

PART 4

FOREGROUND MACHINES

GENERAL

A foreground machine is required only if non-batch user programs are executed. A non-batch program may be loosely defined as a program whose main input arrives as isolated records whose order and volume is unpredictable (unlike batch programs, whose main input is a batched file of records), and where the program cannot delay dealing with the next record until it has finished with the current one. Records may be signals sent by a machine connected to the computer; each signal must be processed (or stored on an I/O device or in memory) immediately it arrives. In other words, the input controls the program with foreground machines, whereas the program controls the input with background machines.

MEMORY ALLOCATION

After IPL, the System Machine will occupy physical memory locations from 0 up to some multiple of 2K words, the exact size depending on how the system was generated. The maximum size of the system machine is 32K words (16 pages) for non Extended mode systems. For systems containing Extended Mode, the maximum length of the system machine is 54K words (27 pages).

The remaining memory locations in the bare machine may be used to create user machines. One background machine and/or one or more foreground machines may be created. These user machines exist only until the next IPL, or until an SCL KIM command is given.

When all user machines have been defined, any memory locations which are still unused are allocated for use as a common dynamic loading area for all machines. The background machine is only required if Standard Processors or user batch programs are to be executed. It may be memory- or disc-resident (as specified by the user on the SCL DCB command). If memory-resident, a number of pages (specified by the user in the SCL DCB command) are permanently reserved for the background machine and cannot be used by foreground machines. If disc-resident, no pages are permanently reserved for the background machine. Instead, the background machine is regarded as a disc-resident task which competes with disc-resident tasks of the foreground machines for memory allocation in the common dynamic loading area shared by all user machines. Memory will be allocated to it according to its software level (specified on the SCL DCB command, or defaulted to 2 less than the Idle Task) in relation to the competing foreground tasks.

MULTI-USER FACILITIES

A foreground machine has only one set of filecodes. It may thus be connected to up to 16 DADs. All the disc files and programs used by a foreground machine must be contained in the Directory for the first user for each of these DADs.

Secondly, if several foreground tasks were run in one foreground machine, all sharing devices, it would be difficult to determine which input was from which task and which output was for which task. Using separate foreground machines enables each application to ensure (via LKM 14 Attach Device and LKM 15 Detach Device) that devices such as card readers and line printers cannot be shared by unrelated tasks.

Thirdly, for unrelated foreground tasks, it is easier to operate if each set of related tasks is performed in its own separate foreground machine, each machine being controlled by a separate user and with its own programs and files.

Fourthly, errors in one foreground machine will not affect activities in the other foreground machines.

MEMORY ORGANISATION

Segments

Each foreground machine consists of a number of segments. The main purpose of this is to allow foreground machines to make optimum use of the MMU (Memory Management Unit) of the P800 range of computers. Foreground machines can thus exceed 16 pages.

Communications Area

Segment 0 is called the Communications Area. It consists of two parts called the Public Library Area and the Dynamic Area.

Public Library Area

The Public Library area contains zero or more user programs.

Dynamic Area

The Dynamic Area contains all the buffers obtained by LKM 4 (get buffer) for programs running in this foreground machine (in any segment).

Segment Organization

Other segments each contain zero or more user programs. Segment 0 is logically part of each of these segments, and programs in these segments may address programs and work areas in Segment 0 by using virtual addresses of 32K downwards (e.g. if Segment 0 was defined as 2 pages on the SCL CMA command, each segment can address it using virtual addresses from 28K to 32K). Segments (other than 0) can have a maximum size of:-

16 - N PAGES

where N is the number of pages in Segment 0. The size of each segment is defined with SCL SEG commands. Note that the virtual addresses of Segment 0 are always from 32K downwards, even in a segment whose size is less than the maximum allowed. Foreground programs are loaded into Foreground Machine segments (including Segment 0) by FCL REP or FCL LOD commands. Programs within one segment may address each other directly. Programs in different segments can only communicate with each other through Segment 0.

Dynamic Loading Area

Not all foreground machine programs are loaded into segments. For programs which do not require to be in memory all the time, the user may declare them to be disc-resident (with FCL SWP or FCL RON commands), or they may be middle-ground programs. MAS will load these programs in and out of memory as necessary. The Common Dynamic Loading Area will be used for this purpose.

The Common Dynamic Loading Area is shared by all foreground machines and by a disc-resident background machine. It should not be confused with the Dynamic Area in Segment 0 of each foreground machine, or the System Dynamic Area in the system machine.

Example

Suppose the system manager defines a 4 page communications area, of which the public library area is 2048 characters:

CMA 4,2048

Also, the systems manager defines 3 segments of 9, 12 and 3 pages respectively:

SEG 1,9
SEG 2,12
SEG 3,3

Now suppose the foreground user declares the following programs, whose size in Kwords is shown in brackets:

LOD 1,PGM1,/F2 (5)
LOD 1,PGM2,/F4 (12)
LOD 2,PGM3,/F1 (4)
LOD 2,PGM4,/F5 (4)
LOD 2,PGM5,/F1 (2)
LOD 3,PGM6,/F2 (5)
SWP PGM7,/F5 (9)
SWP PGM8,/F2 (23)

Then MAS will construct five addressing areas (i.e. 5 MMU tables). Each area is constructed page by page, starting with page 15 and then 14 etc. The pages do not need to be consecutive in memory. The first three areas are allocated pages reserved exclusively for this foreground machine. The fourth and fifth areas are allocated pages from the common dynamic loading area shared by all machines (except for pages 12-15 of these areas, which are in Segment 0 of this foreground machine).

The first area will comprise Segment 1 and Segment 0.

PAGE	WORDS	CONTENTS	
3	0-1K	Unused	1st half page 3
3	1K-2K	PGM2	2nd half page 3
4-8		PGM2	
9	0-1K	PGM2	1st half page 9
9	1K-2K	PGM1	2nd half page 9
10-11		PGM1	
12-14		Dynamic Area	
15	0-1K	Dynamic Area	1st half page 15
15	1K-2K	Public Library Area	2nd half page 15

The second area will comprise Segment 2 and Segment 0.

PAGE	WORDS	CONTENTS	
0-6		Unused	
7		PGM5	
8-9		PGM4	
10-11		PGM3	
12-15		The same pages as pages 12-15 of the first area.	

The third area will comprise Segment 3 and Segment 0.

PAGE	WORDS	CONTENTS
9	0-1K	Unused
9	1K-2K	PGM6
10-11		PGM6
12-15		The same pages as pages 12-15 of the first area.

The fourth area will comprise:

PAGE	WORDS	CONTENTS
0-3		PGM7
4	0-1K	PGM7
4	1K-2K	Unused
12-15		The same pages as pages 12-15 of the first area.

The fifth area will comprise:

PAGE	WORDS	CONTENTS
0-10		PGM8
11	0-1K	PGM8
11	1K-2K	Unused
12-15		The same pages as pages 12-15 of the first area.

Note: the fourth and fifth area is an addressing area for swappable programs. These addressing areas are only filled when the swappable program is loaded. If the program is not active or swapped out, only the pages 12-15 (CMA) are recorded in the addressing area.

FOREGROUND TASKS

Initialising the Task Environment

Once a foreground machine has been defined (as explained in a later section), all the programs which may be executed in it must be defined, unless they are Middle-ground programs.

Each program which is to be executed in a segment of a foreground machine (i.e. a program which is to be permanently resident in memory reserved exclusively for one foreground machine) is called a Memory-Resident foreground program. Memory-resident foreground programs may be re-entrant or non-re-entrant.

Foreground programs which are executed in the Common Dynamic Loading Area shared by all user machines (foreground and disc-resident background) are called Disc-

Resident Foreground programs. They are loaded in and out of the Common Dynamic Loading Area by the system as required. There are three types of Disc-Resident foreground program: Middle-ground, Read-only and Swappable programs. Read-only programs may be reentrant too. Every foreground machine program must be defined before it can be executed, unless it is a Middle-ground program.

For a Memory-Resident foreground program, the definition involves loading it into a foreground machine segment, indicating whether or not it is re-entrant, and specifying the software level (priority) of the program.

For a Disc-resident foreground program, the definition involves specifying whether it is a Read-only or a Swappable foreground program, and specifying the software level of the program.

For a Middle-ground program, there is no need to declare it, nor to connect it to a software level.

Task Scheduling

Once the programs have been defined, they may be used to perform tasks. A foreground task is started either by Activating a program (by an FCL RUN or FCL ACT command given by the foreground machine user, or by an LKM 12 issued by another task in the same foreground machine), or it may be started automatically by the system at a set time, or regularly after a specified period of time has expired. These times are specified by the foreground machine user. For example, a program may be activated at 9.06 a.m., or the user may specify that it is to be activated every 2.6 seconds. This is done either by an LKM 10 issued by another task in the same foreground machine, or by an FCL CNT command from the foreground machine user.

A foreground task ends when the foreground program, which started when the task started, Exits (by LKM 3).

The first task must be started by an FCL RUN, FCL ACT or an FCL CNT command from the foreground machine user. The program activated by this command can then activate other foreground machine programs (by LKM 12) or make them active when timers or clocks reach certain specified values (by LKM 10). These programs can in turn activate, or connect to timers or clocks, other foreground machine programs, and thus create new foreground tasks. Meanwhile the user may have started other tasks by FCL ACT, RUN or CNT commands.

Activation Queues

A foreground machine task is one execution of a foreground machine program. It is possible for the same foreground program to be required for several tasks simultaneously. For example, program 1 may Activate program 7 which then starts. It is then interrupted by program 4 because the latter has a higher priority (a lower software level), and program 4 then activates program 7. MAS maintains an Activation Queue for each non re-entrant foreground program. Every time a foreground machine program is activated (or made active by timers or clocks), a new entry is created in its Activation Queue. Whenever this program becomes the highest priority executable task in the bare machine (i.e. in all machines - system, foreground and background), the first task in its Activation Queue is resumed. When this task Exits, its entry is removed from the program's Activation Queue. Multiple activations of the same foreground machine program are thus processed one at a time on a FIFO (first-in-first-out) basis.

Re-entrant foreground programs (which are always memory-resident when declared with FCL REP commands and always disc-resident when declared with the RON command) do not have activation queues. A re-entrant program can perform several tasks conjointly.

Activation with Scheduled Labels

If a foreground machine program uses an LKM 12 to Activate another program, the system branches to the scheduled label address (if present) when the Activated program Exits.

If the Activating program has lower priority than the Activated program, then the NSI (next sequential instruction) to the LKM 12 will be executed if:

The Activated program enters a Pause or Wait.

The Activating program becomes the highest priority executable task in the bare machine.

If the Activating program has higher priority (lower software level) than the Activated program, then the NSI to the LKM 12 in the Activating program will be executed before the Activated task starts. The Activated task will only be capable of starting when the Activating task enters a Pause, Abort or Wait.

If the Activating program activates itself, then a new entry is created in the Activation Queue for the program and the Activated task can only start when the Activating task Exits. This can never occur if the Activating task is waiting for the scheduled label to complete. A deadlock will thus occur.

Task Synchronisation

If one foreground task wishes to wait for another task in the same foreground machine to complete, then the following procedure may be used:

When program 1 activates program 2 by LKM 12, A8 is set to point to an event byte.

When program 1 wishes to wait for program 2 to Exit, it ensures that A8 is pointing at the event byte, and issues an LKM 2.

When the task starts (i.e. when program 2 is the highest priority executable program in the bare machine and the task activated by program 1 is at the top of the Activation Queue for program 2), program 2 receives the value of A8 set by program 1 when it issued the LKM 12.

Just before program 2 Exits, it should ensure A8 has the same value as on entry, and issue LKM 18 to set the event bit.

Note that if an activated task aborts, bit 15 of A8 will be set to 1 and the scheduled label of the activating task will be started.

FOREGROUND PROGRAMS

Memory-Resident

Memory-resident foreground programs are loaded by the Foreground user into foreground machine segments, with FCL REP or FCL LOD commands. They may be loaded into Segment 0 (the Communications Area) or into any other segment defined by an SCL SEG command, submitted by the System Manager when the Foreground machine was defined.

Memory-resident foreground machine programs may be contained in any DAD assigned to the foreground machine (by SCL FCD commands given by the System manager when the Foreground machine was defined), but they must be contained in the Directory for the first userid of any such DAD. The DAD containing each program is specified on the REP or LOD commands.

Several memory-resident foreground programs may be loaded into one segment. Programs in one segment can address each other directly. All programs can address Segment 0. Programs in different segments of the same foreground machine can thus communicate with each other through Segment 0. They can also communicate via Segment 0 with disc-resident programs of the same foreground machine and executing in the Common Dynamic Loading Area. Once loaded into a segment, memory-resident foreground programs remain in that segment until the next IPL, unless an FCL KIL command is given.

If a memory-resident foreground program is an Overlay program (i.e. it has been created by the Linkage Editor to contain a root segment and one or more overlay segments), the root segment will contain LKM 27 (inserted by LKE) which will, in the case of disc-resident overlays, load the overlay segments from the Load Module file into the same foreground machine segment as the root segment. The root segment will be permanently resident in memory. The overlay segments will be loaded one at a time, as required, into a fixed area in the foreground machine segment.

In the case of programs containing memory-resident secondary load modules, these are loaded into memory before run time by means of LOD commands, and the root segment will contain LKM 57 requests to connect these modules into the program path by modifying the MMU Table. The technique of using memory-resident secondary load modules is discussed in the Software Processors Manual in the section on the Linkage Editor.

Each memory-resident non-re-entrant foreground program (declared by FCL LOD) has a software level (priority) which is associated with every task performed by it.

The software level of memory-resident re-entrant foreground programs (declared by FCL REP) is the minimum software level which may be used. Each task will be attached to the lowest unconnected software level which is not less than this minimum. A program cannot be activated, nor connected to a timer or clock, until it has been connected to a software level. Memory-resident foreground programs are connected to a software level, either by an FCL CNL command from the foreground machine user, or by an LKM 20 issued by another program in the same foreground machine.

Disc-Resident

Disc-resident foreground programs are defined by the foreground machine user (unless they are Middle-ground programs) by FCL RON or FCL SWP commands. They may be contained in any DAD assigned to the foreground machine (by SCL FCD commands given by the System manager when the foreground machine was defined), but must be contained in the Directory for the first userid of any such DAD. Each RON and SWP command specifies which DAD contains the program. Several disc-resident programs can be specified for one foreground machine, but they can only address each other or address memory-resident foreground programs via Segment 0. Disc-resident foreground programs can activate other foreground programs (either disc- or memory-resident) for the same foreground machine. This they do using LKM 12, permitting them to pass parameters to the activated program via A3 or A8 and to receive parameters from it via A8 when it has completed (providing the activating program has a scheduled label or waits for the activated program).

All disc-resident foreground programs may address Segment 0 of the same foreground machine, by using virtual addresses of 32K downwards.

Disc-resident foreground machine programs are loaded and unloaded as necessary in the Common Dynamic Loading Area, also used for the Background machine if the SCL DCB command did not define a memory-resident background machine.

Overlay disc-resident foreground programs partially control their own loading into the Dynamic Loading Area. The root segment is loaded and unloaded as for any normal disc-resident foreground program. The overlay segments are loaded by the root segment as required (by LKM 27 inserted in the root segment by the LKE Standard Processor). Overlay disc-resident programs must be either Middle-ground or defined by SWP and not RON.

Each disc-resident foreground program must have a priority (software level), which is associated with every task performed by the program. A program cannot be activated, nor connected to a timer or clock, until it has been given a priority (either by an FCL CNL command from the foreground machine user, or by an LKM 20 issued by another program in the same foreground machine). Middle-ground programs are connected automatically by MAS, when they are activated, to the lowest unconnected software level.

Disc-resident foreground programs may be Middleground, Swappable or Read-only.

Read-Only Programs

These are declared by FCL RON commands given by the Foreground user. Its core-image is written to (disc temporary file in) the DAD D:CI. This DAD has no user, and is automatically connected by MAS to filecode /F1 of the systemmachine.

If a higher priority program requires to use memory in the Common Dynamic Loading Area occupied by a Read-only foreground program, the latter is overwritten without being written out to the core-image file sectors created by the RON command. When the program is re-loaded, the registers are restored to their values when the program was over-written. Processing resumes at the instruction being executed when the program was over-written. Read-only foreground programs cannot, therefore, modify their core-image, and may contain only instructions and constants. All variable-content memory (memory whose contents vary for different tasks, or at different times during the same task - such as I/O buffers, accumulators and switches) must be contained in memory in the Dynamic Area of Segment 0. LKMs 4 and 5 are used to obtain this memory and to free it. If a Read-only, Reentrant or Read-only-reentrant program tries to modify its core-image, it will be aborted with error 3 (protection) because the pages wherein the program is loaded are set to write protect.

Swappable Programs

These are disc-resident foreground programs which are liable to modify their core-image during the execution of a task. All disc-resident foreground programs which are also Overlay programs must be either Middle-ground or Swappable. Swappable programs are defined by FCL SWP commands from the foreground machine user.

Swappable programs may contain I/O buffers, work areas, switches and other variable-content memory. They do not have to allocate these in Segment 0 with LKM 4, though they may do this if required - because another program in the same foreground machine needs to be able to address them directly. Swappable programs cause the total throughput of the bare machine to be less than would be the case if they were Read-only programs. This is because the swapping operation involves more disc accesses than is required by Read-only programs.

Middle-ground Programs

Middle-ground programs are identical to swappable programs, except that they are not declared by SWP commands, and are not connected to a software level by the user but by MAS itself. Middle-ground programs are located by searching (when they are activated) all DAD's known to this foreground machine in the order /FO to /FF. Only the directories for the first userid of each DAD are examined.

When a middleground program is activated, the system in fact executed first a SWP and a CNL (connect level) command before the activating command is executed. On exit of a middleground program, the system executes a KIL program command, except when the program is aborted. On abortion, the middleground program is not killed and a new activation results in execution of the same core image.

Swapping

When a swappable program is declared (by an FCL SWP command), MAS writes the program from the relevant DAD to the Core Image file (contained in the DAD D:CI - system machine filecode /F1), and reserves an area for the program in the same DAD to contain the Swap Image.

When a task which is to be performed by a swappable program is started (i.e. when there are enough free pages in memory, or a program has been swapped out to free enough pages, and this is the first activation), the swappable program is loaded into the Common Dynamic Loading Area from its Core Image created by the SWP command. This load is done even if the program is already in the Common Dynamic Loading Area (because it has just finished the task that was previously at the top of its Activation Queue). The reason is that, unlike Read-only programs, Swappable programs may modify their instructions and switches while executing a task. To ensure that these are reset to their initial values at the start of the next task performed by the program, the program must be reloaded from its initial core image. However, a special loading mechanism is available to prevent this time-consuming reloading. Of course, using this special loading mechanism the swappable programs must be re-usable.

Whenever the dynamic loading area is occupied, either by a program declared with a SWP command or by a Middle-ground program which has exceeded its minimum core-residence time, the program cannot simply be over-written (as is the case for a Read-only program). The swappable program may have modified its core-image, while executing its current task. MAS writes any pages that have been modified since the program was last swapped into the Swap Image area reserved by the SWP command for this program.

When the swappable program again becomes eligible to run and it is not a new task to be performed (i.e. the interrupted task is to be resumed), it is reloaded by loading all the pages in its swap image, and loading any other pages from its core image.

The core image file in the DAD D:CI (which is defined by system machine filecode /F1) thus contains:

- The initial core image of all read-only and swappable programs for all foreground machines. These are copied from Load Module libraries when the foreground machine users give the FCL commands RON or SWP. The DADs containing the Load Module libraries (in the first userid) containing the programs are specified on the RON and SWP commands.
- A swap image of all swappable programs (declared by SWP, or middle-ground programs) for all foreground machines. Each swappable program has one swap image area, which contains either garbage (if the program has never been swapped out), or the memory-image of the modified pages when it was last swapped-out. The background machine will be one swappable program if the SCL DCB command did not specify a memory-resident background machine.

The physical swapping operation may be delayed because the swap event count is non-zero. Each swappable program has a swap event count. Whenever it issues an LKM which will modify its allocated memory (for example an LKM 1 to Read a record into a buffer embedded in the program), MAS increments the program's swap event count. If a program is to be swapped-out, the physical operation only starts when its swap event count reaches 0.

If the physical swapping of a program is delayed by one of the above causes, no other program in the same machine can be swapped out until the first swapping is finished.

MIDDLEGROUND PROCESSING

To perform Batch processing with more than one user at a time, Middleground processing has been developed.

Middleground processing enables the user to run Batch programs (e.g. Software processors) in a Foreground machine.

Use of Middleground

To run in Middleground, the most suitable way is to declare a Foreground machine with filecodes /EO and /O1 assigned to an interactive device like TY or DY.

The filecode /FO should be assigned to a user's DAD, and the system DAD should be assigned to a higher /Fx filecode. The advantage of this is that all processors, and some LKMs, use as default value for assigning files etc. the first userid on DAD /FO, which is the user's userid in this case.

It will be clear that, for each foreground machine which is meant to be used for Middleground processing, the filecodes /EO, /O1 and /FO, assigned to interactive devices or user DADs respectively, should be independent of other machines.

Middleground programs can be activated via the FCL commands RUN and ACT, and via LKM 12 (Activate). As Middleground simulates Batch processing, the RUN command will be the normal method of activation.

Middleground programs need not be declared (via SWP, LOD, REP or RON commands), nor need they be connected to a level.

On activation, MAS searches for the PCT of the specified program. If not found, the program is assumed to be Middleground. The program is searched for in the directory of the first userid of each DAD declared in the machine, starting with DAD /FO. When found, a PCT is created and the program is loaded and connected to the highest free software level. Then it is activated and handled as a swappable program.

When the program exits, it is disconnected from its level and removed from memory, and its PCT is deleted.

Printing

With printing, care should be taken. The activities of other users must be allowed for, otherwise outputs will be mixed.

Two possibilities exist of avoiding mixed print output:

1. Assign filecode /02 to the interactive DY/TY device to which the filecodes /EO and /O1 have been assigned, or to another device which can be used exclusively by one user. If a user wants to print something, he must warn the other users not to use the lineprinter.
2. A more convenient way is to spool the lineprinter. Outputs of one process are gathered and printed together afterwards, so that mixing of outputs is impossible.

Omission

When /FO is assigned to a user's DAD in an LKE step, the SLIB parameter should be set to some value (NO or a library name), because on the user's DAD the SYSLIB object library (default for SLIB parameter) will usually not be present.

Example

Declare a Foreground machine:

```
FCL:DCF FORG
  D:CMA 1,200
  D:FCD /CO
  D:FCD /C1
  D:FCD /FO,/C1,USER
  D:FCD /F6,/CO,SUPERV
  D:FCD 1,DY18
  D:FCD 2,DY18
  D:FCD /EO,DY18
  D:DEN
FCL:BYE FORG          start the Foreground machine
FCL:RUN LIB
LIB:CSF INAM=1,ONAM=10,ODAD=/FO
CSF:      IDENT      TEST
CSF: --- input of a program
CSF: ---
CSF::EOF
LIB:KPF FCOD=10,FNAM=TEST,TYPE=SC
LIB:LEN
FCL:RUN ASM
ASM:OPT PROG=TEST
FCL:RUN LKE
LKE:OPT CATL=TEST,MAP=YES,SLIB=NO
FCL:RUN TEST
FCL:BYE              end of the session.
```

Note that, running a middleground program, from MAS rel 8, it is sufficient to give the name of the program, without RUN just as in the background machine.

DEFINING A FOREGROUND MACHINE

Sequence of Operations

Only the System Manager may define foreground machines. SCL commands entered from the device assigned to the system machine filecode /EO are used to define foreground machines. Once a foreground machine is defined, it may be started by an SM operator command entered from the device assigned to the system machine filecode /EF. The foreground machine will then start to read FCL commands from the device assigned to its filecode /EO.

Defining a foreground machine involves specifying:

- The name of the machine (to be used subsequently in operator and SCL commands).
- The number of memory-resident segments and the size of each segment.
- The devices and DADs to be used by this foreground machine and the filecodes which identify them.
- If necessary, the System manager may also override certain SYSGEN default values which control the tables MAS constructs for a new foreground machine. These values are the number of scheduled labels usable in user programs in this foreground machine, the maximum number of filecodes which may be assigned by this foreground machine, and the maximum number of blocking buffers which may be used by this foreground machine.

When all SCL commands necessary to define a foreground machine have been entered, the SCL DEN command should be given. A full description of the SCL commands is given in Part 2: The System Machine, Chapter 7. When a foreground machine has been defined and started, the FCL commands specify:

- The memory-resident programs to be loaded (from which DAD and into which segment).
- The disc-resident programs which may be used (and in which DAD they reside). Middle-ground programs are located automatically by MAS.
- The priority (software level) to be associated with each program. All tasks performed by one program (unless it is re-entrant) have the same software level. Middle-ground programs are connected automatically by MAS.
- The programs to be activated immediately, and those to be activated according to timers or clocks.

Once programs have been activated, further FCL commands allow the foreground user to control his application.

Example of a foreground machine declaration:

```
DCF  MNAME,2
CMA  4,2048
SEG  1,9
SEG  2,12
FCD  /C0
FCD  /C1
FCD  /C2
FCD  /O1,TY10
FCD  /O2,LP07
FCD  /EO,TY10
FCD  /F6,/C0,SUPERV
FCD  /FO,/C1,DAD1
FCD  /F1,/C2,DAD2
DEN
```

FCL COMMANDS

When a foreground machine has been started, using the SM operator command, FCL commands will be read from the device that was assigned to filecode /E0 during the machine definition phase.

FCL commands are used to complete the initialization of the foreground machine, load application programs, assign priorities, connect programs to clocks and timers, and activate programs.

The last FCL command should be BYE, which causes the FCL control program to exit.

If an FCL command is rejected, the file codes /01 and /02 are used (in the same way as for the system machine) to log and submit corrections.

Error Messages

Two general error messages can occur while processing FCL commands; these are:

```
INPUT COMMAND I/O ERROR
COMMAND UNKNOWN
```

Further error messages, specific to the command causing them, are described under the relevant command.

Corrections

An input command error results in the following messages being output on the console (filecode /01):

- The erroneous command;
- An error message;
- FCL: (prompting for an FCL command).

The system then waits for the correct message to be input via the device attached to filecode /E0.

Filecodes

The following filecodes must be assigned by SCL FCD commands before starting the foreground machine, since the FCL control program uses them:

- /01 FCL correction device; this must be a console or VDU, and is used to input corrections to rejected FCL commands. Subsequent corrections will be read from the device assigned to filecode /E0.
- /02 Print device; used for outputting FCL error messages. This may be assigned to the same device as filecode /01 or /E0.
- /E0 The device from which FCL commands will be read when the foreground machine is started. This may be assigned to the same device as /01 or /02.

/FO-/FF These are assigned to all the DADs used by this foreground machine. All programs and files for an application are contained in the first Userid Directory within these DADs. At least one DAD must therefore be specified, since otherwise no user programs could be executed in this foreground machine.

If catalogued procedures are used, filecode /FO must define the DAD containing the file F:PROC.

Foreground programs should not use filecodes /EC, /ED or /EE, since these are used for processing foreground catalogued procedures.

Parameters

Positional parameters are used, each of which is limited to 6 ASCII characters (truncated if more). Hexadecimal numbers must be preceded by a slash (/) character.

Declaring Foreground Programs

All user programs (apart from middleground programs) must be loaded or declared using one of the following commands:

- LOD - Load a memory resident program
- REP - Load a re-entrant program
- RON - Declare a read-only program
- SWP - Declare a swappable program.

Default values

References to any disc file in a foreground machine apply to the directory of the first Userid of the a specified DAD (if the DAD filecode is not specified, the default is DAD /FO). However, from MAS release 8.50, these default can be overruled by the FCL JOB command or by LKM 73 (Set default DAD and Userid).

A description of these follows.

FORMATLOD s,p,d[,l]

- s is the segment number into which p is to be loaded, expressed as an integer in the range 0 to the maximum number of segments defined for this foreground machine when the 'SCL DCF' command was entered.
- p is the name of the program which is to be located in DAD d and loaded into segment s.
- d is the DAD containing program p as a load-module file, located via the Directory for the first Userid.
- l is an optional field, specifying the maximum number of scheduled labels which may be started during any task of this program. If this field is not entered, the default value is that specified on the 'SCL LAB' command or, failing this, the SYSGEN default value. Since MAS must reserve space in the System Dynamic Area in order to preserve data as each scheduled label routine is started, it is in the user's interest to keep this reserved area to a minimum. If the number of possible simultaneous scheduled label routines is less than the default value, this field should be specified.

Error Messages

THIS PROG DOES NOT EXIST
PROG. ALREADY LOADED
SEGMENT NO. TOO HIGH
LACK OF ROOM IN THE SEGMENT
DAD FILE CODE ERROR
DYN AREA OVERFLOW
TOO MANY SCHEDULED LABELS
PARAMETER ERROR

Purpose

To load a secondary load module into memory; this is a module output by the Linkage Editor, which can be referenced by one or more other modules called primary modules.

FORMAT

LSM n,fc[,{R | W}]

n is the 4 ASCII character name of the secondary module, output by the Linkage Editor and catalogued in the first userid of the DAD fc. fc specifies the DAD filecode from which the secondary load module will be loaded. Otherwise the DAD invoked is the corresponding one declared in the current user foreground machine under the same filecode.

R is entered if the secondary load module is to be considered read-only.

W is entered if the secondary load module is to be considered as modifiable at execution time. In this case it can be used by only one primary module.

The default value is 'R'.

Remarks

Before referencing the secondary load module, the primary load module must have been linked to it, i.e. the pages of the secondary load module must have been connected by means of LKM 57. This LKM 57 coding is generated by the LKE, linking the primary load module with its secondary load modules.

Error Messages

One of the following error messages will be output if this command is rejected:

UNKNOWN SECONDARY LOAD MODULE
SEC. LOAD MOD. IS SEGMENTED
SEC. LOAD MOD. ALREADY LOADED
DAD FILECODE ERROR
DYN. AREA OVERFLOW
PARAMETER ERROR
DAD SECTOR TOO LONG
I/O ERROR
TOO FEW FREE PAGES
SEC. LOAD MOD. TOO LONG
SLM NAME EXCEEDS 4 CHARACTERS

Format

REP n,s,p,d[,l]

- n is the maximum number of simultaneous activations allowed for program p.
- s is the segment number into which this re-entrant program is to be loaded. It is the number given in the 'SCL SEG' command which defined this segment of the machine. Zero means that the program is to be loaded into the Public Library Area.
- p is the program name, consisting of from 1-6 ASCII characters, defining a Load Module.
- d is the DAD filecode, as specified on the 'SCL FCD' command which defined the DAD of this foreground machine. The re-entrant program p is entered in the directory of the first Userid.
- l specifies the maximum number of scheduled labels which may be simultaneously active.

Remarks

A re-entrant foreground program does not have an activation queue; if the program is already performing a task when a second activation is requested, this second activation occurs immediately. Each new activation causes a new program control table (PCT) to be created.

Once the user has connected a re-entrant program to a level (by means of the FCL CNL command or LKM 20, for example), each new activation will be connected to the next highest unconnected level (i.e. the next lowest priority).

A re-entrant foreground program cannot be specified on a PS, RS or KI operator command. If it is the subject of an AB operator command, all currently active tasks of this program are aborted.

Error Messages

If the command is rejected one of the following error messages will be output:

THIS PROG. DOES NOT EXIST
PROG. ALREADY LOADED
SEGMENT NUMBER TOO HIGH
LACK OF ROOM IN THE SEGMENT
DAD FILE CODE ERROR
DYN. AREA OVERFLOW
TOO MANY SCHEDULED LABELS
PARAMETER ERROR

Purpose

To declare a disc-resident Read-only Program. This command allows the system to allocate disc space (in the core image disc area) to receive the initial core image of the read-only program, and to set up a Program Control Table for the program.

Format

RON p,d[,l][,t][,R,a]

p is the program name.

d is the filecode defining the DAD whose first Directory contains the program.

l is the maximum number of scheduled label routines which will be activated simultaneously.

t is the minimum core-residence time in seconds (default value is in the CVT).

R indicates that the program is to be considered as Read-only-Reentrant.

a is the maximum number of simultaneous activations for the program.

Remarks

Read-only programs cannot be overlaid, must not alter their core image during execution and must contain only instructions and constants. All variables must be allocated in the dynamic area of Segment 0, using LKM 4, and these areas must be freed, before a task ends, using the LKM 5 request.

The maximum length of a read-only program (excluding its dynamic areas) is 16 pages minus the number of pages in Segment 0.

If the Read-only program has been declared as reentrant, the pages occupied by the program are only freed when all activations for the program have been finished.

Error Messages

If this command is rejected, one of the following error messages will be output on the device assigned to file code /01:

PARAM ERROR

PARAM MISSING

USER F.C. PROG NOT ASGN

PROGRAM ALREADY DEFINED

PROGRAM DOES NOT EXIST

DISK I/O ERROR ON READ L.M.

SYSTEM DYNAMIC AREA OVERFLOW

D:CI F.C. /F1 NOT ASGN

SYST D:CI FILE OVERFLOW

DISK I/O ERROR

PROGRAM TOO LONG

PROG SEG. NOT READ ONLY

the read-only program is an overlay program.

Purpose

To declare a swappable program. This command causes the system to copy a program from a user DAD to the system core-image file, D:CI, and to reserve another area in this file to contain the swap image of this program. This second area is only used if the program is swapped out and has modified one or more of its core-image pages in memory at the time the swap-out occurs. The user must declare all the swappable programs used in the application before they are activated, otherwise they will be considered to be Middleground programs.

Format

SWP p,d[,l][,t]

p is the program name.

d is the filecode of the DAD, in whose first directory the program is entered.

l is the maximum number of scheduled label routines which may run simultaneously.

t is the minimum core-residence time in seconds (default value is in the CVT).

Remarks

Swappable programs may be overlay programs.

The maximum size of a swappable program (excluding any buffers assigned in the dynamic area of Segment 0) is 16 minus the number of pages in Segment 0.

Error Messages

PARAM ERROR
PARAM MISSING
USER F.C. PROG NOT ASGN
PROGRAM ALREADY DEFINED
PROGRAM DOES NOT EXIST
DISK I.O. ERROR ON READ L.M.
SYSTEM DYNAMIC AREA OVERFLOW
D:CI F.C /F1 NOT ASGN
DISK I/O ERROR
PROGRAM TOO LONG

Starting the User Program

Once all the user programs have been declared, the application to be run in the foreground machine can be started.

Until the 'BYE' command is given deleting the FCL control program, the user application, once started, executes in conjunction with the control program. Since this usually has a lower priority than any user task, then, with the exception of the 'FCL RUN' command (which causes suspension of the FCL task until the user task has exited), the user task will be executed and the FCL control program will only be able to read commands when the user programs are suspended, have exited, or have aborted.

The following commands are relevant to the starting up of user applications:

- ACT Activate a program
- CNL Connect a program to a level
- CNT Connect a program to a timer or clock
- RUN Run a program.

Purpose

To start a foreground task, i.e. one execution of a foreground program.

Format

ACT p[,a][,b]

p is the program name.

a is a value to be placed in register A3 before the program is activated. It is an optional parameter; the default value is 1.

b is a value to be placed in A4 before the program is started. The default value is 1.

Note: Before activation, the program must have been connected to a level. In the case of middleground programs this connection is made automatically by MAS on receiving the 'ACT' command. For all programs, except reentrant ones, an activation queue per program is created, when the program to be Activated is already active.

Error Messages

PROGRAM UNKNOWN

PROGRAM NOT CONNECTED

LACK OF ROOM IN DYN. AREA

PARAMETER ERROR

TOO MANY REENTRANT ACTIVATIONS

NO SOFTWARE LEVEL FOR REENTRANT PROGRAM

PROGR ABORTED (EVC#0)

the program to be activated has been aborted and there is still an outstanding event. No further activations can be made before this event is ready.

Purpose

To connect a program to a software level prior to activation.

Format

CNL p,n

- p is the program name, and must have been specified on a previous REP, LOD, RON or SWP command for this foreground machine.
n is a number defining the software level of program p.

Remarks

The lower the software level to which a program is connected, the higher is its priority. Whenever an interrupt occurs, the task which is currently executing is suspended and MAS starts or resumes processing the highest priority activated program which is in an executable state.

Each program in the bare machine must have a different software level.

The software level of the Idle Task (specified at SYSGEN) must be higher (lower priority) than that of all other tasks. Only one program may be connected to one level. The maximum level is 239.

Error Messages

PROGRAM ALREADY CONNECTED
SOFTWARE LEVEL ALREADY USED
PROGRAM NOT YET DECLARED
PARAMETER ERROR
LEVEL NUMBER TOO HIGH

Purpose

To specify a start-time for an initial execution of a program, and optionally the time interval for periodic re-activation of the same program.

Format 1 Connect to a Timer:

CNT p,d,l,r,n

- p is the program name. No middleground.
d is the length of one time period, and is an integer in the range 1-5 having the following meaning:
 1 = minutes
 2 = seconds
 3 = 100 msec
 4 = 20 msec
 5 = a non-standard timer (only if the bare machine has the clock option).
l is an indication that the program must be connected to a timer. The parameter is a 1 (one).
r is the reactivation parameter. Once the program has been activated (which will occur for the first time after 'n' periods of length 'd'), it will be re-activated every 'r' periods of length 'd'. The value of r can range from 0-2047, but if it is zero, the program will only be activated once.
n is an integer in the range 0-32767, specifying the number of periods of length 'd' which are to elapse before the first activation.

Format 2 Connect to a Clock:

CNT p,d,2,r,t

- p is the program name. No middleground.
d is the length of one time period (as for Format 1).
r is the reactivation parameter, and is an integer in the range 0-127. Once the program has been activated (at time 't'), it will be reactivated every 'r' periods of length 'd'. If 'r' is zero, the program will only be activated once.
t is the time at which the program is to be activated, entered as:
 HH,MM,SS

where HH is an integer in the range 0-23 (hours);
 MM is an integer in the range 00-59 (minutes);
 SS is an integer in the range 00-59 (seconds).

Remarks

The program 'p' must have been connected to a level before this command is given. A program may be connected to several timers and clock values simultaneously. The system of activation and reactivation is administered via the MAS table T:RTC, which contains seven timer addresses (one for each of the six time periods and one for the clock). Each address in this table points to the first block in a chain of blocks for this timer.

Each of these blocks contains details of the times at which a program is to be activated, and points to the PCT identifying the program. Several blocks in one or more of these chains may point to the same PCT.

These blocks are created by the CNT command or by LKM 10, LKM 22 or LKM 28, and are removed by the DST command or by LKM's 11, 21 and 29.

The non-standard timer is a special hardware option which counts periods of time whose length is specified by the user.

If a non-re-entrant program connected to a timer has not exited before the time for the next activation has expired, the next (and subsequent) activations are placed on an activation queue until the exit occurs. Only one activation can be processed at a time; the queue is processed on a first-in first-out basis. The length of time taken to execute one activation of a program varies from activation to activation, depending on the software level of the program and the nature of the other active tasks occupying the bare machine.

If a program connected to a timer is aborted, the current task never exits, the tasks on its activation queue are ignored and no new activation queue entries will be made. The program can only be reactivated by an 'FCL RUN' command, or the abort condition must be removed by means of the 'FCL RAB' command.

Error Messages

PROGRAM NAME IS NOT ALPHA
PROGRAM NOT FOUND
WRONG TIMER NUMBER
WRONG CNT COMMAND
WRONG FORMAT
WRONG PULSE RATE
PULSE RATE TOO LARGE
WRONG NUMBER OF CYCLES
YOU MUST BUY THE CLOCK OPTION TO USE TIMER 5
WRONG DELAY VALUE
TOO MANY PARAMETERS
WRONG TIME
1 DAY = 24 HOURS
1 HOUR = 60 MINUTES
1 MINUTE = 60 SECONDS
PROGRAM ALREADY CONNECTED TO THIS TIMER
NOT ALLOWED FOR MIDDLEGROUND PROGRAMS

Purpose

Performs the same function as the ACT command, the difference being that no further FCL commands for this foreground machine will be accepted until the activation has been completed and the program has exited or been aborted.

Format

RUN p[,a3[,a4]]

p program name declared in a LOD, REP, SWP or RON command for this machine. The program must have been connected to a software level, or be a middleground program.

a3 value to be placed in A3 on program entry. Default value: 1.

a4 value to be placed in A4 on program entry. Default value: 2.

Error Messages

PROGRAM UNKNOWN
PROGRAM NOT CONNECTED
PARAMETER ERROR
PROGRAM ABORTED (EVC#0)

Controlling the User Application

The following FCL commands can be used to control the running of user programs once they have been declared and started by means of the commands previously described. Most of the functions performed by the following commands can also be performed by means of LKM requests.

Format

ABT p

p is the name of the program to be aborted.

Remarks

The 'FCL ABT' command provides the foreground user with similar facilities to the 'AB' operator command (see Part 1, Chapter 5).

Abort is ineffective as long as the program is in a wait state.

Error Messages

PARAM ERROR
PROG NAME MISSING
PROG INACTIVE
PROG ALREADY ABORTED

Purpose

To assign filecodes to:

- a) another filecode
- b) a physical non-disc device
- c) a disc temporary file
- d) a disc catalogued file
- e) a DAD.

Format 1 - to another filecode:

ASG fcl,fc2

Format 2 - to a physical non-disc device:

ASG fcl,dn[da[ln]]

Format 3 - to a disc temporary file:

ASG fcl,DDdc,ft[,granules[,NC]]

Format 4 - to a disc catalogued file:

ASG fcl,DDdc,ft,fn

Format 5 - to a DAD:

ASG fcl,dk,dan

fcl is the filecode to be assigned and for which an entry is to be created in the filecode table for this foreground machine.
fc2 is a filecode which has already been assigned in this foreground machine, and to which filecode fcl is to be assigned.

Note - the filecode fc2 must not have been assigned to another filecode.
 - if the filecode fc2 is subsequently reassigned, filecode fcl is also automatically reassigned.

dn is a non-disc device name, as given in Chapter 3.
da is the device address. If omitted, the first device encountered in the MAS device table having device name 'dn' will be assigned.
dc is the DAD filecode, in the range /FO - /FF, specified in the SCL FCD command which defined the DAD in which the file to be created or read will or does exist in the Directory of the first userid for this foreground machine.
ft is the file type, and is one of the following:-
 UF - User File
 SC - Source File
 OB - Object Module File
 LM - Load Module File
 EF - Extended Disc File.

ln is a linenumber that can be specified for devices connected to the AMA8. This linenumber may also be added to the AMA8 device address in the da parameter.

granules is the number of granules to be reserved. The default is one.

Note: If the file is to be created by a Direct Write (LKM 1) request, the number of granules must be allocated at Assign time. If a Direct Write is attempted to a sector of a file beyond the number reserved by the FCL ASG command, the write will be rejected with a status of /10.

NC is a code indicating that the granules need not be consecutive, and can only be entered if a number of granules are allocated.

fn is the filename. This file, of type 'ft' and version 0, must exist in the directory of the first Userid of the DAD defined by DAD code 'dc'.

dk is the disc filecode (/CO - /CF).

dan is the DAD name.

Remarks

A new entry will be created in the filecode table for this foreground machine, providing the number of entries does not exceed the value specified on the 'SCL MFC' command when this foreground machine was defined (or the SYSGEN value, if no MFC command was given).

The ASG command is an alternative method of assigning foreground machine filecodes to those of using the 'SCL FCD' command or the LKM 23 request.

In the assign for a TDFM file, the filename specified is the descriptor file. The other files, the subfiles, containing the keys and the data are assigned automatically by TDFM. These subfiles are searched via the volumenumber of the disc where the subfiles reside. This volumenumber is recorded in the descriptor file. Due to this, problems can occur when two discs with the same volumenumber are mounted because the subfiles are then searched on the wrong volume.

Note: The 'SCL FCD' command must be used to assign disc filecodes /CO to /CF. DAD's can also be assigned by LKM 71.

Error Messages

WRONG PARAMETER
FILE CODE UNKNOWN
2ND FILE CODE UNKNOWN
EQUIVALENCE ASSIGNMENT IMPOSSIBLE
TOO MANY PARAMETERS
I/O ERROR
NOT ENOUGH SPACE IN DYNAMIC AREA
DEVICE UNKNOWN
WRONG DAD NAME
ERROR ON DISK FILE CODE
WRONG NO. OF GRANULES
DAD FILE CODE DOESN'T EXIST
DISK UNIT FILE CODE CREATION ONLY IN SYSTEM MACHINE
ERROR IN LAST PARAMETER
DAD NOT FOUND
DISK UNKNOWN
DAD OVERFLOW
WRONG FILE TYPE
WRONG FILE CODE
FILE UNKNOWN

For filetype EF, the following error messages may also occur:

DELETE OLD F.C. ERR, ST= xxxx

ERR IN CREATING FDC, ST= xxxx

ASSIGN ERR, ST= xxxx

For the TDFM errors, an explanation of the status can be found in appendix C, LKM 23.

For a more extensive error description see Chapter 7 SCL ASG command.

Purpose

To cause the FCL task to exit and release the peripheral devices used to receive and log FCL commands.

FORMATS

BYE [m1[,m2]...] from MAS 8.50: BYE [\$\$,][m1[,m2]..]

m1, m2, etc. are machine names. If one or more of these parameters are entered, then those machines will be started as though an SB or SM command had been given for each one after the BYE command.

\$\$ is an indication that open spool files have to be closed and unspooled.

Remarks

Since this command terminates the FCL task, it must be the last given. The user application continues until the last active task has exited, and the foreground machine then enters an idle state until the FCL task for this machine is reactivated using the 'SM' operator command.

Purpose

To cause the FCL task to exit and open spool files to be closed and unspooled (till MAS release 8.50) or to close and unspool one or all open spool files (from MAS release 8.50).

Format

CLS [m1[,m2]..] or from MAS 8.50 CLS [/fc|ALL]

m1,m2 are machine names. If one or more of these parameteres are entered, then those machines will be started

/fc is a spooled filecode which is to be closed and unspooled.

ALL indicates that all spooled filecodes are to be closed and unspooled.

Format

DAS lcl,dn[da[l]]
DAS lcl,lc2

lcl is the linecode to be assigned from 0-255
dn is the device name (see Appendix A).
da is the device address (mandatory for AMA8 devices).
l is the linenumber (for LSM16 and AMA8).
lc2 is the linecode to which lcl has to be assigned by equivalence.

Remarks

The linecode is assigned by inserting it in the consecutive linecode block or by creating an alternate linecode block in the system dynamic area. Whether or not an alternate linecode block is created in depending on the value specified in the DLN command during machine declaration or on the (default) value in the CVT.

Error messages

LINE CODE ERR
LINE CODE NOT NUMERIC
LINECODE MISSING
INV. LINECODE
DEV NAME ERROR
DEV NAME MISSING
FOR AMA8, DEVAD MUST BE GIVEN
2ND LINECODE ERROR
TOO MANY PARAM
DYN AREA OVERFLOW
DEV. UNKNOWN
2ND LINECODE NOT ASSIGNED
DEV. ADDR ERROR
ASSIGN ERR (ST=xxxx)

For an explanation of the status, see Appendix C LKM 48.

Format

DDC lc

lc is the linecode to be disconnected.Remark

This command is translated into a Datacom call LKM 8 with order /10.
For more information about this LKM, see the Datacommunication manuals.

Error messages

LINECODE MISSING
LINECODE ERROR
LINECODE IS NOT NUMERIC
LINECODE TOO BIG
WHAT IS THE SECOND PARAM?
DISCONNECT ERR ST=xxxx
HALT ERR ST=xxxx

The meaning of the status is described in the Datacommunication manuals.

NO DATACOM IN YOUR SYSTEM

No Datacommunication devices were generated.

Format

DDL lc

lc is the linecode to be deleted.

Remarks

The linecode to be deleted is removed from the consecutive linecode table or its alternate linecode block is removed from the system dynamic area.

Error messages

LINE CODE MISSING
LINE CODE ERROR
LINE CODE IS NOT NUMERIC
LINE CODE TOO BIG
WHAT IS THE 2ND PARAM

Format

DHD lc

Remark

The DHD command has the same effect as a DDC command followed by a DHL command.

Error messages

The error messages returned by this command are the equal to the error messages of the DDC command.

DHL

Halt a line

DHL

Format

DHL lc

lc is the linecode to be disconnected.

Remark

This command is translated by the system into a LKM 8, order /11.
For more information, see the Datacommunication manuals.

Error messages

The error messages output by the DHL command are the same as for the DDC command

Format

DLF dc,fn[,ft][,fv]

dc is the DAD filecode identifying the DAD in which the file exists. The file must be catalogued in the Directory for the first userid of this DAD. This DAD code must have been defined by an 'SCL FCD' command when this foreground machine was defined, by a previously given ASG command (format 5) or by a LKM 71 (assign DAD).

fn is the filename of the file to be deleted.

ft is the file type (e.g. SC, OB, UF or LM). Default is UF.

fv is the version (0-7) of the file to be deleted. The default value is the latest (newest) version, i.e. version 0. The version numbers of the remaining older versions of this file are all decremented by one.

Remarks

The user can also use LKM 41 to delete a file. The specified version number must not exceed the value given on the SMV command to the LIB processor, and must exist in the Directory.

Error Messages

DAD F.C. ERROR
DAD F.C. NOT NUMERIC
DAD F.C. MISSING
DAD F.C. NOT ASSIGNED
INVALID DAD F.C.
F.C. NOT ASSIGNED TO A DAD
FILENAME MISSING
FILENAME ERROR
FILENAME IS NUMERIC
FILETYPE ERROR
FILETYPE IS NUMERIC
FILE NOT CATALOGUED
VERSION NBR ERROR
INVALID VERSION NBR
TOO MANY PARAM
DYN AREA OVFL
VERSION NUMBER NOT NUMERIC

Purpose

To disconnect programs that are no longer required from their software levels, so that other programs may be connected and activated. This command can also be used to change the software level of a program by immediately re-connecting to a different level using the 'CNL' command.

Format

DSL p

p is the program name.

Remarks

The number of programs that can be connected to software levels at any one time depends on the number of software levels specified at SYSGEN. The maximum number of programs that can be connected is $\underline{n} - 1$, where \underline{n} is the level specified for the Idle Task. The maximum value of \underline{n} is 239. A program that has been activated cannot be disconnected until it has exited from the last task in its activation queue.

Error Messages

PROGRAM NAME NOT ALPHA.
PROGRAM NOT FOUND
PROGRAM ACTIVE, DSL IMPOSSIBLE
PROGRAM NOT CONNECTED

Purpose

To prevent further activations of a program which is being regularly activated following a 'CNT' command or LKM 10 request.

Format

DST p[,z]

p is the program name

z identifies which timer the program 'p' is to be disconnected from.
Possible values are as follows:

- 1 = minutes.
- 2 = seconds.
- 3 = 100 msec.
- 4 = 20 msec.
- 5 = non-standard timer.

In the event of the optional parameter 'z' not being specified, the default value is all the timers to which the program is currently connected.

Remarks

No further activations of the program will occur due to the specified timer(s), but the program is still known to the system and is still connected to a software level.

Error Messages

PROGRAM NAME NOT ALPHA
PROGRAM NOT FOUND
WRONG TIMER NUMBER
TIMER NUMBER > 5
TIMER 5 CANNOT BE USED
TOO MANY PARAMETERS
PROG NOT CNTCD TO THIS TIMER
PROGRAM NOT CONNECTED TO A TIMER

Purpose

To list specified sectors of a file onto the printer.

Format

DUF fc,sf[,sl]

fc is either a DAD filecode (/F0 - /FF) or a filecode assigned to a disc file.

sf is the number of the first sector to be dumped.

sl is the number of the last sector to be dumped, if omitted only one sector sf is dumped.

Error Messages

FC PRT NOT ASGN
DYNAMIC AREA OVERFLOW
PARAM ERROR
PARAM MISSING
DAD F.C. NOT ASGN
READ FILE I.O. ERROR
OUT OF LIMITS FILE
2ND ADDRESS LESS THAN 1ST ADDRESS

Purpose

To list sequentially all or part of a segment of a foreground machine.

Format

DUM sn,fa[,la]

sn is the number of the segment containing the area of memory to be dumped to the device assigned to filecode /02. Permitted values range from zero to the number given on the SCL DCF command when defining this foreground machine.

fa is the virtual address within segment 'sn' of the first memory location to be dumped (as a 16-bit hexadecimal number).

la is the virtual address within segment 'sn' of the last memory location to be dumped (as a 16-bit hexadecimal number). The default value, if this parameter is omitted, is the virtual address of the last memory location allocated to the segment by the 'SCL SEG' command which defined this segment.

Remarks

Only memory assigned to memory-resident foreground programs and dynamic buffers can be dumped with this command. The 'DM' operator command can be used to dump memory assigned to disc-resident programs of a foreground machine .

The dump addresses specified must be within the given segment, otherwise the command will be rejected.

Error Messages

FC PRT NOT ASGN
DYNAMIC AREA OVERFLOW
PARAM ERROR
PARAM MISSING
SEG. NUMBER UNKNOWN
DUMP ADDRESS FORBIDDEN

Format

EOJ

Effect

The default Userid and DAD in the Foreground machine are reset to the first Userid of DAD /FO.

Remark

The EOJ command has been implemented via a special interface. The coding of this command is stored in the load module EOJ, residing on the system userid. The command is only available in releases from MAS 8.50.

Error message

ERR ON SET USID-DAD, ST=xxxx (in decimal value, the status returned by LKM 73).

Purpose

To move an object file or a member from an object file to the filecode /D5 (/O file) to be an input for the Linkage Editor or the Librarian. This command is implemented from MAS release 8.50.

Format

INC [fc] or

INC libr,[mnam][,usid,dadfc]

fc is a filecode assigned to a device from which the object code has to be read. The device is read up to the EOF mark. The default filecode is /E2.

libr is the name of an object library.

mnam is the name of an object module. If not specified, the whole object library is copied to /D5.

usid is the userid, where the object library can be found.

dadfc is the filecode of the DAD, where the userid resides. The defaults for usid-dadfc is the first userid of DAD /FO, or the defaults, set by a LKM 73.

Remark

The effect of the INC command is exactly the same as for the BCL INC command. The coding of the INC command is not stored in the segment file (D:MASG or D:MSEG) but is read via a special interface from a separate load module INC on the system userid.

Error messages

1ST PARAM (F.C. OR LIBR NAME) ERR

2ND PARAM (MNAM) ERR

3RD PARAM (USID) ERR

4RD PARAM (DAD F.C.) ERR

ERR ON INPUT F.C. ASSIGNMENT, ST=xxxx (xxxx is returned by LKM 23)

ERR ON GET INFO. ABOUT F.C. yy

ERR ON DELETE F.C. /DO

NOT DFM FILE OF TYPE OB

I/O ERR ON GRANTB READ OF INPUT FILE

ERR ON OUTPUT f.c. /D5 ASSIGNMENT, ST=xxxx (returned by LKM 23)

OUTPUT SECT. SMALLER THAN INPUT SECT.

I/O ERR ON INPUT FILE

MODULE NOT FOUND

I/O ERR ON OUTPUT FILE, ST=xxxx (returned by LKM 1)

ERR ON GET INFO ABOUT INPUT F.C.

F.C. TYPE MUST BE DFM OR PHYSICAL

READ ERR FROM DEVICE

BAD GET COMMAND.

Purpose

To change the default DAD and Userid in a Foreground machine. This command is implemented from MAS release 8.50.

Format

JOB [usid,dadfc]

usid is a name of a userid.

dadfc is a DAD filecode. The default for usid-dadfc are: first userid of DAD /FO.

Remark

The implementation of this command is done via the same interface as the INC command, via the load module JOB on the system userid.

Error messages

BAD GET COMMAND

2ND PARAM NO DADFC

WRONG PARAM

ERR ON SET USID-DAD, ST=xxxx

xxxx is a decimal value, indicating the status returned from the LKM 73 (set default usid and DAD).

Purpose

To disconnect and delete a foreground program.

Format

KIL p

p is the program name expressed as ASCII characters.

Remarks

The program must be inactive or aborted. It is disconnected from its software level, which thus becomes free for use by another program. The memory areas occupied by save areas and control tables are also freed.

If the program is memory-resident (declared by a LOD or REP command), the memory it was occupying in its segment becomes free for use by other memory-resident programs.

If the program is disc-resident (declared by a RON or SWP command), the space occupied by it in the D:CI swap-image file may be re-used for other disc-resident programs.

Error Messages

These are output to the device associated with filecode /01 of the foreground machine:

PARAM MISSING

INVALID PROG NAME

UNKNOWN PROGRAM

MORE THAN ONE PARAMETER

PROGRAM IN ACTIVE STATE

PAGE TABLE DESTROYED

CHAIN ADDRESS IN SEG DESTROYED

(Memory-resident programs belonging to the same segment are chained, and this chaining address has been overwritten by the user program.)

SYST FILECODE /F1 INVALID

(The filecode /F1 is not assigned to the system DAD D:CI.)

PROGRAM TO BE SWAPPED OUT

PROGRAM BEING SWAPPED OUT

PROGRAM BEING ABORTED

(The last three error messages indicate that the program to be killed is subject of some system task. The KIL command has to be re-issued when the system task has finished)

Purpose

To release all programs loaded into a memory-resident segment.

Format

KIS s

s is the segment number.

Effect

The pages allocated to the segment are released and thus no more programs can be loaded into it.

Note: All programs in the segment must be inactive or aborted.

Error Messages

One of the following error messages will be printed if this command is rejected:

PARAM ERROR OR MISSING
UNKNOWN SEGMENT
WHAT IS THE SECOND PARAM?
PROG n ACTIVE
PROG n PAGES TAB DESTROYED
PROG n INV. CHAIN ADD. IN SG. *
PROG n SYST. DAD /F1 INCOR. **

n is a program name.

* Memory-resident programs within a segment are chained; the chain address has been overwritten by a user program.

** The file code /F1 is not assigned to the System DAD D:CI.

In the case of the last 4 messages, the command processing is terminated. The programs following the one giving rise to the error are not released and the pages are not freed.

Purpose

To disconnect and free the pages occupied by a secondary load module, so that they can be reallocated.

Format

KLM n

n is the secondary load module name, as specified on the LSM command which was used to load it.

Remarks

The command is rejected if the secondary load module is connected to a primary load module which is still active.

Error Messages

PARAMETER ERROR

SEC. LOAD MOD. ALREADY DELETED

SEC. LOAD MOD. STILL CONNECTED.

(the secondary load module is still in use by a program. This program must first issue an LKM 57 - disconnect a secondary load module)

Purpose

To catalogue a disc temporary file.

Format

KPF fc,fn[,ft]

fc is the filecode defining the disc temporary file to be catalogued. This filecode must have been assigned previously, either by an 'FCL ASG' command to this foreground machine, or by use of an LKM 33 or 21 issued by a program within this foreground machine.

fn is the file name to be placed in the first Directory of the DAD to which this file has been written. It consists of from 1 to 6 ASCII characters.

ft is the type of file, e.g.:

- UF = User File (default value);
- OB = Object File;
- SC = Source File;
- LM = Load Module.

This need not be the same as the type specified when the filecode 'fc' was assigned. The type will be catalogued in the same Directory as the filename, and version number will be set to zero.

Remarks

If previous versions of this file exist (with the same filename and type) within the same Directory, the oldest version will be deleted if its version number is equal to the value specified on the SMV command to the LIB Processor for this Userid. Other versions will have their version numbers incremented by one. Usually a file is catalogued by means of the LKM 40 request, but the KPF command can be used to catalogue a valid file created by a program which terminated abnormally before the LKM 40 could be issued.

Another use of the KPF command is where a foreground application does not have a clearly defined termination. Instead of the user having to activate a special program to catalogue his files, he may issue KPF commands instead.

Error Messages

FILE CODE MISSING
FILE CODE ERROR
FILE CODE NOT NUMERIC
INVALID FILECODE
FILE NAME MISSING
FILE NAME ERROR
FILE NAME IS NUMERIC!
FILE TYPE ERROR
FILE TYPE IS NUMERIC!
TOO MANY PARAM
FILE CODE NOT ASSIGNED
NOT A DISK FILE
ALREADY CATALOGUED
INVALID FILE TYPE
KPF ERROR
DYNAMIC AREA OVERFLOW
I/O ERROR
DIRECTORY OVFL
KEEP FILE ERROR
NO USERID (the file had to be kept on a DAD not containing any Userid)

Purpose

To provide the user with the same information as that provided by use of the PRS command for all programs in his foreground machine, i.e.:

Program Name.
Program Type:
 SWP = Swappable;
 REE = Re-entrant;
 RON = Read-only;
 Snn = Loaded into segment nn.
Software Level.
Active state.
Whether loaded, and if so the address.
Whether swapped out or not.

Format

MAP fc

fc is the file code assigned to the device on which the map is to be printed. The default value is /01.

For an example of a MAP printout, see Chapter 7, the MAP Command.

Error Messages

INVALID PRINT FILECODE
PRINT F.C. ASSGN TO NO DEVICE

Purpose

To declare a node in the temporary object file as input for the Linkage Editor.

Format

NOD nodename[,rovs] [,a]

<u>nodename</u>	is a 6 ASCII character name, conforming the same rules as any external program name or label. This is copied to the temporary object file (/D5) as a card image, to be processed by the Linkage Editor. It defines a branch in an overlay tree.
<u>rovs</u>	is a 4 character segment name, referring to a secondary load module name. This must not duplicate any other entry point, common block or segment name.
<u>*</u>	is a signal to the Linkage Editor to implicitly name the segments in this compilation in ascending order, in accordance with the parameters in the LKE option command.
<u>a</u>	is an absolute address in hexadecimal of a page boundary. It forces the Linkage Editor to load the module at this address

Remark

FCL copies the command to the temporary object file /D5. It will be processed later by the Linkage Editor. The NOD command has been implemented via a special interface, so that its coding is not in the MAS segments file (D:MASG or D:MSEG) but in a separate module NOD on the system userid.

Error messages

```
ERR ON /D5 ASSIGNMENT, ST=xxxx LKM 23 status, printed in decimal)
ERR ON GET INFO ABOUT /D5
WRITE ERR ON /D5
BAD GET COMMAND
```

Purpose

To print a map of the program layout in a user machine.

Format

PCM [fc]

fc is the print filecode; the default value is /01.

Error Messages

INVALID PRINT FILECODE
PRINT F.C. ASSGN TO NO DEVICE.

Purpose

To obtain a list of all filecodes assigned to this machine.

Format

PFC [fc]

fc is the filecode assigned to the device on which the filecodes are to be printed. The default value is /01.

Error Messages

INVALID PRINT FILECODE
PRINT F.C. ASSGN TO NO DEVICE

Purpose

To obtain a list of all the linecodes, which have been assigned in the Foreground machine.

Format

PLC [fc]

fc is a filecode assigned to a device on which the linecodes are printed. The default value is filecode /01.

Error messages

PRINT FILECODE NOT ASSIGNED
PRINT F.C. ASSIGNED TO NO DEVICE

Purpose

To obtain a list of all the software levels which are in use i.e. connected) in the bare machine.

Format

PLV [fc]

fc is the filecode assigned to the device on which the software levels are to be printed. The default value is /01.

Remarks

For each level the system gives in reply:
The machine name.
The program name.
The program type.
Whether or not the program is active.

Error Messages

INVALID PRINT FILECODE
PRINT F.C. ASSGN TO NO DEVICE

Format

PRG p,fc,{A | M | S}

p is the program name (1-6 ASCII characters).
fc is the filecode assigned to the printer on which the listing is to appear.
A is a code indicating that the contents of all registers are to be printed (both main sequence and scheduled label).
M indicates that only main sequence registers are to be listed. Word 5 of the PCT is used to find the save area for these registers.
S indicates that only scheduled label registers are to be printed. Word 26 of the PCT is used to locate the save area for these registers.

Remarks

Only one of the symbols A, M or S must be entered; this field is mandatory.

Error Messages

INVALID PROGRAM NAME
PROG NAME MISSING
PROG NAME TOO LONG
PROG NAME UNKNOWN
INVALID PRINT F.C.
PRINT F.C. MISSING
PRINT F.C. NOT ASSIGN
INVALID PRINT CODE
PRINT CODE MISSING
PRINT CODE TOO LONG
PRINT CODE UNKNOWN

Format

PRS p[,fc]

p is the program name 1-6 ASCII characters).
fc is the filecode of the print device on which the status is to be printed. The default value is /01.

Remarks

The following information about the program 'p' is output to the specified printer:

Program name.

Program type:

SWP = swappable

REE = re-entrant

RON = read-only

Snn = loaded into segment 'nn'

Software level.

Whether active or not.

Whether or not it is loaded, and if so the address.

Whether swapped out or not.

Error Messages

PARAM ERROR

PROG NAME MISSING

PROG NAME UNKNOWN

PRINT F.C. PARAM ERROR

PRINT F.C. BAD ASSGN

INVALID PRINT FILECODE

PRINT F.C. ASSGN TO NO DEVICE

Purpose

To suspend a foreground program. The program can be re-started using the 'FCL RST' command.

Format

PSE p

p is the program name (1-6 ASCII characters).

Remarks

This command gives the user similar facilities to those provided by the 'PS' operator command.

A Middleground program may be suspended between the 'RUN' command and its subsequent exit.

A re-entrant program cannot be suspended, since MAS cannot identify the appropriate PCT.

Error Messages

PARAM ERROR
PROG NAME MISSING
PROGRAM UNKNOWN
PROGRAM INACTIVE
PROGRAM ALREADY IN PAUSE
SYSTEM PROGRAM

Format

RAB p

p is the program name (1-6 ASCII characters).

Remarks

When a foreground program is aborted it remains in the foreground machine, but the abort flag in the PCT word 7, bit 1) is set and subsequently the program can only be activated by an 'FCL CNT', an 'FCL RUN' or 'FCL ACT' command.

The LKM's 10 or 12 and activations because of a previous CNT command are ineffective. The RAB command allows the user to reset the aborted foreground program.

Error Messages

PARAM ERROR
PROG NAME MISSING
PROG NAME UNKNOWN
PROG INACTIVE
PROG NOT ABORTED

Purpose

To allow the foreground user to cancel an I/O operation which did not terminate and caused a 'PU' (Physical Unit) message to be output. This 'PU' message must have ended with 'RY' (Retry), otherwise a release is not allowed.

Format

RDV da

da is the device address given in the preceding 'PU' message.

Remarks

The 'RDV' command provides the user with similar facilities to the 'RD' operator command.

Error Messages

PARAM ERROR
DEVICE ADDRESS MISSING
DEVICE ADDRESS UNKNOWN
DEVICE NOT IN RETRY

Format

RST p[,n]

p is the name of the program to be restarted.
n is a value to be placed in A7 when the program is restarted. The default value is zero.

Remarks

This command gives the user similar facilities to those provided by the 'RS' operator command.

Error Messages

PARAM ERROR
PROG NAME MISSING
PROG NAME UNKNOWN
BAD A7 PARAM VALUE
PROG NOT IN PAUSE

Purpose

To make another attempt to execute an I/O operation, when a previous attempt was unsuccessful but a retry was invited by a system 'PU' message that ended in 'RY'.

Format

RYD da

da is the device address given in the preceding 'PU' message.

Remarks

A 'PU' message ending in 'RY' must have been output before attempting a retry command.

This command provides the user with similar facilities to those provided by the 'RY' operator command.

Error Messages

PARAM ERROR
DEVICE ADDRESS MISSING
DEVICE ADDRESS UNKNOWN
DEVICE NOT IN RETRY

Purpose

To de-assign a particular filecode from the device or file to which it is currently assigned.

Format

SCR fc

fc is the filecode which is to be removed from the filecode table for this foreground machine.

Remarks

If the filecode is assigned to a disc file, the memory used by its blocking buffer is released.

If the filecode is assigned to a disc temporary file and no LKM 40 (Keep File) or 'SCL KPF' command has been given for the file, the granules allocated to it are released.

This command cannot be used to scratch a DAD filecode.

Error Messages

FILECODE MISSING
FILECODE ERROR
FILECODE IS NOT NUMERIC
FILECODE TOO BIG
I/O ERROR
SCR ERROR
DELETE ERROR
DAD F.C.

TIM

Get the Time

TIM

Format

TIM

Remarks

The time is output to the physical device assigned to filecode /01.

If the timer does not contain a binary value which can be converted into a time expressed as HH/MM/SS, where:

HH is a number of hours in the range 0-23,
MM is a number of minutes in the range 0-59, and
SS is a number of seconds in the range 0-59,

then the error message:

THE DATE IS DESTROYED

is output to the device assigned to filecode /01.

Purpose

To modify memory-resident segments including Segment 0) of a foreground machine.

Format

WRM s,a,v1[,v2[,v3]...]

s is the segment number within this foreground machine containing the locations to be changed.

a is the relative address of the first location to be modified, expressed as decimal or hexadecimal to a maximum of 16 bits.

v1, etc. are the new values which will be inserted in locations a onwards, starting at location 'a'.

Remarks

The WM or CR operator commands may be used to modify memory occupied by disc-resident programs of a foreground machine.

Error Messages

WRONG PARAMETER
WRONG SEGMENT NUMBER
WRONG LOCATION
YOU ARE TRESPASSING
WRONG VALUE
NO VALUE

Processor call

Purpose

To RUN a middleground program, without the necessity to type the RUN command.

Format

ccc

ccc is the name of a load module residing in the system userid.

Remark

The effect of the processor call is the same as the effect of the RUN command of a middleground program.