an I/O Operation (DOS and DRTM)

syntax: RY<device address>

use: When the monitor has typed out a PU error message (peripheral unit failure) after an I/O operation and thus requests operator intervention, the operator may type this message to retry that same I/O operation, after he has taken any necessary steps (see Error Handling).

<device address> gives the address of the device on which the I/O operation has to be retried and consists of two hexadecimal characters, as defined at system generation time.

If the operation succeeds now, control is returned to the user with the status in word 4 of the Event Control Block. If it still does not succeed, a new error message may be output by the monitor if a retry is still possible, or the operator may release the device (RD operator message).

example: RY 02

Retry the last I/O operation on the device with physical address 02.

k

Save a Core Image Program onto Disc (DRTM only)

syntax: SAU<name>,<file code>[S]

use: By means of this command the program specified (a core-image program) can be written onto an integer number of sectors of the disc of which the file code is specified. For each program saved, a new entry is made in the General Catalogue of the related disc.

<name> is the name of the program which is to be saved.

<file code> is the file code of the disc onto which this program is to be saved.

S if specified, declares this program to be swappable. In that case, the monitor reserves two areas on the disc: the first one to contain the saved core image program, the second one as a swap area, in case this program is swapped out of memory during execution.

Set Clock (DRTM only)

syntax: SC<HH,MM,SS>

use: This message is given to initialize the clock.

HH indicates the hour (a value from 0 to 23).

MM indicates the minute (a value from 0 to 59).

SS indicates the second (a value from 0 to 59).

The value typed in by the operator will be automatically incremented by the monitor through real time clock interrupts.

When it reaches 60 hours, 00 minutes, 00 seconds, a message will be sent to the operator requesting to update the date.

(if the operator message SD has been given previously; see next message).

example: SC 3,45,2

The clock is initialized with the time 3 hours, 45 minutes, 2 seconds.

Set Date (DRTM only)

syntax: SD, DD, MM, YY

use: This message is used to specify the date. It will not be updated by the monitor (see SC operator message).

DD indicates the day (a value from 1 to 31).

MM indicates the month (a value from 1 to 12).

YY indicates the year (a numeric value of 2 characters).

example: SD 21,6,71

The date is specified as June 21, 1971.

System Status (DRTM only)

syntax: SS[L<software level>]

use: If the operator wants status information of a program running at a certain level or of all levels, he can give this message. The following information is printed out on the typewriter:

- for a specified software level:

<program name> L <start address> L <save area address> L <program status word> L <ECB address for 'Wait'> L <ECB address for 'Activate'>

- if no level has been specified, status information is given for all software levels:

<level 50 status word> / <level 51 status word> / /
<level 63 status word>

Except for the program name, all this information is printed in hexadecimal format.

Start a Program (DOS and DRTM)

syntax: ST[L<name>]

use: This message activates the program specified. If no name is specified, this is assumed to be the background program.

This message is valid only if the program has previously been connected to a software level.

The activated program will be handled by the dispatcher and start with its own priority.

For disc-resident foreground programs, this message may result in a swapping operation, if such programs have been declared swappable.

example: ST PROG

The program with the name PROG is started.

Write into Memory (DOS and DRTM)

syntax: WM <address>, <value1> [<value2> <valuen>]

use: This message can be used to correct the contents of one or more memory locations.

<address> is the first location which is to be corrected (a hexadecimal address of up to 4 characters).

Values 1 to n are values which are to be entered into the memory locations starting from <address>, i.e.

<value1> is put in location <address>

<value2> is put in location <address> +2, etc.

example: WM 4FE, 44F 3FE4

The value 44F is placed in memory location 4FE.

The value 3FE4 is placed in memory location 500.

m

MONITOR REQUESTS

The user program can make requests to the monitor for specific functions. This is done by first giving an LKM (Link to Monitor) instruction and then a DATA directive with a number as operand. This number indicates specifically the function requested from the monitor.

Preceding these requests, certain parameters may need to be loaded into the A7 and/or A8 registers.

Following each request, the system gives the result in the A7 register:

- if A7 = -1, the corresponding processor is not in memory;
- if A7 = -2, the requested function cannot be processed.

After a request has been processed, control is generally returned to the program which issued it.

In the following paragraphs the function and calling sequence of each request is described. The requests are listed in numerical order, according to their DATA number. Some of the requests can be used with the DOS monitor as well as with the DRTM. Others can be used with the DRTM only. This is indicated in the heading of each request.

REQUEST	A7 REGISTER	A8 REGISTER	LKM DATA	USABLE IN		PAGE
				DOS	DRFM	
I/O Requests	I/O order	ECB address	1	X	X	
Wait	---	ECB address	2	X	X	
Exit	---	---	3	X	X	
Get Buffer	Buffer length	---	4	X	X	
Release Buffer	Buffer length	Buffer address	5	X	X	
Pause	Mess. block addr.	---	6	X	X	
Control Abort	Param. block addr.	Routine address	7	X	X	
Connect to Level	Level	Progr. name addr.	8		X	
Disconnect Level	Level	Progr. name addr.	9		X	
Connect to Timer	Progr. name addr.	Parameters	10		X	
Disconnect Timer	Progr. name addr.	Timer number	11		X	
Activate	Progr. name addr.	Param. block addr.	12		X	
Switch	Level	---	13		X	
Attach Device	Device block addr.	---	14		X	
Detach Device	Device block addr.	---	15		X	
Get Time	Time block address	---	17		X	
<i>New</i> <u>Delete Disc File</u>	-	ECB address	18		X	
<u>Assign File Code</u>			19	X		
<u>Load/Link (DOS)</u>			20	X		

TABLE OF MONITOR REQUESTS

Input/Output Requests (DOS and DPM)

calling
sequence:

LDX A7, CODE
LDKL A8, ECBADR
LEI
DADA 1

where CODE must be specified as follows:

	W	R	ORDER		
0	7	8	9	10	15

and ECBADR is the address of the Event Control Block, which contains the parameters for the requested I/O function.

use:

Through this request the user can ask the system to start a certain I/O operation on a peripheral device. The parameters loaded in the A7 register have the following meaning: bits 8 and 9 specify the mode of operation:

W = 1: the requesting program wants to wait for the completion of the requested I/O operation. Only after completion of the requested function, will the return to the calling program take place.

W = 0: a return to the calling program will be made as soon as the transfer has been initiated. The program will give a Wait request later on for synchronization.

R = 1: the program itself will process any abnormal condition concerning the requested operation. The system will return the hardware status in ECB word 4.

R = 0: any abnormal conditions will be processed by the system. The software status is returned in ECB word 4.

ORDER specifies which I/O function is required, by giving one of the following combinations of 2 hexadecimal digits:

01: Basic Read

05: Basic Write.

For Basic (Binary) I/O requests the system does not provide for character checking or data conversion, only for control command initialization and end of operation signals.

02: Standard Read

06: Standard Write.

Standard (ASCII) I/O requests provide, by means of standard conversions, for special features such as error control characters, conversion from external code to internal ASCII and vice versa. There is a checksum and characters are stored seven by seven bits, two characters to a word.

07: Object Write (4x4x4x4 tape format)

08: Object Write (8x8 tape format).

Object I/O requests provide, by means of standard conversions, for special features such as error control characters, checksum and data conversion from external 4x4x4x4 or 8x8 tape format to internal 16-bit format.

0A: Random Read

0B: Random Write.

14: Skip forward to EOS mark.

16: Skip forward to EOF mark.

22: Write EOF mark

26: Write EOS mark.

30: Get information about a file code.

37: Skip backward to EOS mark

38: Skip backward to EOF mark.

For each of these request orders, specific information applies to the various peripheral devices. This information is given in Appendix A at the end of the book.

The Event Control Block, of which the address must have been loaded into the A8 register, has the following format:

0	7	8	15	
EVENT CHARACTER		FILE CODE		WORD 0
BUFFER ADDRESS				WORD 1
REQUIRED LENGTH				WORD 2
EFFECTIVE LENGTH				WORD 3
STATUS WORD				WORD 4
TABULATION TABLE ADDR. OR RELATIVE SECT. NBR.				WORD 5

where:

WORD 0: event character: the monitor stores all the information in the bits of this character as follows:
bit 0 = 1: end of operation has occurred for this ECB.
bit 1: unused.
bits 2 and 3: not significant for I/O.
bit 4 = 1: file code has not been assigned.
bit 5 = 1: abnormal end of operation.
bit 6 = 1: end-of-segment has been read.
bit 7 = 1: end-of-file has been read.

WORD 1: address of the user buffer.

WORD 2: requested length to be read or written in words (card reader) or characters (other devices).

The first character is always the character given by the buffer address.

WORD 3: effective length which has been transmitted in words (card reader) or characters (other devices). This information is stored here by the monitor upon completion of the I/O operation.

WORD 4: status word, stored here by the monitor upon completion of the requested I/O operation.

- For Basic orders, this word will be filled with the hardware status by the control unit. However, if the monitor detects an error in the calling sequence, bit 0 will be set to 1 and the other bits will contain the software status.

- For the other orders, the software status will be returned:

bit 0 = 0: the operation has been successfully completed:

bit 10 = 1: beginning of tape encountered.

bit 11 = 1: end of input medium.

bit 12 = 1: requested length is incorrect.

bit 13 = 1: illegal character code.

bit 14 = 1: an EOS mark has been read.

bit 15 = 1: an EOF mark has been read.

When the operation was not successfully completed, bit 0 is set to 1 and bit 1 is set to 0 (retry also was not possible). In this case bits 2 to 15 give the hardware status.

When the monitor has detected an error in the calling sequence, bits 0 and 1 are both set to 1 and bits 11 to 15 have the following significance:

bit 11 = 1: function is unknown or not compatible with the device.

bit 12 = 1: buffer size or address is illegal.

bit 13 = 1: ECB address is illegal.

bit 14 = 1: device is attached to another program.

bit 15 = 1: an illegal file code has been used.

dd

WORD 5: this word is used by the user to store:

- the relative sector number to be exchanged in the case of a random access disc file.
- the tabulation table address in the case of a standard read operation on ASR or punched tape equipment.

This tabulation table has the following format:

Number of Tackets	First Tacket
Second Tacket	Third Tacket

etc.

The tackets indicate an absolute position in the print line. Characters up to the following tacket are filled with blanks.

Example:

3	10
20	30

Input line: LABEL\OPER\OPERAND\COMMENT

Line in buffer:

LABELL...OPER...OPERAND...COMMENT

1 10 20 30

At completion of the input, the buffer is filled with spaces, but the returned length is the length effectively entered and stored, including the spaces replacing the tabulation codes (\).

ll

Wait for an Event (DOS and DRIM).

calling
sequence:

LBL A8, ECBADR
LNH
DATA 2

where ECBADR gives the address of the Event Control Block (see I/O requests). The first character of the ECB is the event character. If the first bit of this character is set to 1, the event has been completed.

use:

This request causes a program to stop and wait for the completion of an event which has to take place in another program (user or system). If the event has occurred, the dispatcher returns control to the requesting program. If the event has not occurred, the program is put in wait state, to be restarted when the event has occurred.

There are two kinds of events:

-- Wait for the exit of a program:

When a user program activates another program (see Activate), the first word pointed to by the A8 register is the address of the ECB which must be used to wait for the exit of the activated program. As the activated program and the calling program run concurrently, this provides a means of communication between the two programs.

-- Wait for an I/O operation:

The program waits for the end of an I/O operation it has requested.

Notes:

- A scheduled label routine may not contain a Wait request.
- For DRIM: if a program running in the dynamic memory partition activates and waits for the exit of another program, it may be swapped out during the waiting period to accommodate another disc-resident program.

Exit (DOS and DPM)

calling

IKM

sequence:

DATA 3

use:

This request is used to specify the end of a program. The program exit is effected after completion of all I/O operations and after all labels, if any, have been scheduled. The program is put in inactive state if no other program has entered an Activate request for this one. It does remain connected to its priority level, however. If this is a level which can be shared, a switch is implicit in the exit (DPM only: see Time Slicing and Switch inside a Level request).

Get Buffer (DOS and DRTM)

calling

LDK A7, LENGTH

sequence:

LDM

DATA 4

where LENGTH is a constant specifying the requested buffer length in characters.

use:

This request asks the system to reserve memory space in the buffer pool defined as partition P1.

The system answers with the following responses in registers:

A14: the address of the second word of the reserved buffer.

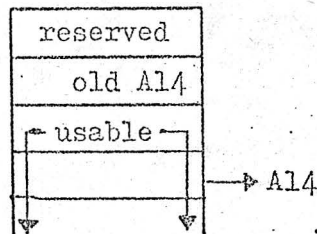
The user contents of A14 is saved by the system and placed in the buffer. This is restored when the user releases the buffer.

A7 = 0: the memory space is reserved for the user in the buffer partition.

= 1: there is not enough memory space for the user in the buffer partition.

= 2: the buffer pool is destroyed and a message is sent to the operator.

The structure of the reserved memory space is as follows:



Notes:

- If 0 is loaded into the A7 register when the request is issued, the monitor will return the address of the upper limit of the buffer partition in the A7 register.
- Programs declared swappable during a SAVE operation must use this monitor request for space allocation for buffers and ECBs.

hh

Release Buffer (DOS and DEPT1)

calling
sequence:

LDX A7, LENGTH
LDXL A14, BUFADR
LEH
DATA 5

where LENGTH is a constant, specifying the length of the memory space to be released, in characters.

BUFADR points to the second available word in the buffer, as given in response to the Get Buffer request (A14).

use:

This request is used to release the memory space reserved previously in the buffer partition by means of a Get Buffer request. The A14 register is restored with the value it contained before the Get Buffer request was made.

The system response is as follows:

A7 = 0: the memory space is released.

A7 ≠ 0: it is impossible to release this memory area because:

- the address is not correct, or
- the length specified is not correct.

Pause (DOS and DRPM)

calling

sequence:

LDK A7,MESLGT
LDK A8,MESBLK
LKM
DATA 6

where MESLGT is a constant, specifying the length of a message which the program may output on being put in Pause state.

MESBLK is the address of a message block containing the message to be output.

use:

This request causes a temporary halt of the running program. It is put in wait state and, if specified, a message is printed out on the operator's typewriter. The program can be restarted only by an operator control message RS.

When the program is restarted, it may be given an additional parameter in register A7 (see RS operator control message).

Note: Under DRPM, when a swappable disc-resident program makes a Pause request, it may be temporarily swapped out.

Keep Control on Abort Condition (DOS and DRIT)

calling

LDEL A7,PARAM
LDEL A6,LABADR
LKM
DATA 7

sequence:

where LABADR is the address of a user label to be scheduled on abort.

PARAM is the address of a 3-word block which will receive the parameters of the abort condition. It has the following format :

Abort Code	0
Aborted PSW	1
Aborted Address (AO)	2

where:

- the abort code has the same meaning as in the regular AB abort message.
- word 1 gives the PSW of the aborted program.
- word 2 gives the address where the abort took place (register AO contents).

use:

By means of this request the user can himself handle abort conditions, i.e. his own routine replaces the abort handling of the monitor. Thus, the program will not be declared aborted by the monitor, but control will be transferred normally to the user routine attached to this request (register A8). The parameters of the abort condition will be placed in a 3-word block and, according to the returned parameters, the user routine can take action. This request is to be given once in a program, at the point from where the user wants control over any abort conditions, so mostly at the beginning of the user program. It is usable only once, so if an abort takes place and the program is restarted, the request has to be given again. When a user decides that, at a certain point in his program, he wants to return to regular abort handling by the monitor, he must again specify this request but load into A7 the address of the EXIT routine.

Intended as a debugging facility

Connect a Program to a Software Level (LRTM only)

calling

LDK A7,NUMBER

sequence:

LDKL AS,PRNAME

LKN

DATA 8

where NUMBER is the number of the software level to which the program is to be connected.

PRNAME gives the address of a 3-word block containing the name of the program.

use:

A program, which may be running at any level, can make a request for another program to be connected to a software level by means of this monitor request.

The system responds as follows:

A7 = 0: the connection has been accomplished.

A7 ≠ 0: it is impossible to make the connection, because

- the program does not exist;
- the program is not compatible with the requested level, i.e. it is an interrupt level;
- the program has already been connected to a level.

ll

Disconnect a Program from a Software Level (DEEM only)

calling

LDK A7, NUMBER
LDKL A8, PRNAME
LDK
DATA 9

sequence:

where NUMBER is the number of the software level to which the program was connected.

PRNAME gives the address of a 3-word block containing the name of the program.

use:

By means of this request, the program PRNAME is disconnected from the software level indicated in register A7. If this program was connected to a timer, it is disconnected from that as well.

The system responds as follows:

A7 = 0: the program has been disconnected.

A7 \neq 0: it is impossible to disconnect the program, because

- the program was not connected to this level (result: a NO-operation);
- the program is busy.

MM MM

Connect a Program to a Timer (DFTM only)

calling
sequence:

LDR A7, PRNAME
LDEL A8, PARAM
LDM
DATA 10

where PARAM specifies the necessary parameters in one word as follows:

NC	NTIM	PR
0	7 8 9 10	15

in which NC is the number of cycles of the timer before the first activation (a number from 0 to 255).

NTIM is the number of the timer (0 is assumed to be the real time clock).

PR is the requested pulse rate, i.e. the number of cycles of the timer between two activations (a number from 0 to 63).

PRNAME points to a 3-word block containing the name of the program.

use:

By means of this request, a program running at a software level can be connected to a timer, according to the parameters given in register A7.

At generation time it is possible to define three different software timers, whose pulse rates are related to the cycle time of the real time clock.

The program must have been connected to a software level, otherwise it can not be started by the dispatcher.

The system responds as follows:

A7 = 0: the connection has been accomplished.

A7 ≠ 0: connection is impossible because:

- the timer specified has not been defined;
- the program has already been connected to a timer.

M.M.

Disconnect a Program from a Timer (DRTM only)

calling

LDX A7, PRNAME
LDKL A8, NUMTIM
LKM
DATA 11

sequence:

where NUMTIM is a constant specifying the number of the timer. PRNAME points to a 3-word block containing the name of the program.

use:

By means of this request a program can be disconnected from a timer specified in register A7. If the program had also been connected to a software level, that connection remains.

The system responds as follows:

A7 = 0: the program has been disconnected.

A7 ≠ 0: it is impossible to disconnect the program because:

- the timer specified does not exist;
- the program specified does not exist.

Activate a Program (DRTM only)

calling
sequence:

LDKL A7,IRNAME
LDKL A6,BLOCK
LKH
DATA 12

where IRNAME points to a 3-word block containing the name of the program which is to be activated.

BLOCK points to a 2-word block of the following format:

reserved for the system	(ECB)
parameter block address	

The first word of this block is updated by the system at the exit of the activated program. If the calling program has requested a wait for the activated program, this word must be considered as its Event Control Block(ECB); see the Wait request. The second word contains the address of a parameter block, which may serve as an interface between the activating and the activated program. If the activated program is in the same partition as the activating program, the activated program can write in this block without problems, because there is no memory protection. If, however, the activated program is not in the same partition, the calling program must, before activating this one, send a Get Buffer request to build its parameter block in the buffer partition. In both these cases register A14 will contain the address of the parameter block when the activated program is started.

use:

This request can be made by a program running at any level, to activate a software level program. Depending on their priority levels, both the calling and the activated program may be processed concurrently.

If the activated program is busy, the request is recorded in a stack and will be processed in due time (see also DRTM: Program Control Table).

If the activated program must run in the dynamic partition, the loader can be activated to load the program.

Note: This may result in a swapping operation if one disc-resident program activates and waits for the exit of another one.

The system responds as follows:

A7 = 0: the request has been processed.

A7 ≠ 0: it is impossible to process this request, because the program has not been connected to a level.

PP

Switch inside a Software Level (DRTM only)

calling
sequence:

LDX A7, LEVEL
LKM
DATA 13

where LEVEL is a constant specifying the number of the level in which the switching must be done.

If LEVEL is specified as 0, the level to be switched is equal to the level of the requesting program plus one, i.e. if this request is made in a program on level 52 with 0 in the A7 register, the switching is to take place on level 53.

use:

By means of this request it is possible to have time-slicing inside a software level by halting execution of the program running on that level and giving control to the next program on the same level, according to timer interrupts (see also DRTM: Time Slicing).

Attach a Device to a Program (DRTM only)

calling
sequence:

LDKL A7,DEVBLK
LKM
DATA 14

where DEVBLK points to a 2-word block of the following format:

DN	
	XX

in which DN is the device name (2 ASCII characters).

XX is the physical device address (2 hexadecimal characters).

use:

By means of this request a program running at any software level can reserve the use of the device specified to itself; in that case no other program can perform any I/O operations on this device.

Note: therefore great care must be taken using this request, for system response time will deteriorate when frequently used devices are attached to low-priority disc-resident programs (see next monitor request).

The system responds as follows:

A7 = 0: the device has been attached to this program.

If the device does not exist, or if it is impossible to attach it, the program is aborted.

If the device has already been attached to another program, the requesting program is put in wait state until the device specified has been detached (see following request).

Detach a Device from a Program (DRTM only)

calling
sequence:

LDKL A7,DEVBLK
LKM
DATA 15

where DEVBLK points to a 2-word block of the following format:

DN	
	XX

in which DN is the device name (2 ASCII characters).

XX is the physical device address (2 hexadecimal characters).

use:

By means of this request a device, which has been previously attached to a program through an Attach request, is detached from that program.

To release the device as quickly as possible, this request must be made as soon as the program concerned no longer requires the use of this device.

The system responds as follows:

A7 = 0: the device has been detached.

A7 ≠ 0: it is impossible to detach the device because:

- it is busy with an I/O operation for this program;
- it is attached to a different program;
- the device specified does not exist.

Get Time (DRTM only)

calling

LDXL A7, TIMBLK
LJH
DATA 17

sequence:

where TIMBLK points to a 6-word block which will be returned by the monitor upon receiving this request. Its format is as follows:

DAY
MONTH
YEAR
HOUR
MINUTE
SECOND

The values are given in ASCII.

use:

Upon receipt of this request the monitor will print out the date and time.

XX

Basic Read (X'01')Operator's Typewriter:

All characters are entered on 7 bits until the requested length is reached.

ASR Tape Reader:

All characters are entered on 7 bits. The reader stops one character after an Xoff code has been read.

High-speed Tape Reader:

All characters are entered on 8 bits, without checking or special features, until the requested length is reached.

Card Reader:

All the words are entered and stored in Hollerith code on 12 bits (4 to 15). In each word the column image is right-justified. The words are stored until the requested length is reached. The length is given in words.

Basic Write (X'05')

Operator's Typewriter:

All characters are output without checking or special features. This order can be used to print something and have the answer on the same line.

ASR Tape Punch:

All characters are output without checking or special features. The eighth bit is always punched.

Line Printer:

All characters are output without checking. There is no control character.

High-speed Tape Punch:

All characters are output without checking or special features.

Standard Read (X'02')

Operator's Typewriter:

ASCII characters are entered on 7 bits, with the following special features:

- the special characters, coded from X'0' to X'1F', are ignored.
- code X'7F' (Rub-out or Delete character) is ignored.
- code X'5F' (←) can be used to delete the preceding character. If several ← are used consecutively, an equal number of preceding characters will be deleted.
- code X'5B' (↑) is used to delete the line preceding it, up to the next carriage return.
- code X'0D' (carriage return) indicates end of block. It is the last character to be entered. It is not transmitted to the user's buffer.
- code X'0A' means 'line feed'.
- code X'5C' (\\) is used as a tabulation symbol (see ECB word 5). If the address of the tabulation table is zero, or if the number of tackets is zero, or if the storage address is greater than the last tacket, the code X'20' is stored in the buffer. In other cases, X'5C' is not stored and replaced by spaces, as indicated by the tackets in the tabulation table.

ASR Tape Reader:

For ASCII characters, the same features apply as for the keyboard: the code for carriage return must be preceded by the code for Xoff.

For object code in 4x4x4x4 tape format, the first character identifies the object format. It must be in the range from X'18' to X'1F' and is converted to a number from X'0' to X'7' and stored on one character. The second character contains the word-count of the input block, excluding the first word and the checksum. Each punched row (4 bits) entered after this identifier is stored on one half-character up to the checksum. When the checksum has been read, input is stopped. The 8x8 tape format cannot be read on the ASR tape reader. To start the reader, an Xon code is sent by the system before entering the characters.

High-speed Tape Reader:

Same as for the ASR tape reader. In addition: for object code in 8x8 format, the first character, identifying the object code format, must have one of the following values: X'10', X'1' to X'4' or X'15' to X'17'. It is converted to a number from X'0' to X'7'. Each punched row (8 bits) entered

after this identifier is stored on one character up to the checksum. The second character is the length of the block, in words, excluding the first word and the checksum.

Card Reader:

All words are read in Hollerith code, on 12 bits, converted and stored in ASCII code, on 7 bits, until the requested length is reached. Words which are not in Hollerith are converted into the ASCII code for X'20' and a 'data fault' status is returned in the software status (ECB word 4: bit 13 is 1). There is no special code. However, EOS and EOF marks are detected (bits 14 and 15 in the software status).

Standard Write (X'06')

Operator's Typewriter:

All characters, except X'0' to X'LF' (special code characters) are output without checking. At the end of a line, a carriage return and line feed are output.

ASRTape Punch:

Same features as for the keyboard. At the end of a line, the following character sequence is output: LF -- Xoff -- CR -- Rub-out

High-speed Tape Punch:

Same as for ASR tape punch with the following exception: the first character is a control code: if it equals X'30' or X'31', it is output as Line Feed; if it is different, it is not output.

Line Printer:

All characters are output without checking, except for the control code. It must be stored in the word preceding the buffer address. This control code may have one of the following three values:

- + (X'2B'): print the line without advancing the paper (superposition).
- 0 (X'30'): advance two lines before printing.
- 1 (X'31'): skip to top of page before printing.

All other control codes are used as normally: advance one line and print. At the end of the buffer, after the requested length, one word must follow to be used by the system for a print code.

If the requested length is more than one line, the system puts a print code after the maximum length and the buffer will be printed on two or more lines.

Object Write 4x4x4x4 Tape Format (X'07')

ASR Tape Punch:

The first character is output on one row, converted from X'0' - X'7' to X'18' - X'1F'. Each following character is output on two rows; to avoid special (ASCII) code each row is converted. The second character contains the length of the block in characters, excluding the first character. At the end an 8-bit checksum is performed and punched, followed by an Xoff code.

High-speed Tape Punch:

Same as for ASR tape punch, except that the second character contains the length in words.

Object Write 8x8 Tape Format (X'03')

High-speed Tape Punch:

The standard object code is output in 8x8 format, where the first character is a format character and is output on one row, converted as follows :

X'0' → X'10'

X'1' to X'4' → X'01' to X'04'

X'5' to X'7' → X'15' to X'17'.

The second character contains the length in words, excluding the first word. An 8-bit checksum is performed and punched.

Write EOF mark (X'22')

Operator's Typewriter:

An end-of-file mark is output as follows: :EOF LF Xoff CR Rub-out.

ASR Tape Punch:

An end-of-file mark is output as follows: :EOF LF Xoff CR Rub-out

High-speed Tape Punch:

An end-of-file mark is output as follows: :EOF LF Xoff CR Rub-out

Line Printer:

An end-of-file mark is output as follows: :EOF

Write EOS mark (X'26')

Operator's Typewriter:

An end-of-segment mark is output as follows: :EOS LF Xoff CR Rub-out

ASR Tape Punch:

An end-of-segment mark is output as follows: :EOS LF Xoff CR Rub-out

High-speed Tape Punch:

An end-of-segment mark is output as follows: :EOS LF Xoff CR Rub-out

Line Printer:

An end-of-segment mark is output as follows: :EOS

Read up to End-of-Segment (X'14')

High-speed Tape Reader:

The tape is read until an :EOS statement has been read.

Card Reader:

The cards are read until an :EOS statement has been read.

Read up to End-of-File (X'16')

High-speed Tape Reader:

The tape is read until an :EOF statement has been read.

Card Reader:

The cards are read until an :EOF statement has been read.

Return Information about a File Code (X'30')

By means of this order it is possible to find out the assignment of a file code. The information will be returned in the Event Control Block:

ECB - word 0: Event Character + File Code.

ECB - word 1: Device Name (2 ASCII characters):

TY = operator's typewriter (listing)

TR = ASR tape reader

TP = ASR tape punch

PR = tape reader

PP = tape punch

LP = line printer

CR = card reader

CP = card punch

MT = magnetic tape

TC = cassette tape

NO = no device

ECB - word 2: maximum record size.

ECB - word 3: left character: unused.

right character: device address.

ECB - word 4: status = 0.

If the file code in ECB word 0 is set to zero, the other words of the ECB will also contain zeros.