

GENERAL

A program in any machine, be it system, foreground or background, may issue a request for monitor services by means of the LKM instruction, but the validity of these requests and their subsequent execution vary according to the type of machine issuing the request. In addition, within any one machine where a particular LKM is permitted, there sometimes exist variants of these LKM instructions which modify their execution.

The LKM requests are described in the order of the DATA n which follows the LKM instruction. If there is no indication of the machine to which the LKM request applies, it applies to both foreground and background machines.

LKM 1 - I/O Requests

Purpose

To initiate an action on, or retrieve information about, a device or file.

Applies to: Foreground, Background.

Calling Sequence

```
LDK   A7,L      (or LDKL if bits in 1st character are set)
LDKL  A8,M
LKM
DATA  [-]1
[DATA N]
```

Where L = Request Order Code (see below)
M = Event Control Block (ECB) Address
N = Scheduled Label Address.

Request Order Codes:

- bit 0 = 1 Non capital letters (a-z) converted to capital ones (AMA8 and ASCU4Z).
- bit 1 = 1 An address of an SCT (special character table) is recorded in word 5 (ECB+10) of the ECB (AMA8 and ASCU4Z).
- bit 6 = 1 Timeout period is specified in location ECBHD (word 6) of the ECB (for teletypes etc.)
- bit 6 = 0 Default value of timeout defined at Sysgen is to be used.
- bit 8 = 1 Implicit Wait: The requesting program will be put into a wait state until the operation is terminated.
- bit 8 = 0 No Implicit Wait: Control is returned to the calling program as soon as the request is recorded.
- bit 9 = 1 User Error Action: The requesting program will process all abnormal or error conditions. The hardware status is returned in this case.
- bit 9 = 0 System Error Action: The system performs the standard error action and returns an error status to the calling program.

Bits 10-15 of A7 define the function required:

- /00 Get device/file description
- /01 Basic Read
- /02 Standard Read
- /05 Basic Write
- /06 Standard Write
- /0A Direct Read (Disc File)
- /0B Direct Write (Disc File)
- /11 Direct Read (DAD or Disc Unit)
- /15 Direct Write (DAD or Disc Unit)
- /10 Replace a bad track
- /12 Replace a bad track (CDC disc)
- /13 Seek to track zero (CDC disc)
- /18 Write home address and premark the track (CDC disc)

- /30 Get Information about a Filecode
- /22 Write EOF Mark
- /26 Write EOS Mark

For Magnetic Tape and Cassette the following extra codes are available:

- /16 Skip Forward to EOF Mark
- /24 Write EOV
- /31 Rewind (also DFM file)
- /32 Fast Search forward to tape mark (cassette only)
- /33 Skip 1 block backwards
- /34 Skip 1 block forwards (MT only)
- /35 Fast search backward to tape mark (cassette only)
- /36 Skip backward to EOF mark
- /37 Lock (cassette only)
- /38 Unlock.

For Flexible Disc the following request order codes are available:

- /21 Read VTOC
- /25 Write VTOC
- /2D Door Lock
- /2E Door Unlock
- /2F Write Deleted Data Address Mark
- /3A Compound Read
- /3B Compound Write
- /3C Search Key with Mask
- /3D Write Deleted Data Address Mark and Verify
- /3E Search Key
- /3F Write Sector and Verify.

For all these functions MAS is compatible with present systems, the DFM/TDFM functions remaining unchanged. DADs can be accessed via filecodes /FO - /FF, they can be read by any user program in the batch machine, but written only by the BCP, EDF and Librarian processors. Foreground users can access their DADs and discs directly.

The physical disc units (filecodes /CO - /CF) are only accessible to foregroundprograms and the Librarian processor. The orders /11 and /15 are used to read and write a sector of the disc. Great care should be taken to use direct disc access.

The ECB Layout

An ECB is a 5, 6 or 7 word table containing required parameters, but for those functions not requiring words 6 or 7, these words may be omitted. For get information order codes (/00 and /30), the ECB is considerably longer.

bits	0 1	7 8	15
ECB 0	E L		Filecode
ECB 2	Start Address of Record Area		
ECB 4	Requested Record Length		
ECB 6	Effective Length		
ECB 8	Returned Status		
ECB 10	See below		
ECB 12	See below		

ECB -2 may be used to contain the address of an ECB to whose event this I/O operation is linked (multiple wait; see LKM 2). On completion of this I/O operation, the event bits of all chained ECB's will be set.

For Flexible Disc the layout is as follows:

bits	0 1	7 8	15
ECB 0	E L	Filecode	
ECB 2	Buffer Address (containing delete pattern if order /3D is used)		
ECB 4	Requested Length		
ECB 6	Returned Effective Length		
ECB 8	Returned Status		
ECB 10	No. of first sector to be read/written		
ECB 12	zero	Timeout value	

Explanation of Tables

ECBFC (byte 0): E and L (bits 0 & 1) refer to event handling (See 'Wait for Event' request, LKM 2).
 (byte 1): Bits 8-15 contain the filecode.

ECBBF (bytes 2 & 3) : Start Address of the record buffer area.

Note: For flexible disc the following apply:

- For request code /3D (delete) the buffer should contain the delete pattern.
- for request codes /3C and /3E the buffer should contain the search key block.

ECBRL (bytes 4&5): The length in characters of the record to be written, or the maximum length of record to be read. If an odd number of characters is read, MAS will round up to an even number by adding 1.

ECBEL (bytes 6&7): The actual length in characters which has been transferred. This is returned by the Monitor upon completion of the operation.

ECBST (bytes 8&9): The status returned by the Monitor upon completion of the operation. These are described below under 'Returned Status'.

ECBSC (bytes 10&11):

- a) For Direct Access on Disc Files (DFM) or DAD devices, it specifies the relative sector number, from the beginning of the file or device.
- b) For 'Disc Physical Unit' (filecodes /CO - /CF), it specifies the cylinder number.
- c) For flexible disc, this is the absolute sector number except in the case of search orders /3C and /3E, for which the user leaves it blank and MAS returns the actual sector number found in a successful search operation.
- d) For AMA8 and ASCU4Z devices, this is the address of the SCT (special character table, when bit 1 in the order is set.

ECBHD (bytes 12&13):

- a) For 'Disc Physical Unit' (filecodes /CO - /CF), it specifies the Head Number (byte 12) and Physical Sector Number (byte 13) for the transfer.
- b) For such devices as displays, teletypes and floppy disc, the timeout period is specified in byte 13, byte 12 being set to zero, when bit 6 in the order is set.

Returned Status (ECBST)

- a) Zero: The operation terminated satisfactorily.
- b) Positive: The operation was completed, but the following conditions were encountered:
- 1 EOF encountered (Read).
 - 2 EOS encountered (Read).
 - 4 Data Error.
 - 8 Incorrect Length.
 - /10 End of tape, end of media, request done.
 - /20 Beginning of tape.
 - /40 End of tape reached, but the current record has been read or written (warning signal).
 - /80 EOY mark detected.
 - /100 No data on MT or cassette.

c) Negative (bit 0 set):

Bit 1 = 0: Bits 2-15 indicate the hardware status.

Bit 1 = 1:

- /C001 Filecode illegal or not assigned.
- /C002 Device attached to another program.
- /C008 Buffer address or requested length invalid.
- /C010 Function unknown or incompatible with the device or file, or invalid sector number for Floppy Disc.
- /C020 Write protection on Disc File.
- /C040 End of medium: current operation aborted.
- /C080 Timeout.
- /C100 Disc Queue overflow.
- /C200 Dynamic Buffer overflow; no disc blocking buffer free.
- /C400 Blocking overflow (No free granule).

Notes: Status code /C010 is returned when trying to write to cassette with a length greater than 256, whereas status code 0 is returned when trying to write to magnetic tape with a length greater than 4096.

Order code /30

The request order /30 (Return Information About a Filecode) is processed by a disc-resident program; the user only has to specify the filecode in ECBO. The information is returned as follows:

ECBBF	Device Name:
DY	display
TK	cassette
MT	magnetic tape
LP	lineprinter
CR	cardreader
DK	disc
PL	plotter
PR	papertape reader
PP	papertapepunch
FL	0.25m data floppy
DD	DAD device
DL	logical disc file
EF	extended (TDFM) file
NO	file code assigned to NO device
TY	typewriter

ECBRL Maximum Record Size (in characters).

ECBEL

a) For non-disc devices:

Bits

0-9 These may be set but are not significant to the user. A description of these (device dependent) bits may be obtained from the Trouble Shooting Guide, in the DWT chapter.

10-15 Contain the device address.

b) For Disc Files (DFM):

Bits 8-15 contain the DAD filecode (/FO - /FF).

c) For DAD devices:

Bits 8-15 contain the disc filecode (/CO - /CF) on which the DAD is located.

ECBST Status of the operation:

Zero = OK

/C001 = Illegal Filecode.

/C008 = Invalid Buffer Address and/or Requested Length.

ECBSC:

a) For Disc File (DFM):

Bit No.	Meaning if set
0	-
1	Write Protected
2	Source
3	Object
4	Undefined File (User File)
5	Load Module
6	Temporary Disc File (Catalogued file if <u>not</u> set)
7	Direct Access
8	Sequential Access
9	EOF mark written
10	-
11	-
12	Consecutive File
13	-
14	-
15	-

b) For DAD:

Bits 0-7 No. of Interlaces

Bits 8-15 No. of sectors/granule.

c) For AMA8 and ASCU4Z

If bit 0 = 1: linenumber in bits 1-15 (0-7 for AMA8, 0-3 for ASCU4Z).

For extended files and filecode assigned to NO device only the device name is filled in and the rest of the ECB is set to zero.

Order Code /00 - Get Device/File Description

This order has been introduced as an extension to order code /30, since further details concerning the characteristics of a device or file are required for some system processors.

The user enters the event bits (E & L in the table) and the filecode in ECB0 as usual before issuing the request. The required information is returned as follows:

For information common to all devices:

ECB2: bits 11-15 contain the assign type:

- 0 The filecode is assigned to a physical device
- 2 The filecode is assigned to a DFM file
- 4 The filecode is assigned to a DAD device
- 6 The filecode is assigned to a TDFM file.

ECBST: Returned Status.

The contents of the other locations depend on the device type. For non-TDFM devices or files, the ECB consists of 15 words numbered by byte (0, 2, 4, ... 28), which contain the following information:

a) For a physical device:

bits	0	1	7	8	9	10	15
ECB0	E	L					Filecode
2	Reserved			Type = 0			
4	Device Name						
6	Reserved				Device Address		
8	Status						
10	Reserved				Device Type		
12	Best (Max.) Length						
14							
16							
18	Linenumber +/-8000 (AMA8 and ASCU4Z)						

Explanation

Device Name is 2 ASCII characters, as for order /30.

Device Address occupies the least significant 6 bits.

Device Type occupies the least significant 6 bits, thus:

- /00 Typewriter
- /02 Display
- /04 Card Reader
- /08 Paper Tape Reader
- /0C Paper Tape Punch
- /10 Line Printer
- /14 Plotter
- /18 Magnetic Tape
- /1C Cassette Tape
- /20 X1215 Removable Disc
- /21 X1215 Fixed Disc
- /22 CDC Disc (400 cyl., 5 heads)
- /23 CDC Disc (800 cyl., 5 heads)
- /24 CDC Disc (400 cyl., 19 heads)

- /25 CDC Disc (800 cyl., 19 heads)
- /26 X1216 Removable Disc
- /27 X1216 Fixed Disc
- /28 Fixed head disc
- /29 CDC-CMD 16M removable disc
- /2A CDC-CMD 16M fixed disc
- /2B CDC-CMD 48M fixed disc
- /2C CDC-CMD 80M fixed disc
- /2F Floppy disc.

b) For Line Printers:

- ECB14 No. of remaining lines (in binary) on the current page.
- ECB16 No. of lines/page (binary).

c) For Disc Devices:

- ECB14 The VTOC sector size in characters
- ECB16 Absolute sector address of VTOC
- ECB18 No. of tracks/cylinder
- ECB20 Address of the Bad Track List sector
- ECB22 No. of sectors/track
- ECB24 No. of interlaces in the first DAD, containing the VTOC
- ECB26 Pack Volume No. (binary)
- ECB28 Type (only for CDC-CMD) lay out:
 Bit 0 = 0: removable part
 Bit 0 = 1: Fixed part
 Bit 10-15: device type.

d) For a DFM File:

	0 1	7 8	15
ECB0	E L	Filecode	
2	Reserved	Type = 2	
4	'DL'		
6	DAD Filecode		
8	Status		
10	Flags as ECBSC in order /30		
12			
14	GRANTB addr in the DAD		
16	Relative Current Sector Number		
18	Addr of Current Sector in the DAD		
20	Rel. Highest Sector No. in the DAD		
22	Sector Size in characters		
24	No. of sectors per granule		
26			
28			

e) For a DAD device:

	0 1	7 8	11 10	15
ECBO	E L		Filecode	
2	Reserved		Type = 4	
4	Device Name = `DD`			
6	Disc filecode (/Cx)			
8	Status			
10	No. of sectors/track			
12	Sector length in characters			
14	No. of cylinders of the DAD			
16	Addr of the 1st cylinder of the DAD			
18	No. of sectors/granule			
20	No. of tracks/cylinder			
22	No. of interlaces			
24)			
26) DAD Name			
28)			

f) For a TDFM file:

	0 1	7 8	15
ECBO	E L		Filecode
2	Reserved		Type = 6
4	`EF`		
6	DAD Filecode of Descriptor File		
8	Status		
10	Disc pack number (binary)		
12)		
14) DAD name		
16)		
18)		
20	(Userid		
22	(
24)		
26)		
28	(Extended File name		
30)		
32) Maximum number of records (4 char)		
34)		
36) Creation date (4 char)		
38)		
40) Current number of records (4 char)		
42)		
44) Number of deleted records (4 char)		
46)		

Order Codes /10 and /18 - Replace a Bad Track

This function is only used by the Librarian Processor when a bad track is detected during a Premark operation; its purpose is to flag the track as bad and replace it by a good one. It will modify the 'Bad Track Table' kept at the beginning of the disc and in memory, and it will premark the replacement track in the same format as that of the bad one.

Before issuing the LKM, the cylinder number and head number of the track to be replaced are entered into the ECB thus:

ECB0		Disc Filecode (/Cx)
2	unused	
4	Record Length in characters	
6	unused	
8	unused	
10	Cylinder No.	
12	Head No.	No. of records in track

After successful completion of the operation, the system returns the cylinder and head numbers of the replacing track in the same locations, i.e. in ECB10 (cyl. No.) and ECB12 (head No. left byte).

Besides the status code in A7, the following are returned in ECB8:

Zero	Successful termination
/C001	Invalid filecode (not disc)
/C010	Invalid order (requesting program not Librarian Processor)
/C020	Overflow of Bad Track Table, but bad track is flagged
/C021	Too many bad tracks (more than spare tracks)
/C023	Track capacity error (too many records and/or recd. too long)
/C200	Dynamic area overflow - but the bad track is flagged.

Floppy Disc Order Codes

See Appendix E, Peripheral Drivers.

Write Home Address (CDC Disc)

This order uses the same ECB layout as for order /12.

Value of register A7 after LKM 1

When A8 contains a valid ECB address (within the program boundaries) the value of A7 is unchanged.

When A8 is invalid, A7 contains: -1.

LKM 2 : Wait For Event

Purpose

To synchronize tasks within a program or between one or more programs.

Calling Sequence:

```
LDKL A8,ECB
LKM
DATA [-]2
[DATA Scheduled Label]
```

The Event Word pointed to by A8 is modified on termination of the awaited operation. The format is:

0	1	15
E	L	

E=0 Event has not occurred
E=1 Event has occurred. E is always set to 1 after completion of a LKM 2
L=0 Single Event
L=1 Chained event; when the event occurs, it implies that the system must set the other event, the ECB address of which is given in (A8) - 2 (See 'Set Event' request).

Effect

When a program issues this wait request and the event has already occurred, control is returned immediately to the requesting program. If the event has not occurred, the requesting program will be suspended until the event does occur. The event can either be a single one or an OR of several, allowing the requesting program to wait until any event among a list of events has occurred.

Remarks

If bit 15 of A8 is set when this command is issued by a swappable program, it will be swapped out until the awaited event occurs.

Returned Status:

When control is returned to the requesting program, the reply in A7 is as follows:

A7 = 0 The program was in a wait state.
A7 = -1 The ECB address (or one of a chain of addresses) is invalid.
A7 = 1 The event is already set.

LKM 3 - Exit

Purpose

This depends on whether the request is issued by a foreground or background program. In a foreground machine the purpose is to terminate a task, while for the background machine it can be used to request a return to MAS or to condition the execution of subsequent commands in the Job Stream when batch processing.

Foreground Programs

Calling Sequence:

LKM

DATA 3 N.B. Scheduled labels are ignored if used with LKM 3.

Effect:

If issued by a scheduled label routine, then the next scheduled label for this task is entered. If there are no more scheduled label routines for this task, control is returned to the main program at the instruction following the last scheduled label interrupt.

If the request was issued by a foreground program and uncompleted scheduled label routines exist, the program is suspended until all such scheduled label activity has terminated.

If no scheduled label activity remains when this request is issued by a foreground program, the activation queue for the program is examined and the oldest entry, if any, becomes an eligible task. If, however, the activation queue is empty, subsequent action depends on the manner in which the program was declared, e.g.:

- for LOD: Nothing happens.
- REP: Nothing happens.
- SWP: The swap image on disc and the pages it occupies in memory are freed. The initial core image on disc remains.
- RON: The pages in memory are freed, the core image on disc remains.

Middleground Programs

When middleground programs exit, they are disconnected and their memory pages freed. The swap image on disc (if any) is deleted, even though the program may be reactivated later.

Background Programs

An LKM 3 request can be issued by a scheduled label routine on termination, or by a background program in order to return control to MAS. The background program may be the Batch Control Processor (BCP), another type of processor, or a user program.

Calling Sequence:

[LDK A7,L]

[LDKL A8,M]

LKM

DATA 3

L = A BCP Severity Code or Exit Code (see below).

M = The address of a Job Parameter Table (JPT) if the requesting program was the BCP.

Effect:

If the requesting program is the BCP, A7 is not examined, but the JPT pointed to by A8 is accessed. This will indicate either that the :EOB has been encountered and that batch processing is to terminate, or it will contain details of the next program to be executed (see Part 3, The Batch Machine).

If the requesting program is another Processor, or a user program, A8 is ignored and A7 is examined. The following bits have significance:

Bit 8 = 0 No post mortem dump is produced.

Bit 8 = 1 The program has aborted and a post mortem dump is produced in accordance with the value of the DUMP keyword specified on the BCL RUN or Processor call command, i.e.:

DUMP=ALL: Both system and background machines are dumped.

DUMP=PROG: Only the background machine is dumped.

DUMP=NO: No dump is produced.

Bits 9-15 These contain an exit code, or severity code, which is used by the BCP to determine subsequent processing of the job stream. The value ranges from 0 to /7F according to the severity of the error leading to the exit command. The following conventions should be adopted by user programs:

Code	Meaning
0	No error; normal termination.
/01-/2F	Minor Errors (see Chapter 8).
/30-/3F	Errors which require subsequent user programs to be ignored.
/40-/4F	Errors preventing the execution of a Linkage Edit (e.g. a compilation error causing no object code to be produced).
/60-/6F	Language Processor errors.
/70-/7F	Serious errors requiring further batch processing to be terminated (e.g. disc overflow, dynamic area overflow).

If the requesting program is attached to a scheduled label routine, control returns to the interrupt point in the main task, unless there is more outstanding scheduled label activity to be completed. In this case all such activity is completed before the return is made, the program being suspended, if necessary, while awaiting completion of the service on which the scheduled label routine is dependent (e.g. an I/O request).

If the exit code is /FFFF, the execution of any part of the program is suspended. It has the same effect as LKM 46 (Abort) except that the activation queue, if any, is re-started as soon as any pending events have occurred.

On exit of any foreground or background program, the exit code, i.e. the value of A7 upon LKM 3, is printed on filecode /01.

LKM 4 - Get Dynamic Buffer

Purpose

To acquire work areas, scratchpad areas, or I/O buffers; used mainly by re-entrant programs which must safeguard data against corruption when interrupted.

Calling Sequence

A7 is loaded, either with zero or the length of buffer required (expressed in characters), and the sign bit is set as follows:

- bit 0 = 1: The program is to be suspended until buffer space becomes available.
- bit 0 = 0: Control is to be returned to the requesting program immediately if no buffer space is available.

The following statements are then issued:

```
LKM  
DATA [-]4  
[DATA M]      where M is the address of a scheduled label routine.
```

Effect

- 1) A7 originally non-zero

If the request is successful, A7 is zero on return to the calling program while A14 contains the address of word zero of the acquired buffer, whose layout is:

Byte	Contents
-6	Chain Word
-4	The original contents of A14
-2	The length of the Buffer (L)
0	First word of the buffer <- (A14)
2	.
4	.
.	.
L-2	Last word of the buffer.

N.B. The contents of A14 are restored when the buffer is released.

If the request is unsuccessful and the sign bit of A7 was originally zero, then control is returned to the calling program with A7 = 1.

If the request is unsuccessful and the sign bit of A7 was originally 1, the program is suspended until buffer area becomes available. In the case background programs, if this period exceeds the TIME parameter specified in the program's Job Parameter Table, then the program will be aborted.

- 2) A7 originally zero

In this case the effect is different for background and foreground machines:

- a) Foreground Machines: No dynamic buffer is obtained, but:
 - for memory-resident programs A7 returns with the end address of the segment containing the program.
 - for disc-resident programs A7 returns containing the address of the last word in the last page allocated to the program.

Remarks

Dynamic areas are allocated in Segment 0 of the foreground machine. Programs in Segment 0 must not address the dynamic area using virtual addresses.

- b) Background Machines: Again, no dynamic buffer is acquired, but A7 returns containing the highest address in the background machine.

Remarks

- 1) A background program which issues an LKM 4 or assumes that it can use virtual addresses to access memory beyond the load module must be activated by a BCL RUN or non-standard processor call command in which the SIZE parameter specifies enough pages additional to the load module to accomodate the additional areas required.
- 2) For memory-resident background programs, dynamic buffer areas are allocated from the pages exclusively reserved by the SCL DCB command, which are not being used by the currently active background program.
- 3) For disc-resident programs, dynamic buffers are allocated from the common dynamic loading area, shared by all machines, to a maximum of 16 pages.
- 4) If the value in A7 is set to zero, the system returns the highest address in the segment in A7

On return, the following error status may be returned in A7:

A7 = -2: chain link in dynamic area destroyed.

LKM 5 - Free Dynamic Buffer

Purpose To release the buffer areas acquired by the LKM 4 request.

Calling Sequence

A14 should be pointing to word zero of a buffer obtained as a result of a previous LKM 4 request. If it is not, it must be set up to do so.

LKM

DATA [-]5

[DATA L] where L is the address of a Scheduled Label Routine.

Effect

If the buffer is successfully freed, the contents of A14 are restored from location -4 of the buffer (where they were saved by the LKM 4 request), and A7 returns with a status code of zero.

If A7 = -1 when control is returned, then A14 was found to contain an invalid buffer address when the request was issued.

If A7 = -2, the chain in the dynamic area is destroyed.

Remarks

If the dynamic area was in an 'overflow state', i.e. there is at least one program suspended waiting for buffer space, then this program will restart when the LKM 5 request is actioned.

Purpose

To output a message on the operator's console and to suspend the calling program pending an operator's restart (RS) command.

Calling Sequence

Load A7 with the message length in characters.
Load A8 with the address of the message buffer.
LKM
DATA [-]6
[DATA L] where L is a Scheduled Label Routine address.

Effect

The program is suspended and the message that appears on the operator's console (System Machine Filecode /EF) has the following format:

`MACHINE m PRO: p PSE'
`console message'

where `m' is the machine name

`p' is the program name

and `console message' is a string of up to 72 ASCII characters, the first two of which should be the control characters /ODOA or spaces (since MAS will substitute /ODOA for the 2 leading spaces). These control characters can also be embedded in the message to introduce newlines, since they have the effect of positioning all the characters following them in the message text at the beginning of the next line.

A status code is returned in A7 as follows:

A7 = 0: Request was successful.

A7 = -1: The address in A8 was invalid.

A7 = -5: Dynamic Area overflow; the message could not be sent and the program has not been suspended.

Note: Swappable programs will be swapped out.

LKM 7 Keep Control on Abort

Purpose

To retain control in the event of MAS deciding to abort a program or after an LKM 3 (Exit) has been issued. Instead, control is returned to an address specified by the program issuing this request.

Calling Sequence

Load A7 with a 'User Abort Label'
Load A8 with the 'Abort Control Block Address'
LKM
DATA [-]7
[DATA L]

Where: L is a Scheduled Label Routine Address.
- 'User Abort Label' is the address to which control is returned in the event of an abort condition arising.
- 'Abort Control Block' is a four word area set up by MAS when an abort condition occurs subsequent to the issuing of this request. The format of this control block is:

	0	7	8	15
0	Zero		Abort Code	
2	Program Status Word (PSW)			
4	Instruction Counter			
6	(A8) when the program was aborted			

Effect

When this request is received MAS alters the program's control table (PCT) so that whenever an abort condition arises subsequently, control will be returned to the address specified in A7, the User Abort Label.

When an abort or exit condition arises and control is transferred to the Abort Label address, MAS enters the Abort Control Block address in A8 (all other registers are saved) and the instruction counter is loaded with the User Abort Label. These values are also entered in the Abort Control Block.

The abort codes in the Abort Control Block have the following meanings:

- /01 Too many Scheduled Labels
- /02 Invalid Instruction.
- /03 Memory Protect Violation.
- /04 Dynamic Area Corrupt.
- /05 Not Used.
- /06 Operator Abort.
- /07 Too Many Blocking Buffers Requested.
- /08 Disc Overflow.
- /09 Disc Queue Overflow.
- /0A Memory Overflow During Program Load.
- /0B BCL RUN: TIME Exceeded.
- /0C BCL RUN: PRINT Exceeded.
- /0D BCL RUN: PNCH Exceeded.
- /0E Floating Point Error
- /0F LKM 46 Issued (Abort)
- /10 :JOB/:EOJ Read by Batch program.
- /11 I/O Error in loading the program root.
- /12 I/O Error in loading an overlay segment of this program.
- /13 Fatal error in spooling the I/O request.
- /14 Debug error.
- /15 Debug fatal error exit.
- /16 Program aborted by another program in the same machine via LKM 76.

Remarks

- 1) A re-entrant program has one Abort Label for each task, i.e. for each PCT. The LKM 7 request only affects the Abort Label of the issuing task.
- 2) If a program issues several LKM 7 requests, only the last one is effective.
- 3) A program does not have to re-issue an LKM 7 after an abort, unless it is desired to alter the effect; multiple aborts will be handled by the most recently specified User Abort Label.
- 4) If the Abort Code is /06 (Operator Abort), the program must ensure that it is aborted by issuing an LKM 3 with an appropriate severity code in A7.
- 5) Immediately on entering the Abort Label routine, an LKM 31 (Cancel Retain Control on Abort) request should be issued, as a loop would result if an abort condition occurred within the routine itself.
- 6) Abort codes /0B, /0C, /0D and /10 apply only to the background machine.
- 7) The implementation of this request under MAS is slightly different from that under other systems, as the contents of A7 and A8 have had to be interchanged. Scheduled Label processing requires that A8 points to the Abort Control Block instead of to the Label, as with the DOS/BOS systems.

Returned Status

A7 = 0	Request Recorded.
A7 = 2	System Machine Dynamic Area overflow; the request cannot be recorded.
A7 = -1	Invalid address(es) in A7 and/or A8.

LKM 8 - Datem II or TMS Request

Purpose

To request Datacommunications I/O operations.

Calling Sequence

```
LDK      A7,R
LDKL     A8,A
LKM
DATA     [-]8
[DATA    L]
```

where: R is a request order
A is the address of a Line Control Block (LCB)
and L is a scheduled label address.

Request orders

(A7) =	1	(A) +	Read with Echo
	2		Read without Echo
	3	(A) +	Read with Echo - with Timeout
	4		Read Without Echo - with Timeout
	5		-
	6		Write
	7		Write with Timeout
	C		-
	D		Change line definition
	E		Get line definition
	10		Stop the exchange
	11	*	Disconnect the line
	12	(S)	Search pattern
	13	*	Wait for a call
	14	(A) -	Accept data
	15		Set timeout.

(S) = Synchronous Lines only
(A) = Asynchronous Lines only
* = Switched lines only
+ = Full duplex lines only
- = Not possible with AMA8 Control Unit under multiplexed input.

Note : All these requests are serviced without wait for completion. For asynchronous lines, if register A7 contains a negative number it means that the user needs extra characters following the termination (automatic request with synchronous lines).

LCB Structure

	0	7 8	15
0	E		Line Code
2	Buffer Address		
4	Requested Length		
6	Effective Length		
8	Status Code		
10	Edit/Convert Table Name		
12			Timeout Value

Table Explanation

Word 1	Buffer Address = Address of the memory area where the transmission block is to be found.
Word 2	Buffer Length = Length of the buffer. This is a number of characters $1 < n < 32767$ (decimal).
Word 3	Transmitted Length. This information is returned by the Monitor on service completion and provides the user with the effective length of the transmitted block.
Word 4	Service Status. This information is returned by the Monitor. It will be meaningful only when the event bit is set (it must not be modified until the event bit is set). No bit set no error detected. Bit 0 set logical line busy 1 Disconnected line (switched lines) 2 Illegal filecode (no correspondence between provided filecode and system physical lines). 3 Illegal request: . Incompatible order . Negative block length 4 Character(s) lost 5 End of carrier detection 6 Timeout request may not be served: . no room in table . timeout value out of range (SYSGEN option) 7 Buffer overflow 8 Transmission stopped 9 Power failure (option) 10 Time over 11 Break detection (Asynchronous lines only) 12 Command refused 13 Parity error (hardware detected) 14 Throughput error 15 Modem not operable.
Word 5 (U)	Terminator/Edit Table Address. This word is to be filled with the Terminator/Edit Table address if a check is to be done, or with zero. Its contents must not be modified before service completion.
Word 6 (U)	Timeout Value. This word provides the interval of time after which the service is posted complete. The contents of this word will be meaningful only with order numbers 3 and 4 (Read with Timeout), 15 (Set Timeout Control), and 7 (Write with Timeout). In the special case of orders 4 and 7, with Synchronous Transmissions, the time control will be inhibited when the first character is input. Unit of timeout control is the tenth of a second. On option the contents of this word may be checked against a max. Time Control value introduced at SYSGEN time.

Remarks

If editing or terminating characters have been defined at SYSGEN, the SCT name must be entered in word 5 of the LCB.

Programs using LKM 8 must be memory-resident.

For more details see P800 Datacommunications User Manuals.

LKM 10 - Connect A Program to a Timer or to the Clock

Applies to: Foreground Programs only.

Purpose

To cause a foreground program to be activated after a given interval and/or at regular intervals afterwards, or to activate a program at a specific time of day and optionally at regular intervals afterwards.

Calling Sequence

A7 is loaded with the address of a Program Name Block.

A8 is loaded with the address of a Parameter Block.

LKM

DATA [-]10

[DATA L]

Where: L is the address of a Scheduled Label routine.

: The Program Name Block is a three-word area containing the Program Name in up to 6 ASCII characters, left justified and space-filled to the right.

: The Parameter Block has a format which varies according to the desired effect of this request:

a) Format 1 - Connect to a Timer

bits	0	1	3	4	15
	0	NTIM			PR
					NC

Where NTIM is the Timer Number:

- 0 - Hour
- 1 - Minute
- 2 - Second
- 3 - 100 MS
- 4 - 20 MS
- 5 - Non-Standard clock

NC is the number of cycles before the first activation (Range: 0 - 32767).

PR is the pulse rate or reactivation rate, from 0 - 2047. When PR is 0, only one activation will occur and then automatic disconnection occurs.

b) Format 2 - Connect to a Clock

bits	0	1	3	4	7	8	9	15
	1	NTIM		HH				PR
				MM				SS

Where NTIM is the Timer Number, as for Format 1.

HH/MM/SS is the absolute time in hours, minutes and seconds at which the first activation will occur.

PR is the reactivation rate, ranging from 0 - 127. If PR is zero, only one activation occurs (as for Format 1).

Effect:

LKM 10 may be used instead of an FCL CNT command to connect, to a timer or clock, a program in the same foreground machine as that in which the issuing program is running.

If Format 1 is used, the program specified in the Program Name Block will be activated after NC periods of length NTIM have occurred, and will be re-activated every PR periods of length NTIM.

If Format 2 is used, the specified program will be activated when the system clock time reaches the specified hour, minute and second, and will be re-activated every PR periods of length NTIM.

Returned Status

One of the following status codes is returned in A7 following execution of the request:

- 0 Connection has been performed.
- /D Invalid address in A7 and/or A8.
- 2 Dynamic area overflow.
- 2 Timer unknown (NTIM was not in the range 1-5), or, if 5, the bare machine does not have the non-standard clock hardware option.
- 3 Program unknown, i.e. not connected to a level or not defined for this machine.
- 4 The program to be connected was a middleground program.

Remarks

- 1) If a scheduled label is used with LKM 10, it is executed immediately MAS has created the timer activation table entry, and then the next sequential instruction in the issuing program is executed. Thus, the execution of the scheduled label routine is quite independent of the activation or execution of the program connected to the timer and does not imply that the connection has been made.
- 2) The program to be connected must be known to the system and reside in the same (foreground) machine as the connecting program.
- 3) When the program is activated by the clock or timer by means of this LKM, A3 will have been set to zero. This provides a means of distinguishing activation by an LKM 10 from activation by FCL RUN or FCL ACT. In addition, the value in A4 can be used to accumulate statistics during the development of such a program:
 - A3 = 0, A4 = -1 : Normal activation
 - A3 = 0, A4 = -3 : Queued activation.

The contents of A4 can be used to check whether the execution time of the program is compatible with the timer interval, for if A4 is -3 when the attempt is made to activate it, it means:

- a) Execution time is longer than the timer interval.
- b) The program has a too low priority.
- c) The program is also being activated by LKM 12, FCL RUN or FCL ACT commands and these activations are disrupting the activation of the program by timer or clock.

- d) If the program requires dynamic buffers, it is having to wait for dynamic area to be released.
- e) The program issues LKM's which require the System Machine dynamic area and this is in heavy demand, causing the program to wait.
- f) Another program is competing for devices required by this program.
- g) The program is waiting unnecessarily for events; the use of event chains, multiple user record areas for LKM 1's without implicit waits and scheduled label routines would reduce execution time.

LKM 11 - Disconnect a Program from a Timer or Clock (Foreground Only)

Purpose

To disconnect the program from one or more timers.

Calling Sequence

A7 is loaded with the address of a Program Name Block.
A8 is loaded with the Timer Number.
LKM
DATA [-]11
[DATA L]

Where: L is the address of a Scheduled Label routine.

'Program Name Block' is a 3-word area in which the user enters the name of the program to be disconnected (left justified and with trailing spaces as necessary).

'Timer Number' is as specified for LKM 10, except that if A8 contains /FFFF the specified program will be disconnected from all the timers to which it is currently connected.

Effect

The current activation of the disconnected program and all activations in its activation queue, together with all future activations by FCL RUN, FCL ACT, LKM 12 etc., will still be processed. Only new activations from a timer or clock are stopped by LKM 11. The program remains connected to a level.

Returned Status

The contents of A7 when control is returned to the calling program are:

- 0 Disconnection performed.
- 1 Invalid address in A7.
- 2 Timer unknown, or the program is not connected to a timer or clock.
- 3 Program unknown.

Remarks

- 1) LKM 11 may be used instead of an FCL DST command to disconnect from a timer or clock a program running in the same foreground machine as the issuing program.
- 2) If a Scheduled Label routine is specified with LKM 11, it will be performed immediately MAS has updated its timer control tables. A return is then made to the issuing program with A7 set as above.

LKM 12 - Activate a Program (Foreground Only)

Purpose

LKM 12 can be used instead of an FCL ACT command to cause a new activation queue entry to be created for a program in the same foreground machine as that in which the issuing program is running.

Calling Sequence

A7 is loaded with the address of the Program Name Block.
A8 is loaded with the address of a two-word Activation Block.
LKM
DATA [-]12
[DATA L]

Where: L is the address of a Scheduled Label routine.

'Program Name Block' is a three-word area in which the name of the program to be activated is entered (left justified and space filled to the right).

'Activation Block' is a two-word area, the first word of which is not used by the system, but can be used as an event word for task synchronisation. The second word of this block is a parameter word, the contents of which are transferred to A4 of the activated program.

Effect

The following information is transferred to registers of the program to be activated:

- To A3, the contents of A3 of the calling program.
- To A8, the contents of A8 of the calling program.
- To A4, the contents of word 1 of the Activation Block.
- To A14, zero.

Remarks

- 1) The program to be activated must have been defined by an FCL RON, SWP, REP or LOD command for this foreground machine and have been connected to a level by an FCL CNL command or by an LKM 20 given from within this foreground machine. Alternatively, it must be a middleground program, in which case, after creating the activation queue entry, MAS will search the directories of the first Userid in all the DAD's defined for this foreground machine in the order /FO to /FF. The program will then be loaded into memory as a swappable program and connected to the lowest unconnected software level.

If the activated program is re-entrant a new entry is created in the Monitor Table and the program can be started immediately. It is automatically connected to the next free level below that specified when the re-entrant program was loaded and connected to a level.

3) Task synchronisation can be specified by using a scheduled label with the LKM 12. The scheduled label routine will be given control when the activated task has exited. Note that if the activated task has been aborted, bit 15 of A8 will be 1 when the scheduled label routine is entered. Meanwhile, the next instruction in the activating task will be executed. When the activating task reaches a point where it cannot proceed until the activated task has completed, it must wait. This is done using the Event Word pointed at by A8 when the LKM 12 was given, as follows:

- Program 1 issues an LKM 12 and the Event Word pointed at by A8 has zero in bit 0. Bit 1 is set to 1 if this event is chained to other events, otherwise it is zero.
- When Program 1 reaches a point where it cannot continue until Program 2 has finished, it issues an LKM 2 with A8 pointing to the event word used by the LKM 12.
- When Program 2 starts, it receives:
 - A3 and A8 of Program 1 at the time it issued the LKM 12.
 - A14 set to zero.
 - in A4, the contents of the second word pointed at by A8 in Program 1 when it issued the LKM 12.
- Just before Program 2 Exits, it sets bit 0 of the Event Word pointed to by A8 to 1 by issuing an LKM 18 (Set Event).
- The scheduled label routine (if any) specified by Program 1 when it issued the LKM 12 will then be performed. When that exits, the scheduled label routine (if any), specified by Program 1 when it issued the LKM 2, will be performed. When that exits the next sequential instruction after the LKM 2 in Program 1 will be executed.

4) If the activated program is in another segment, the parameters and parameter block must be stored in the dynamic area of Segment 0, otherwise memory-protect violation can occur.

Returned Status

One of the following codes will be returned in A7 when execution of the requesting program is resumed:

- 0 Program activated or an activation queue entry has been created.
- 1 Invalid address in A7 and/or A8.
- 3 The program has not been defined for this machine or (in the case of middleground programs) cannot be located in any of the First User DAD Directories.
- 4 System Dynamic Area overflow; a new activation queue entry cannot be created or no space to create a PCT for a middleground program.
- 5 The program has not been connected to a level, or there are not enough free pages to load a middleground program.
- 6 The program has previously been aborted and can only be activated by an FCL ACT or RUN command.
- 7 Too many reactivations for a re-entrant program, or there are no more levels available for a re-entrant program.
- 8 No software level available (middleground programs).
- 9 I/O error during activation of a middleground program.
- 10 D:CI (swap DAD) overflow (middleground programs).

LKM 13 - Switch Program Levels (Foreground Only)

Purpose

To modify the relative priority of up to 30 programs.

Calling Sequence

LDK	A7,0
LDKL	A8,P
LKM	
DATA	[-]13
[DATA	L]

where: L is the address of a scheduled label routine.
P is the address of a three word parameter block containing the program name, left justified and space filled.

Effect

For the purposes of this request, the software levels to which programs are attached may be divided into groups of 30 levels (30-59, 60-89 etc.). Only the levels within the group to which the named program belongs are modified. If no other program exists apart from the one named in the parameter block, then no change occurs; however, if there is an eligible program of higher priority than the named program, then the priorities of these two programs are interchanged. If there are more than two programs, a cyclic shift occurs. The program with the highest priority is re-assigned to the level of the program with the lowest priority in the group; each other program is re-assigned to the level of its next higher program in the group.

Returned Status

The following status codes are returned in A7 on completion of this request:

0	Completed
1	Unknown program name
2	Program not connected to a level
-1	Invalid address in A8.

LKM 14 - Attach a Device or a File

Purpose

To obtain exclusive use of a device or file via its filecode.

Calling Sequence

LDK	A7,M
LDKL	A8,N
DATA	[-]14
[DATA	L]

Where M is a Wait Flag with the following values:

M=0 Control is returned immediately to the requesting program if the device/file is unavailable.

M≠0 The program is suspended until the required device or file becomes available.

N is the address of a word containing the filecode in bits 8-15. The filecode defines the device or file to be attached. A disc or a DAD may not be attached in this way.

L is the address of a Scheduled Label Routine.

Returned Status

A7 contains one of the following codes when control is returned to the calling program:

0 The device or file has been attached.

-2 A8 contains an invalid address or the filecode is not assigned or an attempt is made to attach a DAD, a disc or a TDFM file.

-3 Device is already attached to another task. This status code only applies if A7 was zero when the request was made.

-4 The device to be attached is spooled.

Remarks

1) Using a Scheduled Label routine only affects the return address after the request has been activated.

2) For a re-entrant foreground program which is performing several tasks concurrently, the device or file can only be attached to one of the tasks at a time.

3) Attachment of a disc file is only valid within one machine.

LKM 15 - Detach a Device or File

Purpose

To detach a device or file which has already been attached to the requesting program.

Calling Sequence

```
LDKL      A8,N  
LKM  
DATA [-]15  
[DATA    L]
```

Where N is the address of a word containing the filecode in bits 8 - 15.
L is the address of a Scheduled Label routine.

Returned Status

A7 contains one of the following codes when control is returned to the calling program:

- 0 Device or file detached.
- 2 A8 contained an invalid address, or filecode not assigned or the device or file does not exist or is not attached to this program.
- 1 The filecode is not attached to a device or file.

LKM 17 - Get Time and Date

Calling Sequence

```
LDK      A7,M
LDKL     A8,N
LKM
DATA [-]17
[DATA    L]
```

Where: M = 0 if time is to be returned in ASCII format.
M ≠ 0 if time is to be returned in binary format.
N = Address of a Time Block.

The Time Block is a 6-word area having the following format:

	A7=0	A7#0
0	Day	Hours
2	Month	Minutes
4	Year	Seconds
6	Hours	Tenths of Seconds
8	Minutes	Fiftieths of Seconds
/A	Seconds	Pulse Rate.

This Time Block is completed by MAS before control is returned to the calling program.

The Pulse Rate is zero if the standard clock is used, otherwise it is the number of non-standard pulses per standard pulse as determined at Sysgen when the non-standard clock was generated.

Returned Status

When control returns to the requesting program A7 will contain one of the following codes:

- 0 OK
- 1 A8 contained an invalid address.
- 1 A clock interrupt occurred during the processing of this request which caused a change of day to occur. The Day entry in the table is the previous day and will have bit 0 set to 1. The time however is correct. Another LKM 17 can be issued to get the correct date.
- 2 The date has not been initialized.
- 3 The clock has not been initialized.
- 4 Neither date nor clock has been initialized.

LKM 18 - Set Event

Purpose

To inform MAS that an event has occurred, so that the system can restart all programs that are suspended awaiting that event.

Calling Sequence

```
LDKL   A8,N
LKM
DATA [-]18
[DATA  L]
```

Where: N is the address of word 1 of an Event Block.
L is the address of a Scheduled Label routine. The scheduled label will be performed before the next sequential instruction in the calling routine when control is returned to the requesting program after completion of the LKM.

The 'Event Block' is a two word table having the following layout:

bits	0	1	15
	Address of linked Event Block (only if L=1)		
(A8)->	E	L	

Where: E is the Event Bit and is set by the Monitor to indicate that the event has occurred.
L is zero for a single event and 1 for a chained event. If L is zero then word 0 is unused, since in this case the event block is the last or only block in the chain.

Effect

For a single event MAS sets bit 0 of the word whose address is contained in A8 (i.e. word 1 of the Event Block).

For a chained event MAS will set bit 0 of the word pointed to by A8 and pick up the address of the next event block from word 0 of this Event Block. The chain is followed until the chained event bit (L in the diagram above) is found to be zero.

Returned Status

The status code returned in A7 upon completion of this request is either:

0 Event bit(s) set.
or -1 Invalid address in A8.

Remark

The address in A8 is a 16-bit relative or logical address. Care must be taken that the address given in A8 is not coincidentally the same as that in the calling sequence of another event handling procedure in a different segment of the same machine, otherwise the result will be unpredictable.

LKM 20 - Connect a Program to a Level (Foreground Only)

Purpose

To connect a program to a software level; this is required before the program can be activated. This request can be used instead of the FCL CNL command.

Calling Sequence

LDK	A7,N
LDKL	A8,M
LKM	
DATA	[-]20
[DATA	L]

Where: N is the level number to which the program is to be connected. It must be numerically less than the number of the idle task and must not already have a program connected. For a re-entrant program the level number specifies the minimum software level to which any task performed by the program is to be connected. MAS will connect any new task to the lowest unconnected software level above this minimum.

: M is the address of a Program Name Block, which is a three word area containing the Name of the program to be connected, to a maximum of 6 ASCII characters, left justified and space-filled to the right. This program must have been previously defined and must not be already connected to a level.

: L is the address of a Scheduled Label routine.

Returned Status

On return to the requesting program A7 will contain:

0	Connection made
-1	Invalid address in A8
-2	Program is already connected
-3	Program is unknown
-5	Level too high.

LKM 21 - Disconnect a Program from a Level (Foreground Only)

Calling Sequence

```
LDKL      A8,N  
LKM  
DATA [-]21  
[DATA    L]
```

Where: N is the address of a three word Program Name Block, in which the name of the program to be disconnected is entered in ASCII characters, left justified and space-filled to the right.

: L is the address of a Scheduled Label Routine.

Effect:

The program is disconnected from its level and also from any timers that may be connected to it.

When the program has not been connected to a level no action is performed and no error status is given.

Returned Status

When control is returned to the requesting program (which occurs when the request and any associated Scheduled Label routine has been completed), A7 may contain one of the following codes:

- 0 Disconnection performed or program not connected.
- 1 Invalid address in A8.
- 2 Program is active (running, in wait state or aborted).
- 3 Program unknown.

LKM 22 - Wait for a Given Time (Foreground Only)

Purpose

To suspend a program until either a specified time has elapsed or an event bit in an event block or chain of event blocks has become set.

Calling Sequence

```
LDKL      A8,N
LKM
DATA [-]22
[DATA     L]
```

Where: L is the address of a Scheduled Label Routine.

- N is the address of word 1 of an Event Block having the following layout:

bits	0	1	2	3	4	15	
word -2	Address of linked Event Block (if L = 1)						
word 0	E	L					<-(A8)
word 2	0	NTIM					
word 4	NC						

In this table:

- L is set if this is one of a chain of event blocks, in which case the address of the next is given in word -2.
- E is the event bit.
- NTIM is the length of the wait time unit (bits 1-3, bit 0 is zero):
 - 0 = hours
 - 1 = minutes
 - 2 = seconds
 - 3 = 100 ms
 - 4 = 20 ms
 - 5 = 1 non-standard clock period.
- NC is the number of periods, defined by NTIM, which must elapse before the program is reactivated.

Effect

The issuing program is suspended until either the specified time has elapsed, or an event bit in the chain of event blocks (if any) has become set.

Returned Status

When the program suspension ends, control is returned to the scheduled label associated with the LKM 22 (if any) and upon completion of this, to the calling program. The status code returned in A7 when the program suspension ends may be one of the following:

- 0 Wait has ended.
- /D Invalid address in A8.
- 2 Invalid timer number.
- 2 System dynamic area overflow.

Notes: If the DAD Filecode (word 1) is entered as zero, the effect depends on whether the requesting program is background or foreground:

- a) Background; the file will be created on the DAD specified on the BCL :JOB command.
- b) Foreground; the file will be created on DAD /FO or the defaultDAD set by the JOB command or LKM 73

: If the file is to be created by a Direct Write LKM 1, the number of granules must be allocated at Assign time.

3) Type 2 - Assign a Filecode to the latest Version of a Catalogued File

bits	0	1	2	7	8	15
Word 0		P		Assign Type		Filecode
1	DAD File Code (/FO - /FF)					
2	File Name (characters 1-2)					
3	File Name (characters 3-4)					
4	File Name (characters 5-6)					
5	File Type					

Assign Type = 2, but bit 1 may be set (giving an Assign type of /42), in which case MAS will set the Write Protect Flag in its tables and thus prevent other tasks writing to this file until the background job or foreground task has ended.

DAD Filecode Either the DAD in which the file is contained or zero. If zero, then:

- a) for foreground programs: DAD /FO or default DAD.
- b) for background programs: the DAD specified on the BCL :JOB is assumed to contain the file. The file is always searched in the current JOB userid, so the DAD filecode must contain zero or the current JOB DAD.

File Name is contained in words 2-4 and consists of up to 6 ASCII characters, left justified and space-filled to the right.

File Type Is one of the standard mnemonics; see above (Type 1 Assign).

Note: For foreground machines, the library of the directory of the first entry in the DAD catalogue is scanned to locate the file, the default DAD and Userid set by the JOB command or LKM 73 are taken into account.

4) Type 3 - Assign a Filecode to Another Filecode

0	7	8	15
Assign Type		Filecode 1	
Filecode 2			

Assign Type = 3

Filecode 2 contains a filecode to which Filecode 1 is to be assigned, but note:

- a) Filecode 2 must already be assigned and not by a Type 3 assignment, i.e. a filecode cannot be made equivalent to another which is itself equivalent to a third.
- b) Filecode 1 must not be a permanent filecode of the batch machine, i.e. one declared by an SCL FCD command.
- c) If Filecode 2 gets a new assignment, Filecode 1 automatically gets the same assignment.

5) Type 4 - Assign a Filecode to a Specified Version of a Catalogued File

The Assign Block is as for Assign Type 2, with an additional seventh word containing the version number, a value ranging from 0 (the latest version) to 7 for the oldest version. The only other difference is the Assign Type which becomes 4.

6) Type 5 - Assign a Filecode to Another Users Disc Catalogued File

word	0	7 8	15
0	Assign Type	Filecode	
1	DAD Code		
2	Filename (characters 1-2)		
3	Filename (characters 3-4)		
4	Filename (characters 5-6)		
5	Filetype		
6	Version		
7	Userid (characters 1-2)		
8	Userid (characters 3-4)		
9	Userid (characters 5-6)		
10	Userid (characters 7-8).		

- Assign Type = 5
- DAD Code The Filecode of the DAD on which the file is to be created. If this is zero, then:
 - a) Foreground programs; DAD /FO or the default DAD is used.
 - b) Background programs; the DAD specified on the BCL :JOB command or the DAD that the BCP evaluated as the JOB DAD. The DAD catalogue is then searched for the USERID specified in words 7 - 10 of the Assign Block. The User Directory is then searched for the Filename, Filetype and Version specified in Words 2 - 6.
- The Filetype is one of the standard mnemonics given for Assign Type 1.

first Userid in the DAD. The default is taken, when the first word of the Userid field is left zero.

7) Type 6 - Assign a Filecode to a TDFM Descriptor File

word	0	7 8	15
0	Assign Type	Filecode	
1	DAD Filecode		
2	Filename (characters 1-2)		
3	Filename (characters 3-4)		
4	Filename (characters 5-6)		
5	Zero		
6	Zero		
7	Userid (characters 1-2)		
8	Userid (characters 3-4)		
9	Userid (characters 5-6)		
10	Userid (characters 7-8).		

Assign Type = 6

DAD Code = The Filecode of the DAD on which the file is to be created (see Assign Type 1).

Remarks

- a) **Foreground Machines:**
If this Filecode has already been assigned (by an SCL FCD or FCL ASG command or an LKM 23 or LKM 33 request), that assignment is replaced. The user must ensure that no other program in this machine will require to use the old assignment for this filecode.
- b) **Background Machines:**
The assignment only affects the JOB Filecode Table maintained by MAS. When the next BCL :JOB command is read, the JOB Filecode Table is reset to the values originally specified by means of the SCL FCD commands when the background machine was defined.
- c) When a filecode is assigned, re-assigned or deleted, whether by FCL or BCP command or by LKM request, the system does not check whether or not it is being used.
- d) Before assigning the filecode, the system will delete the old assignment if any, then try to make the new assignment. Thus when the assign cannot be done, the old assignment no longer exists, and the User has to issue a new request to re-establish the old assignment if necessary.
- e) The assignment of the filecode remains unchanged until the next assignment is requested, either by another assign request or by a control command.
- f) When an assignment is made to a TDFM file, only the descriptor file need to be specified. The subfiles (KEY and DATA files) are assigned automatically. The subfiles are searched via the volumenumber of the disc where they are created. This volumenumber is recorded in the descriptor file. When a user mounted two discs with the same volumenumber, the subfiles may be searched on the wrong disc, which can lead to 'strange' errors.

Returned Status

When control is returned to the requesting program one of the following status codes will be returned in A7:

Value	Meaning
0	Assignment made
1	Disc I/O Error
2	Dynamic Area Overflow; filecode table cannot be created.
3	DAD unknown
4	Device unknown or specified devicename-devicedress not existing.
5	Required granules unavailable
6	Filename unknown
7	Second filecode unknown or already assigned by equivalence
8	Permanent Filecode must not be assigned
9	Invalid Assign type
/A	Userid not found in DAD catalogue
/B	Illegal Filetype (not SC, UF, OB or LM)
/C	Illegal Filecode (/CO-/CF or /FO-/FF)
/D	Filecode already assigned to an attached device or file or illegal address in A8.
/E	Too many filecodes (more than system default or MFC command value)
/F	Too many granules requested for the sector size (for non-consecutive files) or greater than 32K sectors (consecutive files) or specified number of granules negative.
/1D	Unauthorized assignment made to a file.
/8002	I/O pending on the filecode to be re-assigned (i.e. file code is in use)

/100 Userid of one subfile unknown
/101 Name of one subfile unknown
/102 Dynamic area overflow in System Machine
/103 One pack not on line
/104 DAD of one subfile unknown
/105 DAD of one subfile not assigned
/106 ECB not completely in user program
/107 Unknown filecode for DAD of descriptor
/108 File already assigned in another machine with non-identical DAD
filecodes for subfiles
/109 File protected, but session is without security back-up
/110 File protected in back-out mode, but back-out file not on-line
/111 FCT entry of Disc of Descriptor not found
/112 File already assigned, but no entry in FCT for DAD of descriptor.
/113 Filecode was assigned to a still accessed file.

Note: Even if the assignment request is rejected, the old assignment for this job will have been deleted.

LKM 24 - Delete a Filecode

Purpose

To request the system to remove a filecode which has been previously assigned. The entries of the filecode are freed in the various tables of the system.

Calling Sequence

```
LDKL      A8,N
LKM
DATA [-]24
[DATA     L]
```

Where: L is the address of a Scheduled Label Routine
N is the address of a Delete Word. This is a single word having the following format:

0	7 8	15
Zero	Filecode to be deleted	

Effect

a) Foreground Machines:

The filecode is removed from the Filecode Table for this foreground machine and in addition, if the filecode was assigned to a disc temporary file, the granules allocated will be freed.

b) Background Machines:

The filecode is deleted from the MAS Filecode Table for the current background job, but not from the MAS Filecode Table for the background machine. If the filecode was assigned by an SCL FCD command when the background machine was defined, this filecode will be restored when the next BCL :JOB command is read by the BCP.

If the filecode was assigned to a disc temporary file, the granules allocated will be freed for use by other files.

If the filecode was unknown, the system returns as if it was deleted.

If the filecode was a permanent one, it is set to delete, but the FCTe (file code table) is not freed.

Returned Status

When control is returned to the requesting program, A7 may contain one of the following codes:

Code	Meaning
0	Operation completed
1	I/O error while reading the file to free the granules (temporary disc file), however the filecode will have been removed.
/C	Invalid filecode (/CO-/CF)
/17	Filecode is used by input spool.
/8001	The filecode to be deleted is a DAD filecode with one or more logical files assigned to it.
/8002	The filecode is in use by some pending I/O.

-1 Invalid address in A8.

LKM 25 - Read Unsolicited Key-in

Purpose

To read input from the operator's console (filecode /EF of the system machine) without having to use an LKM 1, which would block the operator's console until the message has been keyed in. The message for an unsolicited key-in is input after a control panel interrupt and is transported by the system to the program, expecting the key-in. The program can either wait for the message (via a LKM 2) or handle it via a scheduled label.

Calling Sequence

```
LDKL      A8,N
LKM
DATA [-]25
[DATA     L]
```

Where: L is the address of a Scheduled Label Routine.
- N is the address of word 0 of an Event Block, the format of which is:

bits	0	0	1	2	15
Word	-1	Address of next ECB in the chain (if L=1)			
	0	E	L		
	1	Buffer Address for operator message			
	2	Maximum message length			
	3	Actual Message Length (set by MAS)			
	4	Reserved			
	5	Identity Characters			

Where: E is the Event Bit which is set by MAS when the key-in occurs.
- L is the chained event-block indicator. Zero if this is the last or only block in a chain of Event Blocks; if set to 1, word -1 of the block contains the address of the next block in the chain.
- 'Actual Message Length' in word 3 of the table will contain either the character count of the input message or zero. If zero, it indicates that an LKM 26 has been issued cancelling the Key-In request.
- The 'Identity Characters' in word 5 are those used in the operator KI message to remove the LKM 25 request. They serve to identify which LKM 25 the message refers to when there are several LKM 25 requests outstanding.

Effect

The request is recorded by the system and placed in a queue.

The user may use an LKM 2 (Wait for an Event) to suspend his program until the key-in occurs, or he may use a Scheduled Label routine with the LKM 25. In this latter case, the scheduled label will be started as soon as the key-in occurs; in the meantime processing is not suspended.

Remarks

1) If the program issues several LKM 25 requests with the same identity characters, they will be serviced on a first-in first-out basis. As one is serviced, it is removed from the queue, word 3 of the associated event block is updated and the event count is decremented.

- 2) If word 3 of the event block is zero on return, it means that the request was cancelled by means of an LKM 26. In such a case, any scheduled label routine associated with the LKM 25 is started immediately.
- 3) An LKM 25 must not be issued by a re-entrant program, since the KI message does not identify the PCT required.
- 4) A task cannot Exit if it has an outstanding LKM 25 request; the request must first be cancelled, either by an LKM 26 or an operator's KI command.

Returned Status

A7 = 0	Request recorded.
A7 = -2	Invalid address in A8 or in the Event Block
A7 = 2	Dynamic area overflow; request not recorded.

LKM 26 - Cancel 'Read Unsolicited Key-in' Request

Purpose

To cancel an LKM 25 request (Read Unsolicited Key-in). This may be necessary, for example, since the latter request can be issued asynchronously and if no key-in is received, the program will be unable to Exit.

Calling Sequence

```
LDKL      A8,N
LKM
DATA [-]26
[DATA    L]
```

Where: L is the address of a Scheduled Label routine.
- N is the address of an Event Block, the format of which is as described for the LKM 25 request.

Effect

The request is removed from the queue, the event count is decremented and word 3 of the Event Block is set to zero. The Scheduled Label routine associated with the LKM 25 is then performed.

Return Status

```
A7 = 0 Request cancelled.
A7 = -1 Invalid address in A8.
A7 = 1 No previous LKM 25 for this Event Block issued by this task.
```

LKM 27 - Load an Overlay Segment

This request is generated automatically by the LKE Standard Processor for overlaid programs.

Calling Sequence

LDKL	A8,N
LKM	
DATA	27

Where: N is the address of a Load Block, the layout of which is as follows:

	0	7	8	15
0	Event Byte		Ascendant No.	
2	Load Address of the Segment			
4	Sector Address of the Segment			
6	Effective Length			

Returned Status

Upon completion of the request, control is returned with one of the following status codes in A7:

A7 = 0 Segment has been loaded
A7 = -1 Invalid address in A8
A7 = -16 I/O error during load, the program is aborted with abort code /12.

Remark

The sectors of the overlay segment are loaded from the original load module. Every time a new granule has to be accessed the GRANTB is read in. So some performance can be gained when the load module is a consecutive one, because in that case no GRANTB reading is to be performed.

LKM 28 - Set Timer (Foreground only)

Purpose

To cause an event to occur after a predetermined time.

Calling Sequence

```
LDKL      A8,N
LKM
DATA      [-]28
[DATA     L]
```

Where: L is the address of a Scheduled Label Routine.
- N is the address of word 0 of an Event Block, the layout of which is as follows:

Word -1	0	1	3	4	15
	Address of linked ECB (if L=1)				
0	E	L			
1	0	NTIM			
2	NC				
3	RT				

Where: E is the Event Bit, set by MAS when the specified time interval has elapsed.
- L is the 'linked block' indicator; it is zero if this is the last or only Event Block in the chain, otherwise 1.
- NTIM is the Timer Number, as follows:
0 = hours
1 = minutes
2 = seconds
3 = 100 ms
4 = 20 ms
5 = non-standard clock.
- NC is the number of NTIM time periods required to elapse before the Event Bit is set.
- RT is the 'reset timer' indicator. It is set to zero by MAS when the Event Bit is set and set to 1 by an LKM 29 (Reset Timer) request which specifies this Event Block.

Effect

After 'NC' cycles of 'NTIM' units of time have elapsed, MAS sets the event bit 'E' and the linked event bits in any chained blocks, as indicated by the value of L.

Remarks

- 1) The issuing task is not suspended when this request is recorded, but when the event occurs any scheduled label routine associated with the LKM 28 is started.
- 2) The issuing task may follow the LKM 28 with an LKM 2 (Wait for Event) request using the same event block. If both LKM's use scheduled label routines, the following sequence of events occurs when MAS sets the event bit:
 - a) The scheduled label for the LKM 28 is performed
 - b) The scheduled label for the LKM 2 is performed
 - c) The next sequential instruction in the issuing program is performed.

It can be seen that the event block may be used to define an OR of the timer and another operation, for example an I/O operation. This is particularly useful in data communications programming, to determine either that an I/O operation is complete or that the terminal concerned has timed-out. For instance, suppose that an LKM 1 (I/O request) is issued and the I/O event block is chained to another event block. An LKM 28 is then issued to signal the end of the timeout period and the event block of this LKM 28 is chained to the same event block as that to which the I/O request is chained.

This latter event block therefore defines an OR of the timer and the I/O requests. When either completes, the event bit will be set.

An LKM 2 is issued (specifying this common event block) when execution of the program reaches a point beyond which it cannot continue until either the transfer is completed or the terminal is to be considered inoperable. If the I/O operation is completed, then the scheduled label associated with the LKM 1 is performed; if the device times-out, then the scheduled label routine associated with the LKM 28 is performed. When control is returned to the calling program, therefore, one of these two scheduled label routines will have been performed.

- 3) When the LKM 28 request is recorded, the event count is incremented. It is decremented again when:
 - a) the specified time has elapsed.
 - b) an LKM 29 (Reset Timer) request is received before the time has elapsed.

In both cases the event bit will be set and any associated scheduled label will be started. The value of RT (Word 4 of the Event Block) indicates which of these two cases applies, since RT = 1 if reset by an LKM 29 and zero if the specified time has expired.

Returned Status

When control is returned to the next sequential instruction to the LKM 28, A7 will contain one of the following status codes:

- 0 The request is recorded.
- /D Invalid address in A8.
- 2 The Timer Number (NTIM) is invalid.
- 2 System dynamic area overflow.
- 4 Set timer request done by a middleground program.

LKM 29 - Reset Timer

Purpose

To cancel the effect of an LKM 28 (Set Timer) request.

Calling Sequence

LDKL	A8,N
LKM	
DATA	[-]29
[DATA	L]

Where: L is the address of a Scheduled Label Routine.
- N is the address of word 1 of an Event Block, the format of which is described under LKM 28 - Set Timer.

Effect

The event count is decremented, the event bit is set in word 1 of the Event Block and word 4 of the Event Block is set to 1. Any associated scheduled label routines are then performed before control returns to the instruction following the LKM 29.

Returned Status

A7 = 0	Reset performed
A7 = -1	Invalid address in A8
A7 = 1	No previous LKM 28 (Set Timer) request has been recorded for the specified timer.

LKM 30 - Queue Handling Request

Purpose

To allow the system to take over the routine maintenance of user queues of data areas. It relieves the user of the need to chain sequential message buffers, for example, or to rechain as processing is completed or new messages are received.

Calling Sequence

LDK	A7,N
LDKL	A8,M
LKM	
DATA	[-]30
[DATA	L]

Where: L is the address of a Scheduled Label Routine.
- M is the address of an Event Control Block.
- N is a function code indicating the service required. The function code may have the following values:
1 Put an area into the queue.
2 Get the next area in the queue.
3 Cancel the 'Get Next' request.
4 Release the current area in the queue.

The format of the ECB pointed to by A8 varies according to the function requested.

1) Put an Area into the Queue

ECB	0	15
0	Not Used	
1	Buffer Address	
2	Number of Queues	
3	Status	
4	QNAM1 (3 words)	
5	being the name of	
6	the first queue	
7	QNAM2	
.	.	
.	.	
.	.	
.	.	
N	QNAMn	

Where: Word 1 contains the address of the buffer which is to be placed in the queue. These buffers are described below.

- Word 2 specifies the number of queues into which the buffer is to be placed.
- Word 3 contains a status code, returned by the system after processing the request. These are detailed below.
- Words 4-6 contain the first Queue Name as 6 ASCII characters, left justified and space-filled to the right.
- Words 7-N contains Queue Names 2-'n', indicating the names of the queues into which the buffer is to be placed.

2) Get the Next Area in the Queue

	0	1	15
ECB 0	E	L	Event word
1	Buffer Address		
2	Not Used		
3	Status Code		
4	Queue Name,		
5	up to 6 ASCII		
6	characters.		

Where: E is the Event Bit, set when MAS has processed the request.
 - L is the Link Indicator, set to 1 if this ECB is linked into a chain of ECB's.

- Word 1: the address of the first word of user data. This is returned by MAS when E is set.
- Word 3 contains the status returned by MAS upon completion of the request (see below).
- Words 4-6 give the queue name from which the next data area is to be retrieved.

3) Cancel 'Get Next' Request

	0	15
ECB 0	Not used	
1	Not used	
2	Not used	
3	Status Code	
4	Queue Name for	
5	which the Get Next	
6	request was issued	

The status codes returned by MAS when this request has been completed are described below.

4) Release the Current Area in the Queue

	0	15
ECB 0	Not used	
1	Buffer Address	
2	Not used	
3	Status Code	
4	Not used	
5	Not used	
6	Not used	

Where: Word 1 contains the address of word 0 of the buffer to be released.

- Word 3 will contain a status code inserted by MAS when the operation is complete. These status codes are described below.

Format of the Buffer Area

The memory areas to be placed in the queues handled by the LKM 30 requests have the following format:

Word -2	Reserved for MAS and LKM 4	
-1	F	Length of this data area
0		First word of user data
1		----->
-		
-		
n		
Entry for A		Pointer to this word in next buffer
Queue 1		in this queue
	B	Pointer to start of User Data (word 0)
Entry for A'		as for A
Queue 2	B'	as for B
	C	Number of Queues (QCNT).

The buffer shown above assumes that the area has been placed in two queues, QUEUE1 and QUEUE2. Location C (QCNT) therefore contains 2.

The two words A and B are repeated as many times as necessary, so that there is one pair for every queue in which this area has been placed by the system.

The user must not change any of the data outside words 0 - n until the buffers are released (QCNT = 0), as these areas are used by MAS in the queue handling process.

All the user has to do, before issuing his first PUT request, is to ensure that there are enough words in each buffer area to contain two words for each queue entry, plus one word for the QCNT and two words at the beginning of the buffer.

In the diagram showing the buffer format, if:

F = 0 The buffer belongs to the dynamic area and must be released as soon as it has been processed.

F = 1 the buffer is not to be released even when it has been completely processed.

Processing

The buffers are chained and removed from the queues on a first-in, first-out basis. For each machine several queues may be used, each of which is identified by its name (unique within one machine) and entered into the system's control tables when the first request is made naming that queue. It is not necessary to declare it at the time the machine is specified.

Put in Queue Request (A7 = 1)

This request is used to chain the buffer to the last one already in the queue. If the queue name is not known to the system, MAS will create a new entry in its Queue Table. The user must enter the queue name in his ECB together with any other queue names to which this buffer belongs, and with the total count of queues.

This request is always processed immediately, and the calling program resumes processing as soon as the request is entered and the status code has been returned.

Get Next Area in the Queue Request (A7 = 2)

This request indicates to MAS that the previous area has been completely processed (either by the issuing program or another within this machine using the same queue name). The QCNT of the previous area is decremented, and if it reaches zero and if F is also zero it will be released. Then the next buffer will be obtained.

If the queue is empty, the request will be recorded (even if the queue name is unknown, in which case a new entry will be made in the system's Queue Table), but the issuing program will not be suspended. The user can check for the completion status by means of an LKM 2 (Wait) using the ECB or, better still, a scheduled label routine.

When the 'Get' request is recorded the event count is incremented; it will be decremented when the requested area is retrieved.

Cancel 'Get Next Area' Request (A7 = 3)

This is used to cancel the effect of the previous 'Get Next Area' request, and it also indicates that the current area is ready for release. The system will then decrement the queue count.

Release the Current Area in the Queue (A7 = 4)

This is used to indicate that the area last retrieved by a 'Get Next' request has been processed and is to be released. QCNT will be decremented.

Returned Status

The status codes returned by MAS in word 3 of the ECB have the following meanings:

- 0 Request completed.
- 1 Invalid address in the control block.
- 2 Buffer cannot be released (because, for example, it is a data area within a user program).
- 1 'Cancel Get Next' received. This status code is returned either to the scheduled label routine associated with the 'Get Next' request, or to the LKM 2 associated with the ECB used by the 'Get Next' request.
- 2 A 'Cancel Get Next' request has been issued, but the queue of 'Get Next' requests is empty.
- 3 The buffer is too short to contain all the chain pointers.
- 4 The 'Get Next' request has been queued (queue is empty).
- 5 Dynamic area overflow. Request not performed.
- 6 Current area has been released.
- 7 'Release Current Area' request invalid.

In A7, -1 is returned when A8 contained an invalid address or when A7 contained an invalid order (<1 or >4).

Remarks

The buffers belong to the user machine and are accessible by user programs. They can be acquired either by LKM 4 (Get Dynamic Buffer) requests or included as data areas within a program. The user may overwrite the data parts of the buffer areas (words 0 - n in the diagram above), but the other areas should not be overwritten until the buffer is released. The only exception is bit 0 of word -1 (bit 'F' in the diagram), which the user may set to 1 in order to prevent the automatic release of a dynamic area buffer when the queue count (QCNT) becomes zero.

LKM 31 - Cancel 'Keep Control on Abort'

Purpose

To cancel the effect of the previously issued LKM 7 (Keep Control on Abort) request.

Calling Sequence

LKM
DATA [-]31
[DATA L]

Where L is the address of a Scheduled Label Routine.

Effect

MAS removes the 'User Abort' label from the program's PCT (Program Control Table) and returns to the program.

Returned Status

A7 = 0 Request performed.
A7 = /15 No previous LKM 7 was recorded.

LKM 32 - Set/Reset File Attributes

Purpose

To change the attributes of a file or the version number that will be used as the highest version in any 'Keep File' requests.

Calling Sequence

```
LDK      A7,N
LDKL     A8,M
LKM
DATA     [-]32
[DATA    L]
```

Where: L is the address of a scheduled label routine.
- N is a number defining the function required:
A7 = 0 Set specified bits to 1
A7 = 1 Reset specified bits to zero
A7 = 2 Set highest version number.
- M is the address of a File Descriptor Block, the format of which is as follows:

Bits	0	7 8	15
Word 0	DAD filecode		
1	Userid (characters 1-2)		
2	Userid (characters 3-4)		
3	Userid (characters 5-6)		
4	Userid (characters 7-8)		
5	Filename (char. 1-2)		
6	Filename (char. 3-4)		
7	Filename (char. 5-6)		
8	Filetype		
9	MASK	Version	
Bits	0 3 4	12 13	15

Where: 'DAD Filecode' (word 0, bits 8-15) is in the range /F0 - /FF.
- 'Userid' is up to 8 characters, left justified and space-filled to the right.
- 'Filename' is up to 6 characters, left justified and space-filled.
- 'Filetype' specifies the type of the file (SC = source, etc.).
- 'MASK' (word 9, bits 0-3) is a pattern of bits representing the file attribute flags which it is required to set or reset (see LKM 40, Keep File).
- 'Version' (word 9, bits 13-15) represents a file version number which is to be used as the highest version number, for each Keep File request processed, for the whole library.

Effect

If A7 = 0 The bits corresponding to the MASK are set to 1.
If A7 = 1 the bits corresponding to the MASK are reset to zero.
If A7 = 2 the version number (in word 9, bits 13-15) is treated as the highest version number when processing Keep File Requests for the whole library.

Returned Status

A7 may contain one of the following values when control is returned to the calling program:

- 0 Request performed.
- 1 Invalid address in A8.
- 1 I/O error.
- 3 DAD not assigned.
- 6 File not catalogued in the directory.
- 9 Filecode not assigned to a DAD.
- /A Unknown Userid.
- /1D For background only: the specified Userid is not that of the JOB. Only the owner or the System Manager (JOB USID=SYSTEM) may modify the file attributes.
- /1E The calling program is not the Librarian processor (Background only).
- /1F Invalid order in A7.

LKM 33 - Check and Assign a Filecode

Purpose

To check whether a filecode has been assigned and if possible to re-assign the filecode to a temporary disc file. This request is used by various processors to check whether the temporary output file has been opened or not. For example, the Assembler uses it for checking and assigning temporary object output file (/D5).

Calling Sequence

```
LDKL      A8,N
LKM
DATA      [-]33
[DATA     L]
```

Where: L is the address of a Scheduled Label Routine.
N is the address of an Assign Block having the following layout:

Bits	0	7 8	15
Word 0	1		Filecode
1	DAD Code (/F0-/FF)		
2	Nr and Type of Granules		
3	Not Used		
4	Not Used		
5	File Type.		

- Word 0: - the left byte is set to 1 (assign type)
- the right byte contains the filecode to be checked.
- Word 1 - the filecode of the DAD on which the file will be assigned (if not assigned); if zero, then the DAD of the JOB command will be used (background) or DAD code /F0 or the default DAD (foreground).
- Word 2 - bit 0 indicates the type of granules:
bit 0 = 0, consecutive granules are allocated.
bit 0 = 1, non-consecutive granules are allocated.
- bits 1-15 contain the number of granules required (if this is entered as zero, 1 granule is allocated).
- Word 5 - The file type; one of the standard mnemonics:
SC Source File
OB Object File
LM Load Module
UF User File.

Effect

If the filecode specified in the Assign Block is already assigned, or has been assigned to a temporary disc file and EOF sector has not been written to this file, then the request has no effect. Or the old assignment is deleted and MAS reassigns the filecode according to the parameters given in the assign block.

Returned Status

When control is returned to the requesting program one of the following status codes will be found in A7:

- 0 Assignment done.
- 1 I/O error on disc.
- 2 Dynamic area overflow; entry cannot be created in the filecode table.
- 3 DAD unknown or invalid.
- 5 Disc overflow: requested granules unavailable.
- 9 Invalid assign type (not equal to 1).
- /B Invalid file type.
- /C Invalid filecode (background machine only).
- /E Filecode table overflow.
- 1 Invalid address in A8.

LKM 34 - Check and Write an EOF on a File

Purpose

To ensure that an EOF mark has been written on a file and that the file is closed.

Calling Sequence

```
LDKL      A8,N
LKM
DATA [-]34
[DATA     L]
```

Where - L is the address of a Scheduled Label routine
- N is the address of a word containing the filecode of the relevant temporary file in bits 8-15.

Effect

If an EOF mark has already been written to the file, nothing happens. If an EOF mark has not yet been written, one is written into the next sector of the file.

Returned Status

Upon return to the calling program, A7 will contain one of the following status codes:

```
0      Request performed
1      Disc I/O error
4      Filecode unknown (not assigned)
/C     Invalid filecode (not assigned to a disc file)
/11    Catalogued file
-1     Invalid address in A8.
```

Remarks

This request can be used instead of an LKM 1 (type /22) when it is not known if the file is closed.

LKM 35 - Get a Program's Characteristics

Purpose

To obtain information about the calling program.

Calling Sequence

```
LDKL      A8,N
LKM
DATA      [-]35
[DATA     L]
```

Where: L is the address of a Scheduled Label Routine.
N is the address of a table to which the system will copy the Program Control Table (PCT) and other relevant details such as the Job Parameter Table (background machines) or machine details (foreground machines). This table must be long enough to contain all the information transferred; 64 words in the case of foreground machines and 76 in the case of the background machine.

The format of this table is as follows:

Word	Bit	Contents
0	0-15	Address of next PCT in the chain, or zero.
1-3		Program Name (6 ASCII characters, left justified and padded with spaces).
4		Program start address.
5-7		Reserved.
8	0	1 if system program, otherwise zero.
	1	1 if Exit has been issued but not completed.
	2	1 if Scheduled Labels outstanding.
	3	1 if Read-Only program.
	4	1 if Swappable Program.
	5	1 if Memory-Resident Program.
	6	1 if Middleground Program.
	7	1 if Re-entrant Program.
	8	1 if Program swapped.
	9	1 if Background Program.
9	0	1 if Program not connected.
	1-15	Software Level to which it is connected.
10		Address of ECB on which main program is waiting.
11-13		Addresses of Scheduled Labels to be performed when this task exits.
14		Address of the activation queue for this program.
15		Event count for this program.
16-19		Reserved.
20	0-7	Zero.
	8-15	DAD filecode where the program resides.
21	0	1 if program load module is consecutive.
	1-15	Address of GRANTB sector of the load module; unused if the file has consecutive granules.
22		The number of pages in the region containing this program.
23-24		Reserved.
25		Program Load Address.
26-27		Reserved.
28		Address of the Parameter Block for LKM's 7, 25 and 57.
29		For a re-entrant foreground program, the maximum number of simultaneous activations allowed.
30		Reserved.
31-33		Relevant to disc-resident programs only.

For the foreground machine the table continues as follows:

Word	Bit	Contents
34		Reserved.
35-37		Machine Name (6 ASCII characters, left justified).
38-44		Reserved.
45		Machine Status:
	0	Zero if event occurred.
	1	Zero.
	2	1 if System machine.
	3	1 if Foreground machine.
	4	1 if Background machine.
	5	1 if SM command received and BYE not received.
	6	1 if filecode /EO defines an interactive device (TY, CRT).
	7	1 if catalogued procedure being used.
	12	1 if no middleground program allowed.
	13-15	Return Code used by FCL.
46		MCT address.
47-52		Reserved.

For the background machine the table continues thus:

Word	Bit	Contents
34-37		Userid (8 ASCII characters, left justified and padded with spaces).
38		Maximum execution time (TIME).
39		Maximum number of print lines (PRINT).
40		Maximum number of punched records (PUNCH).
41		Current execution time.
42		Current number of printed lines.
43		Current number of punched records.
44	0-7	Filecode of the DAD containing the current program.
	8-15	JOB DAD filecode.
45		Directory address of the Userid in JOB DAD.
46	0	1 if JOB Userid is SYSTEM.
	2	EOJ of current job received.
	4	1 if /EO is assigned to an interactive device (TY or DY).
	5	1 if filecode /O2 is assigned to the same device as /EO: input commands from /EO do not need to be listed.
	6	1 if /O2 is assigned to the same device as the error recovery file. It is unnecessary to send the error message to both.
	7	1 if the filecode /EO is assigned to the same device as the error recovery file. In interactive mode the processor does not have to reprint the erroneous command on the error recovery file before reading the correction.
	9	1 if pure batch processing mode. Zero if interactive mode; error recovery filecode (word 53) contains the filecode where the correct command can be obtained.
	10	1 if BCP is currently executing.
	11	1 if the LIB processor is currently executing.
	12	1 if a catalogued procedure is being used.
		0 if the command input is /EO.
46	14-15	00 if DUMP=NO) 01 if DUMP=ALL) Set by the BCP. 11 if DUMP=PROG)
47	8-15	Value of the 'ABCD=' parameter on :STP command.
48		Address of the current BCL command in BCP CCT.
49	0	1 if the load module for BCP has consecutive granules.
	1-15	Disc address of the BCP.

50	0	1 if the program to be activated by the BCP has consecutive granules.
	1-15	Disc address of the program to be activated by the BCP.
51	0-7	Abort code of the program just completed.
	8-15	Exit code for the last BCL command.
52	0-7	Filecode used by the BCP to read the current BCL command.
	8-15	'CODE=' parameter value of most recent :STP command.
53	0-7	Filecode defining the ERR device for recovery of command errors.
	8-15	Highest exit code received in this step.
54		System directory address (USERID=SYSTEM, DAD=/F0).
55		No. of pages required to load the BCP.
56		No. of pages required to load the program.
57-75		Save area used to transmit the registers from a background program to the BCP when an abort occurs. Contents are: P-register Program Status Word General registers A1 to A14 Floating-point registers FR1 to FR3.

Returned Status

A7 = 0	Request complete.
A7 = -1	Invalid address in A8.

LKM 36 - Signal the Start of a JOB (BCP Only)

Purpose

This request is used exclusively by the BCP when a BCL :JOB command is received, to request MAS to initialize the JOB tables.

Returned Status

A7 = -3 The requesting program is not the BCP.

LKM 37 - Keep Control on Floating Point Error

Purpose

To avoid the automatic abort of a program if a floating-point error occurs.

Calling Sequence

```
LDKL    A7,A
LDKL    A8,N
LKM
DATA    [-]37
[DATA   L]
```

Where: A is the address within the calling program to which control should be returned by MAS when the error occurs.
- L is the address of a Scheduled Label routine.
- N is the address of a four-word control block in which MAS will return details of the program's status when the error occurred. The format of this block is:

```
word 0   Status
      2   PSW
      4   Instruction Counter
      6   Contents of A8 when the error occurred.
```

Effect

Control is returned to the address specified in A7 in the event of a floating point error occurring.

Returned Status

When control is returned to the calling program, or to any associated scheduled label routine, upon completion of this request A7 will contain one of the following status codes:

```
0      Request recorded.
-1     Invalid address in A7 or A8.
2      System dynamic area overflow; request not recorded.
/16   Floating point hardware option not present on the bare machine.
```

Remarks

- 1) Apart from A8, no other register will have been changed when the specified error exit is taken.
- 2) If a program issues more than one LKM 37, only the last applies.
- 3) The effect of this request is cancelled explicitly by the LKM 38 request, or implicitly when an error occurs, in contrast to LKM 7 (Keep Control on Abort).
- 4) The program must cancel this request before exiting, using the LKM 38 request.

LKM 38 - Cancel 'Keep Control of Floating Point Error'

Calling Sequence

LKM
DATA [-]38
[DATA L]

where L is the address of a Scheduled Label routine.

Effect

The effect of any previously issued LKM 37 is removed; after this request has been recorded by the system, the normal error action will occur when a floating point error is detected.

Returned Status

A7 = 0 Done
A7 = /15 No previous LKM 37 has been issued by this program, or it has already been cancelled.

LKM 39 - Get Machine Options

Purpose

To obtain information about the resources of the bare machine.

Calling Sequence

```
LKDL      A8,M
LKM
DATA      [-]39
[DATA     L]
```

where: L is the address of a Scheduled Label Routine.
M is the address of a Bare Machine Block (BMB) provided by the user program.

Effect

Upon return, the relevant fields in the Bare Machine Block (BMB) will have been filled as follows:

Word	Bit(s)	Contents
0	0-15	No. of characters in the BMB, excluding this word.
1	0	Set to 1 if floating point instructions are used by more than one program or task, i.e. if MAS must save floating point registers. This bit is altered by the SCL commands FON & FOF.
	11	Set to 1 if the machine has the floating point option.
2		No. of free pages in the bare machine.
3		Max. No. of programs in the bare machine.
4-10		Reserved.
11		Device type, e.g. CR, LP etc.
12	0-7	Filecode, if disc device.
	10-15	Device Address.
13		No. of lines per page if device is LP, or Volume Number if disc.
14		Reserved.

Words 11 - 14 are repeated as many times as required to describe all the devices in the machine. The last word is zero.

Returned status

```
A7 = 0   Block filled
A7 = -1  A8 contained an invalid address
A7 = 1   Block too small to contain all information, however it is filled till
         it was full.
```

LKM 40 - Keep File

Purpose

In the background machine, this request is used to prevent a temporary disc file from being released and to enter it into the user directory. In the foreground machine, the purpose is the same; the request may also be used in place of the FCL KPF command to keep a temporary disc file created by another program of the same foreground machine.

Calling Sequence

A8 is loaded with the address of a File Block, the format of which is described below, then:

```
LKM
DATA      [-]40
[DATA     L]
```

where L is the address of a Scheduled Label Routine.

The format of the file block is as follows:

bits	0	7 8	15
Word 0			Filecode
1)		
2)	Userid	
3)		
4)		
5)		
6)	File name	
7)		
8		File Type	
9	A B C D	Zero	
bits	0 1 2 3		

The Filecode (word 0, bits 8-15) must have been assigned to a temporary disc file.

The Userid (of 1-8 ASCII characters) exist in the DAD in which the temporary file has been created. If the first word of the Userid is zero, then for background machines, the Userid of the JOB command is used, while for foreground machines, the first Userid in the DAD or the default DAD-Usid is used.

The File Type (word 8) is a two-character ASCII code such as:

```
UF  User File
SC  Source File
LM  Load Module
```

but note that only the LIB processor may issue an LKM 40 for a load module.

Word 9 contains flag bits (bits 0-3), shown in the table as A, B, C and D; the significance of each bit is:

```
A = 1 if the file is to be unshared.
B = 1 if the file is to be write protected.
C = 1 if the file is to be a system file.
D = 1 if the file is to be an invisible file.
```

Returned Status

```
A7 = 0  File catalogued.
1       I/O Error.
2       System Dynamic Area Overflow.
4       Unknown Filecode.
```

- 9 Filecode not assigned to a disc file.
- /0A Userid unknown.
- /10 Directory overflow: no more entries can be created.
- /11 This filecode is assigned to a catalogued file.
- /12 Invalid File Type.
- /13 Keep file was requested by a background program for a file on a DAD which was not the :JOB DAD
- /14 There was no Userid on the DAD (foreground only)
- /1D All versions of the file to be kept existed already. As no new version could be created, the file with the highest version number had to be deleted, but this can only be done by the owner or by the system user which was not the one that issued this keep file.
- 1 Invalid address in A8.

Remarks

- 1) When the file type is OB (object), the system does not alter the entry in the directory; it assumes that this has already been created. Thus, this file type must not be used in this request by user programs, but only by the LIB Processor.
- 2) The file to be catalogued is implicitly the latest version of the file (version 0). The version numbers of all other entries in the directory, having the same filename and file type, are incremented and any version greater than the maximum, created in this way, is automatically deleted and its granules released.
- 3) If a scheduled label routine is specified with this request, it will be executed before returning to the instruction following the LKM 40. However, all I/O operations will be allowed to terminate before this return is made.
- 4) If a status code of 1 (I/O error) is returned, it is possible that some or all of the version numbers have been incremented and that the old maximum version has been deleted.
- 5) If a status code /10 (Directory full) is returned all existing version of the file are incremented, but there will be no version 0 as the file to be kept could not be inserted in the directory.

LKM 41 - Delete a File

Purpose

To remove a catalogued file from the directory. The granules belonging to the file are released and can be re-used immediately, so the user must not delete a file that is still in use.

Calling Sequence

```
LDKL      A8,B
LKM
DATA      [-]41
[DATA     L]
```

where: L is the address of a scheduled label routine.
- B is the address of a File Block, the layout of which is as follows:

	0	7 8	15
Word 0			DAD Filecode
1)		
2)	Userid	
3)		
4)		
5)		
6)	File Name	
7)		
8		File Type Code	
9		Version Number	

- Userid is 1 to 8 ASCII characters, left justified and padded with spaces. The default value is the Userid of the JOB (background), or the first Userid of the DAD or the default DAD-Usid (foreground). Only the LIB processor may specify the Userid.
- File Name is 1 - 6 ASCII characters, left justified and padded with blanks, and is the name of the file to be deleted.
- Filecode is that of the DAD in which the file is located. If zero, /FO is assumed for the foreground machine, while the DAD of the JOB command is used for the background machine. This entry must be zero for user programs; only the LIB Processor may specify this filecode.
- File type code can be:
 - UF User file
 - SC Source file
 - LM Load Module
 - OB Object File (LIB processor only)
 - EF File managed by TDFM.
- 'Version number' is in the range 0 - 7, and specifies which version of the named file is to be deleted.

Effect

A file with the specified filename, type and version number will be deleted from the directory of the Userid and DAD specified. If these are not specified, the appropriate default values will apply. Any other entries in the directory having the same filename and type, and with a version number greater than the deleted file, will have their version numbers decremented by 1.

Remarks

- 1) It is the user's responsibility to ensure that the file is not in use, since the system will delete it even if the machine still has one or more filecodes assigned to it.
- 2) An implicit wait is caused by LKM 41; control is not returned to the issuing program until all I/O operations on the directory have been completed. Any scheduled label specified in the LKM 41 requests will also be completed before this return is made.
- 3) Only the LIB Processor may specify the DAD filecode and Userid (words 0 - 4 of the file block); for all user programs these entries should be zero.

Returned Status

One of the following status codes will be returned in A7 upon completion of this request:

- 1 Invalid address in A8.
- 0 Operation completed.
- 1 Disc I/O error: the file may or may not have been deleted and other version numbers may have been updated.
- 2 No Userid in the DAD.
- 6 File not found.
- 9 Words 0 - 4 of the file block were non-zero and the calling program is not the LIB Processor.
- /12 Illegal File Type.
- /1C File Type is OB, but the calling program is not the Librarian.
- /1D System flag is set for this file; only the Librarian may delete it.
- /1E DAD filecode set, but the calling program is not the Librarian.

For the LIB Processor and Foreground programs:

- 3 DAD Unknown.
- 9 Filecode is not assigned to a DAD.
- /A Userid unknown.

LKM 42 - Initialise BCP (BCP Only)

Purpose

The request is used by the BCP to initialize the Monitor tables prior to reading a BCL command. It is not available to user programs, and a status code of -3 is returned in A7 if the request is issued by any such program.

LKM 43 - Allocate Permanent Granules (LIB Only)

Purpose

This request is used only by the Librarian, to allocate permanent granules on a DAD (for example, for a Userid Directory). It is not available to user programs, and a status code of /1E is returned in A7 if the request is issued by any such program.

LKM 44 - Release Permanent Granules (LIB Only)

Purpose

The request is used only by the Librarian, to release granules allocated by LKM 43. It is not available to user programs, and a status code of /1E is returned in A7 if the request is issued by any such program.

LKM 45 - Dump Memory

Purpose

Used by any program to print the contents of the program's registers and areas of memory, on the device defined by filecode /02 of the user machine.

Calling Sequence

LKM	
DATA	45
DATA	first address to be dumped
DATA	last address to be dumped

Effect

The effect is as for the SCL/FCL DUM commands; the memory area specified and the contents of all registers are dumped onto the device with filecode /02.

Notes

- 1) There is no returned status; if either address is invalid, or the second is less than the first, then no dump is performed.
- 2) No scheduled label may be used with this request.
- 3) No registers are modified.
- 4) If the postmortem dump option was not selected at system generation time, this LKM will not work.

LKM 46 - Abort the Program

Purpose

Used by the main program or a scheduled label routine to stop execution and to ignore all outstanding scheduled label routines.

Calling Sequence

LKM	
DATA	46

Effect

For a foreground program, unless it has issued an LKM 7 (Keep Control on Abort), a message is sent to filecode /01 of the foreground machine in order to allow the user to request a dump. The program is abandoned, currently active tasks will not exit and no new tasks will be started.

For the background machine, if an LKM 7 has been issued, control is transferred to the user Abort Label and an Abort Code of 06 is placed in the Abort Control Block. It is then the user's responsibility whether or not to abort by issuing an LKM 3 (Exit) with an exit code and post mortem dump flag in A7. If he has not issued an LKM 7, the post mortem dump flag is set and a dump is performed according to the DUMP parameter on the BCL RUN or BCL Processor Call command which activated the program.

If there was a previous BCL :STP command in this Job and the ABCD parameter was specified, the severity code will be set to the specified value. Otherwise the severity code will be set to /7F.

LKM 47 - User Written Request

Purpose

To allow the user to incorporate his own LKM requests into the system without the necessity of recreating the Monitor's LKM table.

Calling Sequence

```
LDKL      A7,N
LKM
DATA      [-]47
[DATA     L]
```

where: L is the address of a Scheduled Label routine.
- N is a one word entry point reference in which:
bits 0-5 contain the relative entry point within the segment.
bits 6-15 contain the segment number, within the D:MASG file of the System DAD, in which the user-written LKM is stored.

These segments are independent load modules which the user incorporates into the MAS segments file D:MASG of the DAD SUPERV. They are executed in the system machine under the program name X:MASG. They are loaded into the System Transient Area and entered at the entry point specified in A7.

Upon entry, the following registers will have been set up:

```
A1 Address of the PCT for X:MASG.
A2 Return address to X:MASG, used to call a subsequent segment.
A5 The User PCT address.
A6 The user Scheduled Label address (if any), otherwise zero.
A14 Zero.
```

Effect

The calling program is suspended and its event counts are incremented. Before returning to the calling program, the user-written LKM must:

- Decrement the event counts.
- Return the status to the calling program.
- Remove the suspension from the calling program by resetting bit 14 of word 7 of the calling program's PCT.
- Start the scheduled label (if any).
- Exit.

See P800 Programmer's Guide 3, Vol. IV: Trouble Shooting Guide, for a complete description of the system tables.

Remarks

- 1) The size of each module is a maximum of 2039 words; if the user-written LKM exceeds this, it must be contained in more than one segment.
- 2) The next segment may be loaded from the previous one by:
 - a) setting A3 to contain the relative entry point in bits 0-5 and the segment number in bits 6-15.
 - b) issuing an ABR A2 instruction (A2 contains the address of X:MASG)
- 3) A segment may contain more than one entry that can be called, indicated by the value given in the first 6 bits of A7. The segment must start with the addresses of each entry in the segment, because X:MASG branches indirect to the first location of a segment, according to the entry number. X:MASG (the disc resident segment loader) checks whether the called segment has already been loaded, if so, it is not loaded again from the segment file. However, this means that a user written LKM must be coded re-usable, that is, each variable must be initiated, because one cannot be sure that the value assigned to the variable when it was declared has not been changed.

Insertion of a segment in the D:MASG file

After having linked the load module containing the new segment, the load module must be moved to the temporary load module file /D6 (or /L):

```
LIB
CDF fnam,LM,ONAM=/L,ODAD=/Fx
LEN
```

Then the D:MASG file has to be assigned and a utility has to be invoked (OVLGEN). This utility inserts the new segment in the D:MASG file:

```
ASG FCOD=/40,FNAM=D:MASG,USID=MASUP,DAD=/FO
RUN PROG=OVLGEN
```

```
RELOC,SEGNR: 300,xx
```

The OVLGEN program asks for the relocation (always 300) and the segment number, i.e. the place in the D:MASG file where the new segment is to be inserted. The segment number must not be an existing one in the MAS system. Because the number of segments used by MAS differs from release to release, no value can be given here. The segment number consists of two hexadecimal numbers without preceding slash (/).

When the MAS system resides on a CDC disc, the D:MASG file must be copied to a DAD called D:MSEG. This is done as follows:

```
ASG FCOD=/40,FNAM=D:MASG,USID=MASUP,DAD=/FO
RUN PROG=COMASG
```

Running the COMASG utility, DAD filecode /F2 must be assigned to a DAD on the disc on which the D:MSEG DAD resides.

Useful MAS modules

Some standard modules reside on the library MASOB in Usid MASGEN on the starter pack with might be of some help, programming an LKM 47.

A:USIN

The module A:USIN initiates the registers for the LKM. It must be declared as external in the LKM routine and called using A15:

```
CF A15,A:USIN
```

The module performs the following actions:

- Load A5 with the PCT (program control table) of the calling program.
- Load A7 and A8 with the values of A7 and A8 of the calling program.
- Load A12 with the MCT (machine control table) of the machine where the calling program resides.
- Load A13 with the CVT (communication vector table) address.
- Load A14 with a stack address. The stack is the module C:STK, available in the MASOB library. Subroutines used by the LKM should be called, using A14.

The MMU of the calling program is loaded.

A:USR1

This module performs the end actions for the LKM. It must be declared as external and called with an absolute branch:

```
ABL A:USR1
```

The module performs the following actions:

- restore of the calling program registers.
- start of the scheduled label (if any).
- unsuspend the calling program. The program was suspended (on LKM) by the LKM interrupt routine I:LKM.
- decrement the event counts (incremented by I:LKM).
- store the MMU of the calling program into its PCT.
- exit from X:MASG (the program under which the user written LKM was running)

C:SVA7

When the user written LKM must return a status to the calling program, it must be loaded into the calling program's A7 register. The LKM may store the status into the external C:SVA7 (1 word), which is loaded into the user's A7 in A:USR1. If no error occurred C:SVA7 should be set to 0.

R:CUAD

This module checks whether an address supplied by the user is in the calling program. It must be declared as external and its calling sequence is:

```
LDKL  A4,retadd
<load A1 with the address to be checked>
ABL   R:CUAD
retadd RF   INVADD
```

R:CUAD expects A4 to be loaded with the return address. When the address to be checked is not in the calling program, R:CUAD returns to the address loaded in A4. When the address is valid, R:CUAD returns to the address in A4 plus 2. Register A1 must contain the address to be checked. An invalid address should cause setting of A7 to -1 (/FFFF) in the calling program.

R:DMAS and R:DMLS

When the LKM needs a buffer, e.g. for reading a disc buffer, it can be reserved in the LKM's segment, but it can also be asked in the System Dynamic Area. To ask a buffer in the System Dynamic Area, use the module R:DMAS:

```
LDKL  A1,length
CF    A15,R:DMAS
```

The length in A1 must be supplied in characters, the address of the obtained buffer is returned by R:DMAS in A1. If no buffer is available, A1 is set to zero. The programmer should take care not to use space outside the buffer, because this will abort the system with error code /09.

When the obtained buffer is not needed anymore, R:DMLS has to be called to release it. This releasing is obligatory as not relasing might cause System Dynamic Area overflow. R:DMLS is called:

```
LD    A1,bufad
CF    A15,R:DMLS
```

The address in A1 must be an address returned by a previous R:DMAS call. If not, the system will abort with error code /09. Both R:DMAS and R:DMLS must be declared as externals.

Core resident LKM's

Described above is the implementation of a disc resident LKM. Also core resident LKM's can be included in the system. In this case, the module T:LKM has to be adapted. This module is delivered on the starter pack. To include a core resident user LKM, an entry in T:LKM must be filled with:

```
DATA  /8000
DATA  usr1km
```

where: /8000 is an indication the a core resident LKM is involved.
usr1km is the address of the user written core resident LKM.

The core resident LKM should return via the Dispatcher (external M:DIS1) with in A4 zero, or, when there is nothing new to dispatch via the external MLRRTN. Both routines, M:DIS1 and MLRRTN expect 8 registers (A1-A8) in the A15 stack, which are stored there by the general LKM interrupt routine. So these registers may be used in the LKM. without saving.

LKM 48 - Assign a Linecode

Purpose

To assign a filecode to a datacommunications device or to another linecode.

Calling Sequence

```
LDK      A7,0
LDKL     A8,A
LKM
DATA     [-]48
[DATA    L]
```

where: L is the address of a scheduled label routine.
A is the address of an assign block, the layout of which is as follows:

bits	0 1	7 8	15
word 0	Assign Type		Linecode to be assigned
1	Device Name or linecode to which it is to be assigned		
2	S	Device Address	
3	Line Number.		

Description of Assign Block

Word 0: 'Linecode' specifies the filecode to be assigned to the line.
'Assign Type' can be:
0: Assign a linecode to a physical device.
3: Assign a linecode to another linecode.

Assign Type 0

Word 1: 'Device Name' is one of the following:

```
S2 = SLCU2
S4 = SLCU4
A4 = ALCU4 or ALCU2
A8 = AMA8A or AMA8C
NO = No Device.
H1 = HLVCU
L6 = LSM16
SA = SALCU
HV = HLVCUZ
SZ = SLCUZ
```

Word 2: 'Device Address' should be given in binary. If S = 1, the filecode is assigned to the first operable device of the requested type encountered in the Monitor tables (i.e. in the order declared at SYSGEN), whatever its address and line number.

Word 3: This location is used only for AMA8 devices, and gives the line number in binary, from 0 to 7.

Assign Type 3

The linecode given in Word zero is made equivalent to that in Word 1. Words 2 and 3 are unused, but must still be reserved within the calling program.

Returned Status

A7 =0 Assignment OK
-1 Invalid address in A8
2 System Machine dynamic area overflow
4 Unknown Device
7 2nd linecode is not assigned (type 3)
9 Invalid assign type
/C Invalid linecode (=0)

Note

Before assigning the linecode, the system first deletes the old assignment if any. If this request fails, therefore, the old assignment will have been deleted.

See P800M Datacommunications User Manuals.

LKM 49 - Delete a Linecode

Purpose

To delete the linecode of a datacommunications device.

Calling Sequence

```
LDK      A7,0
LDKL     A8,P
LKM
DATA     [-]49
[DATA    L]
```

Where: L is the address of a scheduled label routine.
P is the address of a parameter word, thus:

```
bits 0          7 8          15
     |-----|linecode to delete|
```

Returned Status

On return to the calling program, A7 will contain one of the following values:

```
0      Linecode deleted.
-1     Invalid address in A8.
/C     invalid linecode (=0).
```

See P800M Datacommunications User Manuals.

LKM 50 - Submit a Job to the BCP (Foreground Only)

Purpose

To start a job running in the background machine by issuing a request from a user program. The JOB to be submitted must first have been stored by a program in a catalogued DFM file in a DAD also known to the background machine.

Calling Sequence

```
LDK      A7,N
LDKL     A8,JDB
LKM
DATA     [-]50
[DATA    L]
```

where: JDB is the address of a block defining a job description file on disc.
- L is the address of a scheduled label routine.
- N = 0: Submit a Job
 = 1: Get Information about a Job.

Job Description Block Format

Byte	Contents
bits 0	7 8 15
0	DAD filecode (default is /FO)
2)
4	(
6	Userid (default is first userid of the DAD)
8)
/A)
/C	(Filename
/E)
/10	Filetype (must be UF)
/12	Version number (default is 0).

Effect

On receipt of this request, the JOB referred to is retrieved from the file described in the job description block and entered in the job queue for the BATCH machine.

Remarks

This LKM is actioned via the spooling system for the card reader, and so it is essential that spooling is initiated for the card reader before this request is given. The address of the submitted file and the control block address of the DAD, where it resides are transferred to the DAD D:SPCR and from there submitted to the spooling queue in the System Dynamic Area. Because the submitted file itself remains on the user's DAD, the file must not be altered before the submitted JOB is finished.

In the spool DAD, one sector is reserved for submitted job, so the maximum number of jobs in the submit queue is $(D:SPCR \text{ sectorlength} - 10) / 4$.

If in A7 the value 1 has been specified, status always a status unequal to zero is returned.

Returned Status

The following status codes are returned in A7:

- 0 OK
- 1 Invalid Address in A8
- 2 User DAD Not Assigned
- 4 I/O Error on User DAD
- 6 User ID Not Found
- 8 Unknown File
- 10 Card Reader Not Spooled
- 12 I/O Error on DAD D:SPCR
- 14 Job Queue Overflow.
- 16 Job completed or unknown (N = 1).
- 18 Job running (N = 1).
- 20 Job queued (N = 1).

LKM 51 - Spooling Request (BCP Only)

Purpose

This request is used by the BCP to initiate or end spooling activity. It is illegal from a user program.

Calling Sequence

LDK	A7,N
LKM	
DATA	51

where: N is 0 at the start of a spooled JOB
and 1 at the end of a spooled JOB.

Effect

At the start of a JOB the BCP zeroes A7 and initiates the JOB.

At the end of a JOB, A7 is set to 1 and the disc space previously occupied by the spooled JOB is freed. A status code is returned in A7 on completion of this request, as follows:

0	OK, or no spooled device
1	EOB encountered
-3	LKM 51 was not issued by the BCP.

LKM 52 - Send or Receive a Letter

Purpose

To provide a channel of communication between programs in the same or different machines.

Calling Sequence

LDK	A7,F
LDKL	A8,M
LKM	
DATA	[-]52
[DATA	L]

where: L is the address of a scheduled label routine.
F is a function code which may have one of the following values:
 1 Send a letter
 2 Receive a letter.
M is the message area from which the letter is transmitted (function code 1) or into which it will be transferred (function code 2). The format of this area varies with the function code.

Function Code 1 (Send):

Byte	Contents
0	Unused
2	Mailbox Name (2 ASCII characters)
4	Length of letter in characters
6	Unused
8 on	Message to be sent.

Function Code 2 (Receive):

Byte	Contents
0	E
2	Mailbox Name (2 ASCII characters)
4	Length of message block
6	Length of letter received, in characters (set by MAS)
8 on	Message of length specified in byte 6.

- E is the event bit; set when the letter is received.

Note: If the message length exceeds the value specified by the user in byte 4, it will be truncated.

Effect

Function Code 1 (Send)

If the mailbox has not already been created when an LKM 52 request is issued, MAS creates it in the system machine dynamic area. The user does not create the mailbox. As many mailboxes (each known by its unique name of 2 ASCII characters) may be created as the system dynamic area has space for.

As letters are received by a mailbox, they are placed in a queue and requests for these letters are serviced on a first-in first-out basis. Similarly, if a mailbox is empty when requests for letters are received these requests are also queued and serviced in the same way, as and when letters are received.

Function Code 2 (Receive)

The request is serviced when either:

- the request reaches the head of the queue, or
- there is at least 1 letter in the mailbox.

Synchronisation can be effected using a scheduled label routine or by an LKM 2 (Wait for Event) request on the event bit 'E'.

Returned Status

On completion of this request, one of the following status codes is returned in A7:

- 0 Request completed
- 1 Invalid address in A8
- 2 Incorrect length (zero or negative)
- 3 Dynamic area overflow
- 4 Invalid function code in A7.

LKM 53 - Conditional Dump

Purpose

To provide a means of obtaining a dump of the user program area or buffers when the bits set in a mask word match the corresponding bits in a flag word. Both mask and flag word are provided by the user.

Calling Sequence

LKM	
DATA	53
DATA	M
DATA	F
DATA	L
DATA	A
DATA	B

where: M is the address of the user's mask word
F is the address of the user's flag word
L is a 6-character label, printed at the head of the dump
A is the address of the first location to be dumped
B is the address of the last location to be dumped.

Effect

If the MASK word is non-zero, a logical AND operation is performed between it and the flag word. If the result is equal to the MASK, a dump is performed.

Returned Status

If the MASK word is zero, or if any of the addresses are invalid, no dump occurs. No status code is returned.

Remarks

If the postmortem dump option was not selected at system generation time, this LKM will not work.

LKM 54 - Request or Release a Device (Background Only)

Purpose

This depends on the function code loaded into A7 before the request is issued, but the request provides the same facilities as the BCL commands ROI, REQ and REL.

Calling Sequence

```
LDK      A7,F
LDKL     A8,P
LKM
DATA     [-]54
[DATA    L]
```

where: F is the order, in the range 0 - 2, with the following significance:

- 0 Request operator intervention
- 1 Request a device
- 2 Release a device.

P is the address of a parameter block of 38 words, with the following layout:

bits	0	7 8	15
0	Filecode		
1	Device Name (only if A7 = 1)		
2	72 characters of message to be sent to the operator.		

and L is the address of a scheduled label routine.

Effect

This varies with the function code:

Function Code = 0 (Request Operator Intervention):

A message having the following format is sent to the operator's console:

```
'message' dnda, THEN RS, PLEASE
```

dn is the device name
da is the device address.

The program is then suspended awaiting operator action, and is resumed when the operator enters 'RS' on the console.

If the device is a spooled output device, the message is not sent immediately, but is output when the file is unspooled.

For spooled device the message is:

```
'message' dnda, THEN TYPE IN SP DNDA,C
```

Note: This request is only actioned for non-disc devices.

Function Code = 1 (Request a Device):

A message having the following format is output on the operator's console:

```
MOUNT ON 'message' dnda, THEN RS, PLEASE
```

The program is suspended awaiting operator action, until 'RS' is entered on the console. The device is attached to the background program and the filecode is assigned to it.

Note: This request is used for MT and cassette.

Function Code = 2 (Release a device)

The device is detached from the batch program, but the program is not suspended. The following message is sent to the operator's console:

DISMOUNT 'message' dnda, PLEASE

Returned Status

One of the following status codes is returned in A7 upon completion of this request:

- 0 OK
- 1 Invalid address in A8
- 4 Function code in A7 invalid
- 5 Invalid 'dn' specified (function code 1 - not MT or TK)
- 6 No free operable device (function code 1)
- 1 I/O error (spooled device)
- 2 Dynamic area overflow
- 7 Filecode not assigned (function codes 0 or d 2)
- 8 Filecode not assigned to a physical device, or assigned to NO device (function codes 0 or 2)
- 9 Filecode assigned to a disc (functions 0 or 1), or filecode not attached to this program (function 2).

Furthermore the status codes returned by the LKM 23 (assign) apply when order code 1 is specified.

Remarks

The filecode in bits 8-15 of word 0 of the parameter block is the one already assigned to the device (functions 0 and 2), or which is to be assigned by MAS (function code 1).

Trailing blanks in the message are not output and unprintable characters are blanked.

LKM 55 - Semaphore Handling (Foreground Only)

Purpose

To provide a method of synchronising foreground tasks.

Calling Sequence

```
LDK      A7,F
LDKL     A8,S
LKM
DATA     [-]55
[DATA    L]
```

where: F is a function code having one of the following values:

- 0 Declare a semaphore
- 1 Cancel a semaphore
- 2 Decrement a semaphore by 1 (P-operation)
- 3 Increment a semaphore by 1 (V-operation).

S is the address of a one- or two-word block, containing in the first word the 2-character semaphore name. In the case of function code 0 (Declare a Semaphore), a second word is used containing the initial value of the semaphore.

L is the address of a Scheduled Label Routine.

Effect

1) Function Code 0 (Declare a Semaphore):

This request must be made before a semaphore can be used.

The system sets up a semaphore block in the system dynamic area and, in the case of the first semaphore declaration for a machine, enters its address in location /36 of the MCT. Subsequent semaphores are chained from this first one, thus:

MCT	S1	S2	S3	
-----	-----	-----	-----	
	@S2	@S3	0	Chain Address
	S1	S2	S3	Semaphore Name
	+1	0	-2	Semaphore Value
@S1	0	0	@PCT1	PCT chain of suspended
-----	-----	-----	-----	programs.
		PCT1	PCT2	
		-----	-----	
		@PCT2	0	
		-----	-----	

S1, S2 and S3 are three semaphores; S3 has 2 suspended programs queued.

2) Function Code 1 (Cancel a Semaphore):

The semaphore block is removed from the system dynamic area.

3) Function Code 2 (Decrement a Semaphore): The P-operation.

The current value of the semaphore is decremented by 1. If the new value is negative, the calling program will be suspended and put in a queue of all such programs which have issued request type 2 causing the semaphore value to go negative. The absolute value of the semaphore indicates the number of programs suspended.

4) Function Code 3 (Increment a Semaphore): The V-operation.

The current value of the semaphore is incremented by 1. If the new value is negative or zero (i.e. there were tasks suspended and queued), the first task in the queue will be reactivated.

The Use of Semaphores

The most usual application of semaphores is in cases of competition for resources where deadlock or over-allocation may occur. For example, if two tasks both request a resource held by the other and are suspended waiting for it to become free, neither can free the resource which it holds and which is required by the other. The result is deadlock. Similarly if two independent tasks allocate a resource according to the following algorithm:

If resource A is free,
then assign it.

and they are both competing for the same resource, they may both coincidentally get a TRUE response to the first statement and proceed to attempt to allocate the resource.

These difficulties can be overcome by using P- and V- operations on a semaphore. Suppose, in the second example above, that resource A is allocated a semaphore of initial value 1, then a safe algorithm would be:

P(s) (i.e. decrement the semaphore).
If resource A is free,
then assign it.
V(s) (i.e. increment the semaphore).

Any competing task, attempting to perform a P-operation on S after the first task has performed one, will be suspended until the V-operation is performed.

If a number of independent foreground tasks have a pool of buffers available, the Get Buffer and Release Buffer operations can be effectively synchronised by declaring a semaphore of initial value equal to the number of buffers available.

If the sequence adopted by each task is:

P(s)
Get Buffer
Process
Release Buffer
V(s)

then as soon as the semaphore becomes negative following a P(s) operation, the issuing task will be suspended until another task releases a buffer and performs a V-operation upon the semaphore.

Returned Status

The following status codes are returned in A7 upon return to the issuing program:

Function Code 0 (Declare a Semaphore):

- 0 Completed
- 1 Invalid address in A8
- 1 Invalid function code in A7
- 2 Semaphore already declared.
- 3 Dynamic area overflow.

Function Code 1 (Cancel a Semaphore):

- 0 Completed
- 1 Invalid address in A8
- 1 Invalid Function code
- 5 Unknown Semaphore
- 4 Semaphore cancelled, but there was a program suspended.

Function Code 2 (Decrement a Semaphore - P Operation):

- 0 Completed
- 1 Invalid address in A8
- 1 Invalid function code in A7
- 5 Unknown semaphore
- 6 Semaphore has been cancelled (the program is not suspended).

Function Code 3 (Increment a Semaphore - V Operation):

- 0 Completed
- 1 Invalid address in A8.
- 1 Invalid function code in A7.
- 5 Unknown semaphore.

LKM 56 - Request/Release Pages (Foreground Only)

Purpose

To allow programs within any one foreground machine to acquire and share named working areas.

Calling Sequence

LDK	A7,C
LDKL	A8,B
LKM	
DATA	[-]56
[DATA	L]

where: L is the address of a scheduled label routine.
- B is the address of a parameter block.
- C is the request order code, which may have one of the following values:

0 or /8000:	request pages
1:	release pages
2:	connect pages to the program.

Request Pages (A7 = 0 or /8000)

If request order 0 is used, an immediate return to the program is made with a status code in A7. If /8000 is used and there are insufficient pages to satisfy the request, then the program will be suspended until the required number of pages becomes available.

The parameter block addressed by A8 has the following structure:

0	Page group area name (2 ASCII characters)
2	Relative address of page area
4	Page size (/1000)
6	Number of pages to be allocated
8-38	16-word copy of MMU registers.

The name of the group of pages to be allocated must be unique within a machine.

The relative address is the logical address within the program to which the pages will be connected. It must be a page boundary (0, /1000, /2000 up to /F000). Only the 4 leftmost bits are significant.

The page size is fixed at /1000 (i.e. 2K words), but the default value is zero.

The last 16 words of the table are filled by MAS and will contain, on return to the calling program, a map of the adjusted MMU registers, showing the new page structure.

Remarks

- If the user sets the 'relative address' entry in the table to -1, the system will search for the first unused MMU register, starting with zero. If no free register exists, the pages will be connected to virtual address zero.
- Allocated pages are assigned to consecutive virtual addresses in ascending order, modulo 32K.
- If request order /8000 is used and the program is suspended due to insufficient free pages, an entry is created in the system area to preserve the coordinates of the pages to be allocated. All suspended programs are re-activated as sufficient pages become free.

Returned Status

One of the following status codes will be returned in A7 on completion of this request:

- 0 Request Successful
- 1 Invalid Address
- 1 Invalid Request Order
- 2 Already Declared
- 3 Dynamic Area Overflow
- 4 Insufficient Free Pages
- 5 Invalid Number of Pages (more than 15).

Release Pages (A7 = 1)

For this request the parameter block addressed by A8 consists of one word, containing the name of the pages block to be released.

Note: after this request has been processed, any attempt by this or any other program to access the released pages will result in a page fault error.

Returned Status

The following code is returned in A7 on completion of this request:

- 0 Satisfactory completion
- 1 Invalid Address
- 1 Invalid Order
- 6 Unknown page group name.

Connect Pages to a Program (A7 = 2)

The parameter block addressed by A8 has the following format:

- 0 Page Group Name
- 1 Relative Address
- 2 Check
- 3 Number of Pages to be Connected.

where:

- Page Group Name is the name of a group of pages, which must already have been allocated by Request Pages (order 0 or /8000).
- Relative Address is the address within the requesting program, to which the pages group is to be connected.
- Check may have the values 0 or 1. If 0, the Relative Address in Word 1 must be the same as the Request Pages request; if 1, the addresses may differ. If Check is 1 and Relative Address is -1, the system searches for the first free MMU register and connects from there. The actual address is returned in Word 1. Connection is refused if not enough MMU registers are free.
- Number of Pages to be Allocated must be the same as in the Request Pages request.

These pages must have been previously allocated by another program (using request order 0 or /8000).

Returned Status

Upon return to the calling program, A7 will contain one of the following status codes:

- 0 Satisfactory completion
- 1 Invalid address in A8
- 1 Invalid order
- 4 Insufficient free pages
- 5 Invalid number of pages (differs from that given at request time).
- 6 Unknown page group name.
- 7 Invalid relative address (differs from that given at request time).

LKM 57 - Connect/Disconnect a Secondary Load Module

Purpose

This request enables the calling program to modify its MMU tables to include or exclude the named secondary load module. While the module is included it may be accessed by the calling program.

Calling Sequence

```
LDKL      A8,A
LKM
DATA      [-]57
[DATA     L]
```

where:

- L is the address of a scheduled label routine.
- A is the address of a six word control block with the following lay-out:

	0	15
word 0	SLM name	
1	to be connected	
2	load address	
3	SLM name	
4	to be disconnected	
5	not used	

One can connect (word 0#0) or disconnect (word 3#0) a secondary load module. When both word 0 and 3 are unequal to zero, first the secondary load module specified in words 3 and 4 is disconnected and then the secondary load module in words 0 and 1 is connected.

For the connection of a secondary load module, word 2 must contain its load address. This load address must be equal to the address specified during Link Edit of the secondary load module.

Effect

- a) Connect (word 0#0)
The MMU tables of the calling program are modified to allow it to access the secondary load module area. The calling program must disconnect the secondary load module after use. Connection is refused if the MMU pages to which the module must be connected are not free.
- b) Disconnect (word 3#0)
The MMU tables of the calling program are modified to free the pages allocated to a previously connected secondary load module. The pages may then be re-allocated to the same or a different secondary load module.

Remark

It is dependent on the type of the program, which pages can be used to load a secondary load module. Background programs are loaded from page 0 onwards, disc resident foreground programs are also loaded from page 0 onwards, but from page 15 downwards, pages for the CMA are reserved. Memory resident foreground programs are loaded just before the CMA, which again occupies the highest pages

Returned Status

On return to the calling program, A7 may have one of the following values:

- 0 successful.
- 1 invalid address in A8.
- 1 secondary load module not loaded.
- 2 the secondary load module refers to the ascendant (i.e. the calling program), while the ascendant is a memory resident foreground program. Referring to the ascendant is only allowed, when the ascendant starts at page 0.
- 3 dynamic area overflow.
- 4 page(s) already allocated.
- 5 page(s) already disconnected.

LKM 58 - Wait Multiple (Foreground Only)

Purpose

To enable the user to restart a program when a specified proportion of a total number of events have occurred.

Calling Sequence

```
LDK      A7,0
LDKL     A8,B
LKM
DATA     [-]58
[DATA    L]
```

where: L is the address of a scheduled label routine.
B is the address of the multiple event control block, the layout of which is as follows:

bits	0	1	15
word 0	E	Event Word	
1	Total Number of Events (N)		
2	Number of Events Awaited (P)		
3	Reserved		
4	Reserved		
5	Address of 1st Event		
6	Address of 2nd Event		
.	-		
.	-		
.	-		
.	Address of Nth Event		
.	P words, reserved for MAS to		
.	insert the addresses of the		
.	P ECB's		

Effect

The program is placed in a 'multiple wait state'. When P events out of a specified N events have occurred, it will be restarted.

Remarks

- The range of P can be expressed as $1 \leq P \leq N$.
- This LKM can only be used in the program's main sequence.
- A scheduled label routine, if specified, will be started when at least P events have occurred.
- Bit 0 of the multiple wait block will not be set by the system.
- P and N have a maximum value of 64. This limit is imposed by the system in order to reduce the danger of dynamic area overflow, while not imposing too great a restriction on the user.
- When P events already have been occurred, control is immediately returned to the calling program.

Returned Status

A7 may have one of the following values on return to the calling program:

- 0 Normal termination
- 1 Invalid address in A8
- 2 The request is issued by a scheduled label
- 3 Too many elementary events ($N > 64$)
- 4 $P > N$
- 5 Dynamic area overflow.

LKM 60 - Load/Delete a Secondary Load Module

Purpose

This request loads a secondary load module into memory, or deletes it when it is no longer required.

Calling Sequence

LDK	A7,N
LDKL	A8,A
LKM	
DATA	[-]60
[DATA	L]

where:

L is the address of a scheduled label routine.
N is the order:
0 load the secondary load module
1 delete the secondary load module.
A is the address of a 5-word (for load) or 3-word (for delete) block, with the format:

0-1	Secondary Load Module Name
2	Not used
3	DAD Filecode
4	Type of Page

The secondary load module name is four ASCII characters, left-justified and space-filled; the module is catalogued in the first Userid of the DAD specified in Word 3.

The DAD filecode specifies the DAD from which the secondary load module will be loaded. It need not be supplied if the module is to be deleted.

The Type of Page is two ASCII characters, 'R' or 'W':

'R': the module is read-only;
'W': the module may be modified during execution, so it may be accessed by only one primary load module at any one time.

The type of page need not be specified if the secondary load module is to be deleted.

Effect

a) Load (A7 = 0)

The user must load a secondary module using this request, before connecting it to a primary load module using LKM 57. Loading is not required if the secondary load module is already in memory and has not been deleted.

b) Delete (A7 = 1)

The secondary load module is deleted from memory, if it is not connected to any primary load module.

Returned Status

a) from Load:

A7 = -1 invalid address in A8
0 successful
1 invalid order in A7
2 DAD unknown
3 DAD sector length greater than 512 characters
4 secondary load module already loaded
5 unknown program
6 I/O error
7 secondary load module is segmented
8 too few free pages
9 secondary load module too long
/C dynamic area overflow
/D type of page invalid.

b) from Delete:

A7 = -1 invalid address in A8
0 successful
2 secondary load module already deleted
3 secondary load module still connected to a program.

LKM 62 - Check if DAD assigned

Purpose

To check whether a DAD to be used is assigned in some machine. This LKM is used by the Librarian processor to check whether a DAD to be deleted is still in use.

Calling sequence

```
LDKL      A8,N
LKM
DATA      [-]62
[DATA     L]
```

Where: L is the address of a scheduled label routine.
- N is the address of a nine word block, the layout of which is as follows:

bits	0	15
word 0	DAD filecode or zero	
1)	
2)	DAD name
3)	
4	Disc filecode	
5	Not used	
6)	
7)	Machine name
8)	

Where: DAD filecode is specified when a check is needed to see whether the specified DAD is assigned to this given filecode. If zero is specified MAS checks whether the DAD is assigned to any (DAD) filecode.

If zero is specified, on return this word contains the filecode to which the DAD is assigned.

- DAD name is the name of the DAD to be checked.
- Disc filecode is the filecode of the disc where the DAD resides.
- Machine name is the name of the machine where the DAD is assigned.

Effect

The specified DAD is searched in the filecode tables of all machines. If it is found, a status in A7 is returned and the name of the machine is stored in words 6-8 of the control block.

Remark

If the DAD is assigned in more than one machine, or assigned to more than one filecode, only the first occurrence of the DAD is recorded in the control block.

Returned status

```
A7 = 0      No (not that) filecode is assigned to the DAD in any machine.
A7 = -1     Control block out of the program.
A7 = -2     Word 4 does not contain a disc filecode (/CO-/CF)
A7 = -3     Word 0 does not contain a DAD filecode (/FO-/FF) nor zero.
```

```
A7 = 1      The DAD is assigned in the Batch machine.
A7 = 2      The DAD is assigned in a Foreground machine.
A7 = 4      The DAD is assigned in the System machine.
A7 = 8      The DAD is assigned in the machine from which the LKM is issued.
```

In the latter four cases, the machine name is stored in words 6-8 of the control block and the filecode to which the DAD is assigned is stored in word zero.

LKM 63 - Set Date and Clock

Purpose

To change the date and time as recorded currently in the system.

Calling sequence

```
LDKL    A8,M
LKM
DATA    [-]63
[DATA   L]
```

Where: L is the address of a scheduled label routine.
- M is a control block containing the new Date and Time, with the following layout:

bits	0	15
word 0	Day	
1	Month	
2	Year	
3	Hour	
4	Minute	
5	Second	

The date values must be specified in ASCII format, the time values in binary format. A check is made, whether the date and time are valid.

Remark

Great care should be taken, using this LKM. It affects the whole system and changing date and time can disturb the activation of programs connected to timers in any machine.

Returned status

```
A7 = 0    Date and time changed.
A7 = -1   Control block out of the program.
A7 = 1    Not existing date.
A7 = 2    Not existing time.
A7 = 3    Value of date does not contain decimal ASCII characters.
```

LKM 70 - Interface FCL and Middleground processor

Purpose

To read the last command given to the FCL task in the current machine. This LKM is used by some standard processors like NOD, INC, etc. to read the FCL command with which the processor is called. The length of the command is given in word 0, the command itself in the next words (max. 36) of an area, of which the address is given in A8.

Returned status

A7 = 0 Command moved.
A7 = -1 Area out of the program.

LKM 71 - Assign a filecode to a DAD

Purpose

To assign or re-assign a DAD filecode (/FO-/FF) to a DAD, residing on a disc, specified by its filecode (/CO-/CF) or its packnumber.

Calling sequence

```
LDKL      A8,M
LKM
DATA      [-]71
[DATA     L]
```

Where: L is the address of a scheduled label routine.
- M is the address of a seven word block with the following layout:

bits	0	15
word 0	DAD filecode	
1	Pack-	
2	number	
3	Disc filecode	
4)	
5) DAD name	
6)	

Where: DAD filecode is the filecode to be assigned to the DAD. If it already exists, the old assignment is deleted.

- Packnumber is the volume number of the disc, where the DAD resides in 4 ASCII hexadecimal characters.
- Disc filecode is the filecode of the disc where the DAD resides. If both packnumber and disc filecode are specified, only the disc filecode is examined.
- DAD name is the name of the DAD to be assigned.

Effect

The DAD is assigned to the specified filecode. If the filecode was already assigned, the old assigned is deleted, however if the LKM is used in the Batch machine, it only affects the current JOB. After End-Of-Job the old assignment is reset.

Remarks

- The DAD of the current Job (in Background) or DAD /FO (in Foreground and Background) may not be re-assigned.
- On return both disc filecode and packnumber are stored in the control block.
- No files may have been assigned to a DAD to be re-assigned, nor programs may have been loaded from it.

Returned status

On return A7 contains one of the following status:

- A7 = 0 Assignment done.
- A7 = -1 Control block out of the program.
- A7 = 1 I/O error, reading the VTOC, or the Bittab of the DAD.
- A7 = 2 System dynamic area overflow.
- A7 = 3 DAD does not exist on the specified disc.
- A7 = 6 Packnumber does not consist of 4 hexadecimal characters, or no disc in the system has the specified packnumber.
- A7 = 7 Disc filecode is not /CO-/CF or DAD filecode is not /FO-/FF.
- A7 = 9 Disc filecode not assigned, or not assigned to a disc.
- A7 = /C Filecode to be assigned exists already but has not been assigned to a DAD.

A7 =/D Try to re-assign the Job DAD or DAD /FO.
A7 =/E Too many filecodes assigned.
A7 =/F DAD cannot be re-assigned, some programs are loaded from it.
A7=/10 Disc filecode is assigned to a Data Floppy disc.
A7=/8001 Filecode cannot be re-assigned because some files ave been
assigned to it.

LKM 73 - Set default DAD and Usid (Foreground only)

Purpose

To change the default DAD and Userid from DAD /FO, first Userid to a user defined DAD and Userid.

Calling sequence

```
LDKL      A8,M
LKM
DATA      [-]73
[DATA     L]
```

Where: L is the address of a scheduled label routine.
M is the address of a control block with the following layout:

bits	0	15
word 0	DAD filecode	
1)	
2)	Userid
3)	
4)	

Where: DAD filecode is the filecode of the DAD where the new default Userid resides.

When Userid is not specified, the first Userid in the DAD is taken.

The name of the first Userid is returned in the control block.

Returned status

```
A7 = 0   Done.
A7 = -1  Control block out of the program.
A7 = 1   DAD filecode is invalid (not /FO-/FF) or not assigned to a DAD.
A7 = 2   The DAD does not contain a Userid.
A7 = 3   The specified Userid does not exist in the DAD.
```

LKM 74 Read the default DAD and Userid (Foreground only)

Calling sequence

LDKL A8,M
LKM
DATA [-]74
[DATA L]

Where L is the address of a scheduled label routine.
 M is the address of a five word area.

Effect

On return of this LKM, the area M contains the DAD filecode and the Userid, which are default in the current Foreground machine. On return, the layout is the same as for LKM 73.

Returned status

A7 = 0 Done.
A7 = -1 Area out of the program.

LKM 75 Dump system area

Purpose

To move a part of the system area to the user program.

Calling sequence

```
LDK      A7,0
LDKL     A8,M
LKM
DATA     [-]75
[DATA    L]
```

Where: L is the address of a scheduled label routine.
M is the address of a control block with the following layout:

bits	0	15
word 0	ECBRB	
1	ECBLEN	
2	ECBEB1	
3	ECBEB2	

where: ECBRB is the receiving buffer address.
ECBLEN is the length of the area to be moved.
ECBEB1 is the most significant bits of the emitting buffer address. To be used in future, must be 0.
ECBEB2 is the least significant bits of the emitting buffer address.

Effect

The system area, addressed by ECBEB2 and ECBLEN is used to the user area, from address ECBRB on.

Remarks

Entering this LKM, A7 and ECBEB1 must be 0.

Returned status

```
A7 = 0   Area is moved.
A7 ==-2  On entering this LKM, A7#0.
A7 ==-1  Control block out of the program.
A7 = 1   Address in ECBRB is not in the user program.
A7 = 2   ECBEB1 is not 0.
A7 = 4   Error in length. The length is negative, or ECBRB+ECBLEN is
          greater than /FFFF or ECBEB2+ECBLEN is greater than /FFFF.
```

LKM 76 Abort another program in the same machine (Foreground only)

Calling sequence

LDKL A8,M
LKM
DATA [-]76
[DATA L]

Where: M points to a three word block,containing the name of the program
 to be aborted.
 L is the address of a scheduled label routine.

Effect

The specified program is aborted with abort code /16.

Returned status

A7 = 0 Program aborted.
A7 =-1 Program name block out of the program.
A7 =-2 Program not known in this Foreground machine

LKM 77 Accept an attention key

Purpose

To interrupt output on a device connected to the AMA8 or the ASCU4Z.

Calling sequence

LDKL	A8,M
LKM	
DATA	[-]77
[DATA	L]

Where L is the address of a scheduled label routine.
M is a one word ECB, containing in bits 8-15 a filecode assigned to a physical device for which an attention key request can be issued.

Effect

The attention key request is put in a queue. On receipt of an attention key (the ESC key) the event bit of the ECB is set and the scheduled label (if any) is started. Attention keys are serviced on LIFO (Last-In-First-Out) basis. An attention key request can be cancelled via LKM 78.

Returned status

A7 = 0	Attention key request accepted
A7 = -1	ECB out of the program.
A7 = 1	Filecode not assigned, assigned to NO device, or not assigned to a physical device.
A7 = 2	System dynamic area overflow.
A7 = 3	The filecode is not assigned to a device for which an attention key request is possible.

LKM 78 Cancel the last attention key request

Purpose

To undo the last issued LKM 77

Calling sequence

LDKL	A8,M
LKM	
DATA	78

Where M is a one word ECB containing a filecode assigned to an attention key device.

Effect

The last issued attention key request is removed from the queue, the event of the attention key request is set and its scheduled label (if any) is started.

Returned status

A7 = 0	Request done.
A7 = -1	ECB out of the program.
A7 = 1	Filecode not assigned, assigned to NO device or not assigned to a physical device.
A7 = 3	Filecode not assigned to an attention key device.
A7 = 4	There was no attention key request recorded by this program for this device.

LKM 79 Close a spool file (Foreground only)

Purpose

To close a spooled file so that it can be unspooled.

Calling sequence

LDKL	A8,M
LKM	
DATA	[-]79
[DATA	L]

Where L is the address of a scheduled label routine.
M is a one word ECB, containing a filecode assigned to a spooled file in bits 8-15. The first bit of the control word is an event bit which will be set on completion of the LKM.

Effect

An EOF record is written to the spooled file. The file is put in the unspool queue and is output when it is on the top of this queue.

Returned status

A7 = 0	spooled file closed.
A7 = -1	control word out of the program.
A7 = 1	LKM issued by a Background program.
A7 = 2	no spool command (operator command SP) given.
A7 = 3	filecode not assigned or not assigned to a spooled device.
A7 = 4	I/O error, writing the EOF record on the spool file.
A7 = 5	I/O error, accessing the Spool DAD.

