

PART 3

THE BACKGROUND MACHINE

GENERAL

MAS can support simultaneously a number of foreground machines and, optionally, one background machine. Once created, the background machine exists until the next IPL, unless an 'SCL KIM' command (Kill Machine) is given for it.

BACKGROUND MACHINE PROGRAMS

General Characteristics

Only one background program can be active at a time. Background programs (apart from the BCP) are activated, either by a BCL RUN command or by a BCL Processor-

call command. They are deactivated by an LKM 3 issued by the program (other than in a scheduled label routine).

They may be placed in a Pause state by a SCL PSE command or by a PS operator command (except for the BCP, which may only be placed in a pause state by the BCL PSE command).

Except for the BCP, they may be placed in an Abort state either by MAS or by an AB operator command or by an SCL ABT command or by LKM 46. Unlike a foreground program, this does not cause them to remain in a non-executable state in the machine.

They may enter a Wait state in exactly the same manner as foreground programs.

No Activation queue is maintained for a background program. The active background program is executed whenever it is the executable task with the lowest software level (highest priority) known to MAS.

The background program executes without any communication with the simultaneously active foreground programs. It is not connected to a timer or a clock, and cannot be connected to a level by a foreground program or by a previously active background program.

The background machine may be declared non-swappable or swappable by the system manager, with the SCL DCB command. This will determine whether every background program executed in the background machine is memory-resident or disc-resident. Particularly it allows batch processing and is therefore sometimes referred to as the batch machine.

Memory-Resident Background Programs

If the System manager defines the background machine to be non-swappable (with the SCL DCB command), then all batch programs will be memory-resident. The number of pages specified in the DCB command will be permanently reserved for the background machine, until the next IPL. Background programs are loaded upon activation into these reserved pages, and remain there until the program is deactivated. Operating the Background Machine in this way may reduce the execution time of the Background Machine (because the background programs are never swapped out), but may increase the execution time of the foreground machines (since the foreground machines are not able to use the pages reserved for the background machine).

Disc-Resident Background Programs

Unless the System manager specifically requests a non-swappable background machine, MAS will define the background machine as swappable and will, during the session, allow the background machine to reserve up to 16 pages (32K words) from the same pool of pages that is used for the swappable (declared by FCL RON or SWP commands, or middle-ground) foreground programs. The background machine will be swapped out and in by MAS during its execution. Operating the background machine in this manner may cause the execution time of background programs to increase, but may improve the overall throughput of all the machines, since there will be more memory available for foreground programs. The SIZE parameter on the BCL RUN or Processor-call commands specifies the number of pages which the program assumes it may address, additional to the ones occupied by the load module. When MAS loads or swaps in the program, it will reserve the number of pages required to contain the load module, plus the number of pages of the SIZE parameter.

Programming Techniques

When coding a background machine program, it is not necessary to know whether the system manager will execute the program in a Non-swappable or a Swappable Background Machine. There is no need for a background program to allocate its variables externally, nor for the load module to consist only of constants and instructions. A background program is loaded from the libraries on disc at execution time every time it is activated, so there is no need to reset program variables and accumulators back to their initial values at the end of the program. Background programs may allocate and free dynamic storage using the LKM 4 and 5 Monitor calls. MAS will allocate the dynamic storage from the background machine dynamic area. Although the nature of this area for a swappable background machine differs from that for a memory-resident background machine, the difference is transparent for the programmer providing the SIZE parameter has, if necessary, been used on the BCL RUN command. The LKM 4 and 5 Monitor calls are identical for both types of background machine.

BACKGROUND MACHINE MEMORY ORGANISATION

Background Machine Types

The system manager defines the type of background machine with the SCL DCB command. Two types of background machine are supported by MAS:

- Non-swappable
- Swappable.

Non-Swappable Background Machine

The background machine consists of the number of pages requested in the SCL DCB command. These pages are reserved exclusively for the background machine. All activities in the background machine will use these pages.

The pages may thus contain:

- The BCP (when reading a BCL command)
- A user program activated by a BCL RUN command
- A processor activated by a BCL Processor-call
- Dynamic buffers.

Dynamic buffers are allocated whenever the active program in the background machine issues an LKM 4 Monitor Request. They are allocated by MAS from the pages which were allocated by the DCB command, but which are not being used by the active background program. In other words, the Dynamic Area for a non-swappable background machine consists of the pages reserved for the background machine but not being used by the active background program. This means that the program could in fact simply use virtual addresses to address the dynamic area, without giving any LKM 4. In a swappable background machine this can only be done if the SIZE parameter is given on the BCL RUN command.

Example

Suppose the systems manager defines a 10 page memory-resident background machine thus:

```
DCB 10
```

If an 11K words program is activated (by a RUN command for example) by a background user, then Pages 0-4 and the first 1K of page 5 contain the program. The second 1K of page 5 and pages 6-9 are the dynamic area for LKM 4 or may be virtually addressed directly by the program.

Swappable Background Machine

If the number of memory-resident pages is not specified on the SCL DCB command, the pages for the background machine are allocated dynamically by MAS from the Dynamic Loading Area shared by the foreground machines. At any moment, the entire background machine is regarded by MAS as a swappable disc-resident program, and its memory is allocated exactly as for a disc-resident foreground program.

Example

Suppose the systems manager defines a swappable background machine thus:

```
DCB
```

If an 11K words program is activated by a background user by a RUN command with the parameter SIZE=2, then pages 0-4 and the first 1K of page 5 contain the program. The second 1K of page 5 and pages 6-7 are the dynamic area for LKM 4 or may be addressed directly by the program.

DEFINING A BACKGROUND MACHINE WITH SCL

Introduction

The background machine is defined by the Systems Manager using the following SCL commands:

```
DCB  Define the background machine
FCD  Define a filecode for a machine
MFC  Define the maximum number of spare filecode table entries for a machine
MBF  Define the maximum number of blocking buffers for a machine
LAB  Define the maximum number of scheduled labels for a machine
DEN  End machine definition
LSM  Define a secondary load module for the background machine
KLM  Kill a secondary load module from the background machine.
```

The last two commands must be given outside the DCB-DEN sequence.

Two other SCL commands (DLN and DLC) are used to define the datacommunications lines used by the background machine.

The system machine stops reading the SCL commands when the BYE command has been read from the SCL file on /EO.

The operator may start the background machine with the Operator Command SB, after the DEN command has been processed or by specifying "BATCH" in the BYE command.

Example of Background Machine Definition

```
DCB 8
FCD 1,TY10
FCD 2,LP07
FCD /EO,TY10
FCD /E1,CRO6
FCD /CO
FCD /C1
FCD /C2
FCD /C3
FCD /FO,/CO,SUPERV
FCD /F1,/C1,DAD1
DEN
BYE
```

The SB Operator Command can then be given to start the batch machine, which will cause the first BCL Command to be read from the device assigned by SCL FCD command to the background filecode /EO.

If BATCH is entered as a parameter on the 'BYE' command, the background machine is started automatically.

Filecodes /FO, 1, 2 and /EO mandatory in the background machine.

When, for any physical device, only one has been generated or when the first declared one is meant to be assigned, the device address can be omitted.

Filecodes /C2 and /C3 as they are declared in the example, can only be used, when in the BCP DADs are assigned to these filecodes (via the ASD command or LKM 71).

BACKGROUND MACHINE FILECODES

MAS maintains a filecode table for each machine, which (for user machines - foreground or background) it initialises from SCL FCD commands when the machine is defined. The filecode tables can subsequently be extended or modified by programs which are run in the relevant machine. This is done by LKM Monitor Calls to MAS issued by User programs, or by the command language (FCL or BCL) processors. For the foreground machines this allows unlimited flexibility of task mixes. For the background machine it also has advantages. It is not necessary for every program to have a command defining the files it uses. Files or devices which are used by several programs need only to be assigned once when the background machine is defined, and thereafter every job and program can refer to the file or device merely by specifying the filecode which identifies the relevant entry in the MAS filecode table for the background machine.

Filecodes which are defined by SCL FCD commands remain in the MAS filecode table from the time the SB operator command is entered, to the time the next IPL is made, unless an SCL KIM command is given for the background machine. The BCL :EOB command does not remove them, so there is no need to re-specify them before a subsequent SB operator command is given. New filecodes can be added by Jobs, either by BCL ASG, ASD or REQ commands or by LKM 33, LKM 54, LKM 71 or LKM 23 requests issued by background programs. The same methods can be used to modify the filecodes assigned when the background machine was defined.

Extensions and modifications to the background machine filecode table are cancelled by the BCP at the end of the Job, and the filecode table is restored to the status it had when the background machine was defined.

The BCP and all the Standard Processors have been developed in a way which allows all the devices and files which they use to be defined once and once only by SCL FCD commands. Any temporary filecode assignments which these programs may make are performed by LKM 23 and use the filecodes /DO to /DF. This makes it very easy to define the background machine and to run the BCP and the Standard Processors.

The filecodes which the Systems Manager must assign with SCL FCD commands entered at background machine definition time are /01, /02, /EO, /FO and all disc codes required by user Jobs.

The following are the filecodes reserved for the BCP and Standard Processors:

- /01 error messages to the background user
- /02 printed output
- /EO BCL commands
- /E1 ASCII input files
- /E2 Binary input files
- /D* Standard Processor temporary filecodes
- /ED Catalogued Procedures work file
- /EC Filecode assigned by BCP to the file B:PROC in the Job USID and DAD
- /EE Catalogued Procedures work file
- /FO This filecode defines the System DAD (whose name, unless modified at SYSGEN, is SUPERV). The first directory in this DAD can be accessed as the USERID SYSTEM, even if another name is has been stored in the catalogue of the DAD. This directory must locate the file B:PROC of catalogued procedures for all the background machine users. Note that the system machine filecode /FO also refers to the SYSTEM user in the System DAD, in order to locate the Standard Processors.
- /C* Disc drives.
- /F* User DAD filecodes.

Any other user filecodes assigned should obey the following rules (regardless of how the assignment is made - by SCL FCD, by BCL ASG, BCL ASD or by BCL REQ or by LKM 33, LKM 54, LKM 71 or LKM 23):

- Take care, using any of the filecodes listed above.
- Do not use the filecodes: /DO - /DF, reserved for the Standard Processors and the BCP to assign by LKM.

Filecodes /S, /O and /L

The FCOD parameter occurs on many BCL commands. The background user may, if he wishes, specify the filecodes /S, /O or /L. This facility is provided for compatibility with DOS. The BCP will convert these filecodes as shown below:

FCOD=	CONVERTED
/S	/D4
/O	/D5
/L	/D6.

Background Machine Filecode /EO

This filecode is used for reading BCL commands and Standard Processor commands, in other words, the set of commands which comprise the batch of background machine jobs.

The filecode /EO must be assigned to a device by a SCL FCD command. If the device is a disc file, tape, card reader or paper tape reader it should not be used by any other filecode in any machine. The batch of BCL and Processor commands will be read as a sequential file.

If the device is a console or display, then whenever the BCP wishes to read the next BCL command, it will output the message:

BCP:

and the user must key in the next BCL command. Similarly, the Librarian Processor will output the message:

LIB:

when it requires to read the next Librarian Command. The user should then key in the next Librarian Command (or LEN to stop the Librarian).

The Processors ASM, and FRT do not signal when they require to read a control command. They should be keyed in by the user when appropriate. For example:

```
BCP:ASM
ASM:OPT PROG=/EO
      IDENT  BB
      HJ     DATA  56
      END
```

If /EO is a console or display, the device may be shared by other filecodes and files. In the example above, the source program for the ASM Processor has been keyed in on /EO.

Background Machine ERR Device

The ERR device is used to report all rejected BCL commands, and to request that the user resubmits the corrected command. If a command is rejected, 3 lines are output on the ERR device. These are the command, the error message and the program identifier. For example, if a BCL :LIB Processor Call has an invalid DUMP parameter, the BCP will output on the ERR device:

```
LIB DUMP=BLL
  INVALID PARAM=DUMP
BCP:
```

and the user should re-enter the LIB command on the 3rd line with the corrected DUMP parameter.

Under some circumstances, the first line will not be output. This occurs when it is already the most recent line on the device. For example, if the BCP device (filecode /E0) is assigned to a console or display and the ERR device is assigned to the same device as /E0, then the invalid BCL command will have been keyed in by the user and thus already be displayed. There is no need for line 1 to be output in this case. The same applies if filecodes /O2 and the ERR device are assigned to the same device, since all commands are copied to /O2.

The ERR device may be shared by several filecodes in several machines, if the other filecodes allow this.

The default error correction device is /O1. which specification is mandatory during machine definition.

Background Machine Filecode /O2

This device is used by the BCP to output a copy of all the commands that have been read from /E0. If a command is rejected, the error message sent to the ERR device and the respecified command subsequently entered from the ERR device will also be output on /O2. The date and time is also output whenever a job starts or ends.

If /O2 and ERR are assigned to the same device, the error message and the re-specified command are not output to /O2, since they are already present.

If /E0 is assigned to a console or display and /E0 and /O2 are assigned to the same device, the input to /E0 is not output to /O2.

The device assigned to background machine filecode /O2 may be assigned to several filecodes in several machines if it is allowed to share these other filecode devices.

Background machine filecode /O2 is also used for all "printed" output from the Standard Processors.

Background Machine Filecode /O1

When no ERR command has been given, error messages are output and can be corrected via filecode /O1, which has to be declared during machine definition.

ACCOUNTING ROUTINES

Functions

Accounting routines may be written by the user to perform the following functions for the background machine:

- To verify that the account number specified for each background machine job is correct and to stop the job from being executed if it is not.
- To verify that the password specified for each background machine job is correct and to stop the job from being executed if it is not.
- To output information when each background machine job ends.

Naming Restriction

Two accounting routines may be written, one called USRJOB and the other (which is not required if passwords and/or account numbers are to be verified, but no information is to be output) called USREOJ.

BCP Interface

USRJOB is called by the BCP when a :JOB card is read.

USREOJ is called by the BCP:

- When a BCL :EOJ command is read.
- When a BCL :JOB command is read and it was not the first BCL command read in this batch, and neither was it preceded by a BCL :EOJ command. This means the previous job has ended without an :EOJ command. In this case the BCP calls USREOJ first, and upon return from that routine, calls USRJOB.
- When a BCL :EOB command is read which was not preceded by a BCL :EOJ command, the BCP calls USREOJ.

USRJOB

The routine may examine the account number and password on the BCL :JOB command, by declaring two 8 character external areas ACNT and PASW respectively. USRJOB should not alter the contents of any register except A7, which is set to 0 if BCP is to accept the job (i.e. read the next BCL command and process it), or a non-zero value if BCP is to reject the job (i.e. ignore all subsequent BCL commands until the next :EOJ, :EOB, :JOB command). Having set A7, the user accounting routine USRJOB must return to BCP with an RTN A14 instruction.

USREOJ

This routine ensures, as USRJOB, that all the registers it modifies are restored to their value upon entry. It returns to BCP with an RTN A14 instruction.

Standard Accounting Information

The BCP outputs accounting information for each background machine Job to the background filecode /02.

- the date and time is output to /02 after every :JOB, :EOJ, :EOB, RUN and Processor Call command read by the BCP from /EO.
DATE: MM DD YY // TIME: **H.**M.**S
- the elapsed time, and the number of records printed and punched, is output to /02 at the end of performing each RUN command.
ELAPSED TIME=****,PRINTED RECORDS=****,
PUNCHED RECORDS=****
- the exit code (value in A7 at LKM 3) is printed after each RUN and Processor Call command.
EXIT /**

DEBUGGING NOTES

When a dump of a program occurs, it will be noted that the LM is preceded by some MAS control tables. Some of the information in these control tables may be useful for debugging purposes.

The contents of these control tables is shown below. Note that word 0 refers to the first word shown in the Dump listing.

WORD	CONTENTS
0	program start address (+1 if the program is segmented into overlays).
1	the number of LM disc sectors containing the root or only segment.
2	program length, including the blank common area if it is not absolute.
3	address of the debug symbol table generated according to the DEBUG keyword parameter of the LKE OPT command.

If the program is not an overlay program, word 4 onwards contains the LM. Otherwise, the contents from word 4 are as described below:

WORD	CONTENTS
4	the amount of memory required to execute the program. This is the length of the longest path plus the amount of memory required for the Segment loaded and its control tables.
5	the number of overlay segments in this LM. The root segment is not an overlay segment.

Each overlay segment has a 4-word entry. The entries occur in the same order as the overlay segments appear in the overlay tree (in other words, in ascending segment level within ascending lowest path for this segment). The first 4-word entry begins in word 6 of the dump.

Each 4-word entry has the format:

WORD	CONTENTS
0	<u>bit 0</u> is 1 if this segment is loaded in memory. <u>bits 1-15</u> contain the number of the 4-word entry for the ascendent segment to this segment (0 if the root segment is the ascendent segment).
1	the address where the segment has been or is to be loaded.
2	the relative sector number in the disc LM file from which the segment has been or is to be loaded.
3	the length of the segment.

The last 4-word entry is followed by the segment loader, which in turn is followed by the root segment of the LM.

By examining bit 0 of all 4-word entries, one can determine which path of an overlay program is loaded and was executing when the dump occurred.

BACKGROUND MACHINE OPERATION

Background Machine Activities

The background machine is used to execute the following programs:

- BCP Background Control Processor
- Standard Processors
- Non-standard Processors
- User Batch Programs
- Accounting routine.

The activities to be performed in the background machine are specified with Background Command Language (BCL) Commands which are processed by the BCP.
THE BACKGROUND CONTROL PROCESSOR (BCP)

Method of Operation

The BCP is loaded into the Background Machine when the SB Operator Command is given. The BCP will read each BCL Command from the device /EO and perform the required operation. If the BCL Command specifies a Processor to be executed, or a user program to be run, this program will be loaded into the background machine, overwriting the BCP. When the program has ended, the BCP will be reloaded to process the next BCL command. When the BCP has completed the processing of a BCL command, a return code is set. This return code may be set either by the BCP or by the program which was executed by the previous BCL command. The return code has a value from /0 to /7F.

The return code should be set by user batch programs (in A7 when they exit with LKM 3), using the same conventions as the BCP and the Processors, namely:

/0	No error
/01-2F	Minor errors
/30-3F	Errors which forbid program execution
/40-4F	Errors which forbid Linkage Edit e.g. error in compilation; no object deck has been produced.
/50-5F	Unused
/60-6F	Errors which forbid FORTRAN or Assembly Processor execution, e.g. error in source update.
/70-7F	Errors which forbid execution of the JOB, e.g. disc overflow, no dynamic buffer available.

Operating Modes

The BCP can be run in two modes: Closed-shop and Interactive.

The Closed-shop mode is used if the user responsible for the Job will not be present in the computer room when the Job is executing, and does not want the operator or anyone else to try to correct any errors that occur.

The Interactive mode is used if the user will be present at a terminal when the Job is run, and can thus interactively correct any errors which may occur.

Closed Shop

In this mode, BCL commands are grouped into steps. A maximum return code is specified for each. If any BCL command causes a return code to be issued which exceeds this maximum value, the remaining BCL commands in the step are ignored. BCL :STP commands are used to specify the closed-shop mode, to define the steps and to set the maximum permitted return code for each command in the step.

Interactive

In this mode, if a BCL command is found invalid, the BCP will allow the background machine to correct the command from a specified device. After the command has been corrected, BCP will process it and then resume reading the BCL from the device /EO.

The BCL ERR command is used to specify the interactive mode and the correction device.

If both the ERR and one or more :STP commands are used in one Job, then correcting commands from the ERR device is only possible for commands which ended with a return code less than the current Step value.

BCL - BACKGROUND COMMAND LANGUAGE

Tasks and Programs

The basic unit of work for the background machine is a task. A task is performed by a program. For example, the LIBRARIAN may be used to copy a sequential file of 20 records. In this case the task is to perform 20 times the sequence of instructions which copy a record from one file to another. If this sequence is, say, 10 instructions, then the task consists of 200 instructions. The program which performs this task consists of over 1000 instructions, only 10 of which are relevant to this task.

A background task may not activate other tasks, but it may activate subtasks by using scheduled labels.

The difference between a task and a sub-task is that the former is not synchronised with the activating task, whereas a subtask is.

Foreground tasks can activate other foreground tasks by LKM 10 or LKM 12, and then continue to execute and even exit without any regard as to whether these other foreground tasks have been performed.

Background tasks cannot do this.

Foreground tasks can create subtasks either by LKM 12 with an associated Wait or by scheduled labels. Only scheduled labels may be used by background tasks to create subtasks.

In the background machine, there is at any one moment only one active task. The active task will be performed either by the BCP, or by a Processor (if the last BCL command was a Processor Call command), or by a user program (if the last BCL command was a BCL RUN command).

Only one program may be active in the background machine at a time, and it may only be activated once. Background programs have no activation queue.

Batches, Jobs and Steps

A batch is a set of BCL commands, the last of which is :EOB. For each batch the operator must give the SB command to load the BCP to process it.

The :EOB command deactivates the BCP and thus frees the devices and memory it was using.

Within a Batch, there can be one or more Jobs. A Job is a series of BCL commands from one user. Each user wishing to make use of the background machine during the current session should submit a separate Job. A user may submit more than one Job.

Each Job may consist of one or more Steps. A Step is a sequence of dependent BCL commands. Thus, if a BCL command is not performed successfully due to some error situation, all subsequent BCL commands in the Step should be ignored. For example, a Step might consist of an Assembly followed by Linkage Edit and a Run of a user program. If the Assembly ends abnormally, Linkage Edit and Run should be bypassed.

The hierarchy of operations is as follows:

```
SEVERAL BATCHES PER SESSION
SEVERAL USERS PER BATCH
1 USER PER JOB
SEVERAL JOBS PER USER
SEVERAL STEPS PER JOB
SEVERAL PROGRAMS PER STEP
```

BCL Functions

BCL performs the following functions:

- defines Batches, Users, Jobs, Steps and Programs.
- assigns filecodes used by the background program to devices, files or DADs.
- allows messages to be sent to the operator when a JOB reaches a certain stage.
- controls the loading of programs in the background machine. (For overlay batch programs, the BCP only loads the root segment. The overlay segments are loaded automatically during the execution of the task. This is done by LKM 27 inserted into the segment loader of the root segment by the Overlay Linkage Editor).

BCL Scheduling Commands

Recommendation

Four BCL commands are used for job scheduling in the background machine. These are the :JOB, :STP, :EOJ and :EOB commands.

MAS will abort any program which tries to read a record commencing with one of the strings:

```
:JOB :STP :EOJ :EOB
```

unless the program is the BCP. Placing the command names for the four BCL scheduling commands in the first record position will therefore ensure that only the BCP may read them. (The abort code is /10 if a program tries to read a BCL scheduling command.)

The BCP will, however, accept these BCL commands with the command name anywhere in the record. See the :STP command description for an exception.

If a user background program input filecode is assigned to the same device as the BCL filecode /EO, and if when this user background program is run it fails to detect the end of its input file (because of a missing :EOF record, for example), then it may erroneously read BCL commands for the next JOB if the :JOB command name for the next JOB is not placed in record position 1.

Job Scheduling Commands

These commands control the initiation and termination of job streams and determine the mode of operation (closed-shop or interactive).

:EOB

End of BCL

:EOB

Format :EOB

If the previous command was not :EOJ the BCP will call the program USREOJ (which is either the dummy system supplied program, or a User accounting routine). The BCP will then issue an Exit (LKM 3) to MAS. The background machine, although still defined, will then be closed until the operator enters an SB operator command. No BCL commands which occur after the :EOB command will be read by the background machine until the SB operator command is given by the operator.

The BCL reacts with the message:

END OF BATCH

:EOJ

End of Job

:EOJ

Format :EOJ

The BCP calls the program USREOJ, which is either the dummy program supplied or a User accounting routine.

If the command is placed in positions 1-4 of the BCL record (as is strongly recommended), it will ensure that under no circumstances can the next Job's BCL commands be erroneously processed as input data for this Job.

The BCP will read all subsequent records from background machine filecode /EO and ignore them until it reads a :JOB or an :EOB command.

Format 1 ERR [FCOD=r]
Format 2 ERR [r]

r is the recovery filecode, an optional parameter specifying the filecode from which the BCP will read corrections when the BCL command being analysed is invalid. The filecode specified must have been assigned to a console or display device type. If the parameter is omitted, filecode /01 of the background machine is assumed.

The ERR command is used to put the background machine into interactive mode. If a BCL command is found to have invalid syntax, the BCP will inform the background machine user by sending an error message to the device assigned to the recovery filecode. From the same device the user may submit the corrected BCL command. After this has been processed, the BCP will resume reading BCL from the device /E0. This facility is only available for correcting BCL syntax errors. It cannot be used to correct errors which occur during processing a valid BCL command. For example, if the PRNT parameter of the BCL RUN command is specified as PRNT=A4, then it will be rejected since the parameter value is not numeric, and can be respecified by the background machine user from the recovery filecode device. If, however, the parameter had been specified as PRNT=50 but the program to be RUN requires to output 70 lines of print, then the program will be run until line 51 is to be output, but the user cannot then alter the PRNT value from the recovery filecode device.

The following error message may be output by the BCP if the ERR command is incorrect:

INV. FCOD

The filecode is not assigned, or is not assigned to a console or display device type.

If no ERR command has been supplied in a Job when a BCL syntax error occurs, all error messages are output and can be corrected via filecode /01.

Format 1 :JOB [USID=u[,DAD=d]][,ACNT=a][,PASW=p][,VOLN=v,DNAM=dn]
FORMAT 2 :JOB [u],[d],[a],[p],[v],[dn]

- u is a userid, a string of up to 8 ASCII characters, which specifies the name of the user directory. This name must already exist. Default value is the first userid on the DAD with filecode /FO. The same userid is taken, when USID=SYSTEM is specified, but then some privileged system manager commands can be issued (like LIB commands DCD, DLD, DCU). The specified userid is default during the whole Job for every command. It can be overridden by a USID parameter specified in a BCL command in the Job. If the parameters DAD, VOLN or DNAM are given, the USID specification is obligatory. If DAD specification parameters are omitted, all assigned DADs are searched to find the specified userid, in the sequence of the declaration of the DADs in the machine declaration.
- d is the DAD filecode (/FO-/FF). It specifies if the DNAM and VOLN parameters are omitted, the DAD where the specified userid is located. When VOLN and DNAM are specified, it gives the filecode, which should be assigned to the DAD where the specified userid is to be found.
- a is an accounting number. A string of up to 8 characters, which is not used by the BCP or any other component of MAS, but may be used by a user accounting routine (if any).
- p is a password. This string of up to 8 characters can only be used in a user accounting routine.
- v is a volume number. It specifies, together with the DNAM parameter, a DAD to be temporarily assigned for this Job. The volume number consists of 4 hexadecimal characters.
- dn is a DAD name. It specifies the name of the DAD where the specified userid is located. This DAD is assigned to the filecode given in the DAD parameter or, if the DAD parameter is omitted to the first free DAD filecode, from /F1 to /FF.

The :JOB command must be the first BCL command for each background machine Job. It must thus also be the first BCL command read from the filecode /EO when the background machine is started with the SB operator command. When a :JOB command is given in a session without a previous :EOJ command, the BCP first generates a :EOJ command and then processes the :JOB command. The :JOB command causes the BCP to reinitialise the background machine control tables (to the values set up by MAS when the background machine was defined by the SCL DCB command, etc.), and to locate the user directory from the parameters USID and DAD on the :JOB command.

If the ACNT and PASW parameters are specified, the BCP will pass these to the routines USREOJ (unless it is the first :JOB command in the batch, or if the previous command was :EOJ) and USRJOB. If the dummy system-supplied USRJOB program has been replaced by a User accounting routine, then this routine can, upon return to the BCP, signal whether the Job is to be processed or bypassed. In the latter case, the BCP will ignore all remaining BCL commands until the next :JOB or :EOB command. The reinitialising of the background machine filecode tables whenever a :JOB command is read is necessary, since the previous Job may have modified them by:

- The BCL ERR and :STP commands set the background machine operation mode.
- The BCL ASG and ASD commands can override the filecodes defined at DCB time.

The following error messages may be output by the BCP if the :JOB command is incorrect:

PARAM xxxxx MISSING
GIVE A DAD FILECODE

No free DAD filecode could be found to assign a DAD dynamically. The user should specify a filecode in the DAD parameter, overriding the original filecode given during the SCL DCB session.

WRONG VOLUME NUMBER

The VOLN value consists of more than 4 characters, or a disc with the specified volume number has not been mounted.

INVALID PARAM=xxxx

BAD ASSIGN STATUS=xxxx

The LKM 71 (assign DAD) returned an error status. See Appendix C for the meaning of the status.

I/O ERROR ON DISC

An I/O error occurred during the DAD assignment (LKM 71)

SYSTEM DYNAMIC AREA OVERFLOW

No space could be obtained, to locate the tables for the DAD to be assigned.

DAD NAME NOT FOUND

/FO CANNOT BE REASSIGNED

DNAM and VOLN were given with a DAD parameter value /FO

TOO MANY FILECODES

ERROR READ DAD, STATUS=xxxx

An I/O error occurred, reading the Catalog of the DAD to be assigned..

PARAM USID MISSING

USID NAME UNKNOWN

Error messages output by the command analyser can be found at the end of this chapter.

Note: The semicolon (;) must be the first character on the JOB command and must not occur again within the body of the command.

:STP

Initiate Closed-shop Mode

:STP

Format 1 :STP [CODE=n][,ABCD=c][,NCOD=p]

Format 2 :STP [n],[c],[p]

- n is a numeric value from 0 to /7F, defining the highest BCL error code allowed in this Step. If omitted, 0 is assumed. The n must be higher than the BCL error code from the preceeding step.
- c is a numeric value from 0 to /7F, defining the value that BCP is to assign to the error code if any program executed in the background machine between the occurrence of this :STP command and the next one (or the next :EOB, :EOJ or :JOB command) is aborted. If omitted, aborts cause the error code to be set to /7F.
- p is a numeric value from 0 to /7F, defining the new highest BCL error code allowed in this step. With the NCOD parameter, the current BCL error code can be decreased. When the current error code is lower than the NCOD value, the NCOD is ignored.

The :STP command places the background machine into closed-shop mode, in which a Job consists of one or more Steps, each starting with a :STP command. BCL commands occurring before the first :STP command in a Job are regarded by the BCP as being in a Step having a maximum error code value of 0.

During processing, if a BCL command results in the error code being set to a value exceeding the maximum error code value for the Step, all subsequent BCL commands for the Step will be ignored.

A Step is terminated either by another :STP command, or by any of the following BCL commands: :JOB, :EOJ or :EOB.

The following error message may be output by the BCP if the :STP command is invalid:

INVALID PARAM= xxxx

Note: The :STP command must start in column 1, otherwise it may be read as data by the processor or by a program.

BCL Filecode Allocation Commands

Default allocation

Permanent filecodes for the background machine are allocated by SCL FCD commands, entered by the system manager when defining the background machine and prior to the operator starting it with the SB operator command. These assignments can, however, be modified by background machine Jobs. Such modifications only have effect for the Job in which they are made, since the BCP resets the background machine device assignments back to their machine definition values whenever a new Job is started.

Whenever a :JOB command is read by the BCP, the filecode table is initialised to contain only the filecodes declared at machine declaration time.

A Job modifies its job filecode from the default background machine device assignments by:

- BCL ASD commands
- BCL ASG commands
- BCL SCR commands
- BCL REQ commands
- BCL REL commands
- LKM 23 monitor calls
- LKM 24 monitor calls
- LKM 33 monitor calls
- LKM 54 monitor calls
- LKM 71 monitor calls.

Only modifications by BCL commands are considered in this section. LKM (Link to Monitor) requests are dealt with in Appendix C.

Note:

A permanent filecode cannot be re-assigned to be equivalent to a temporary filecode.

Format 1 ASD FCOD=f,VOLN=v,DNAM=d
Format 2 ASD FCOD=f,DISK=c,DNAM=d
Format 3 ASD f,{c|v},d

- f is a DAD filecode from /F1 to /FF. This filecode is assigned to the DAD specified by the VOLN or DISK and DNAM parameters. f may be an existing DAD filecode, but not /FO.
- v is the volume number of the disk where the DAD to be assigned resides. It must be up to 4 hexadecimal characters.
- c is the filecode of the disk (from /CO to /CF) where the DAD to be assigned resides.
- d is a DAD-name, up to 6 characters. It is the name of the DAD to be assigned.

With the ASD command, the user may assign a DAD, not yet declared during machine declaration. The filecode to be assigned may be free or occupied, but an already assigned filecode cannot be reassigned if DFM or TDFM files are assigned to it. Also the Job DAD filecode cannot be reassigned. Although the disk can be specified via the volume number, its disc filecode (/CO-/CF) must be contained in the filecode table of the Background machine. The temporary DAD filecodes are scratched or reset to their original assignments on detection of a :JOB, :EOJ or :EOB command.

The following error messages may be output due to an erroneous command input, or to a status returned from the executed LKM 71 (assign DAD):

I/O ERROR ON DISC

Error detected during read of the DAD's catalogue

SYSTEM DYNAMIC AREA OVERFLOW

No space to allocate the tables for the DAD to be assigned.

DAD NAME NOT FOUND

The DAD does not exist on the disc indicated by the volume number or the disc filecode

WRONG VOLUME NUMBER

VOLN parameter value contains more than 4 characters or is not found.

DISC F.C. OR DAD F.C. MUST BE /Cx OR /Fx

DISC F.C. NOT ASSIGNED

The filecode specified for the disc has not been specified during machine definition.

/FO OR DAD OF THE CURRENT JOB CANNOT BE REASSIGNED

TOO MANY FILECODES

FCOD VALUE NOT ALLOWED

The FCOD parameter value is not a DAD filecode.

GIVE DISC VALUE FROM /CO TO /CF

VOLN OR DISC PARAMETER MISSING

VOLN OR DISC PARAMETER REDUNDANT

BAD ASSIGN STATUS=xxxx

The LKM 71 returned an error status. For the meaning of that status see Appendix C.

DAD F.C. CANNOT BE REASSIGNED, A DFM OR TDFM FILE IS ASSIGNED TO IT
 INVALID PARAM=xxxx

There are four types of ASG command which may be used in the BCL.

Assign a filecode to a physical non-disc device.

Format 1a ASG FCOD=f,DVCE=n[a[1]]

Format 1b ASG f,n[a[1]]

- f is a filecode to be assigned
- n is the device type (a 2 character code); see Appendix A.
- a is the device address (two hexadecimal digits without a preceding /). If omitted, the first device in the SYSGEN list for the device is assumed. This parameter should be omitted if n = NO.
- l For devices connected to an AMA8 controller, a linenumber from 0 to 7 can be specified. There are two ways to indicate an AMA8 linenumber: The linenumber can be added to the device address, so a+l ,or it can be specified separatily.
Example: an assign of a display connected to an AMA8, device address /18 linenumber 1 can be done as: ASG 2,DY19 or
ASG 2,DY181.

Assign a filecode to a disc temporary file

Format 2a ASG FCOD=f[,DAD=d][,TYPE=t][,NBGR=n][,CONS={YES | NO}]

Format 2b ASGT f,[d],[t],[n],[YES|NO]

- f is the filecode to be assigned.
- d is a DAD filecode (default is the Job DAD).
- t is the filetype code (default = UF).
- n is the number of granules (default = 1).
- CONS specifies whether the granules are consecutive (YES) or not (NO). The default is NO.

Note: If the file is to be created by Direct Write LKM 1, the number of granules (consecutive or non-consecutive) must be allocated at Assign time.

For example, if a file has 8 sectors per granule, and only 1 granule is allocated at Assign time, then if a Direct LKM 1 is given to write sector 23, it will be rejected with status /10.

Assign a filecode to a catalogued file

Format 3a

ASG FCOD=f,FNAM=n[,DAD=d][,TYPE=t][,USID=u][,WPRO={YES | NO}][,VERS=v]

Format 3b

ASGF f,n,[t],[u],[d],[YES|NO],[v]

- f is the filecode to be assigned
- d is the DAD filecode whose catalogue contains the filename (default is the Job DAD).
- n is the filename.
- t is the filetype code (default = UF). One of the codes SC, OB, LM, UF and EF.
- u is the userid of the Directory (default = :JOB Userid).
- v is the version number (a digit from 0 to 7, but not more than the value on the SMV command to LIB specified for this Userid). The default value is 0.

WPRO is used to set (YES) or leave alone (NO) the write protect flag for this file in the MAS tables. The default is NO. The flag is not updated in the directory on disc - this can only be done by the Processor LIB with SPF or RPF control commands - but only in the MAS memory tables. Specifying WPRO=YES is only required if the write protect flag was not set ON when the file was created; nevertheless this background machine job requires the file to be protected from being written to by any foreground machine task until the background machine Job is finished.

Assign a filecode to another filecode

Format 4a ASG FCOD=f,ECOD=e

Format 4b ASGE f,e

f is the filecode to be assigned.
e is the filecode which has been assigned.

Note: filecode 'e' must not itself have been assigned to another filecode. This is to prevent problems if filecode 'e' is scratched.

Filecode 'f' may be a permanent filecode. If filecode 'e' is subsequently reassigned, filecode 'f' is also reassigned.

The following error messages may be output by the BCP if the ASG command is rejected:

PARAM xxxx MISSING
INVALID PARAM=xxxx
BAD ASSIGN STATUS=xxxx

The LKM 23 (assign) which is used for the execution of this command, returned a status. For an explanation of the status see Appendix C.

DEVICE NAME UNKNOWN
I/O ERROR ON DISC

I/O error, reading the Catalog of the DAD, the directory of the userid or the GRANTB of the file to be assigned.

SYSTEM DYNAMIC AREA OVERFLOW

No space to allocate the tables to assign the filecode

NO SPACE FOR FILE DESCRIPTION TABLE
DEVICE OR DAD UNKNOWN
NO SPACE ON DISC

There are not enough free granules in the DAD to assign the temporary file
UNKNOWN FILE NAME

The catalogued file to be assigned does not exist in the directory of the specified userid.

ECOD NOT ASSIGNED
INVALID ASSIGN TYPE

The specified ASG parameters caused the BCP to construct an illegal control block for the LKM 23.

USID IS UNKNOWN

The specified userid did not exist in the DAD Catalog

ILLEGAL FILE TYPE

The file type was not UF, SC, LM, OB or EF.

INVALID FILE CODE

A DAD filecode was not /FO-/FF or a filecode was not /OO-/FF.

DAD FCOD NOT ASSIGNED
PERMANENT FILECODE CANNOT BE REASSIGNED BY EQUIVALENCE
BAD ASSIGN BLOCK ADDRESS

This message can never be caused by the user. It is a Hardware or BCP error.

TOO MANY FILECODES
BAD DEVICE ADDRESS
UNSHARED FILE

The catalogued file that was to be assigned, was not shareable and the user, trying to assign it was not the owner, nor SYSTEM.

TOO MANY GRANULES FOR THE SECTOR SIZE

The NBGR parameter value was more than (DAD sectorlength-10)/2.

BAD LINENUMBER FOR AMA8

In the Format 1 assign, the l parameter was not 0-7.

Additional error messages for the assign of TDFM files:

USER IDENT UNKNOWN FOR ONE SUBFILE

NAME UNKNOWN FOR ONE SUBFILE

DYNAMIC AREA OVERFLOW IN SYSTEM MACHINE

ONE PACK NOT ON LINE

THE DAD OF ONE SUBFILE IS UNKNOWN

THE DAD OF ONE SUBFILE IS NOT ASSIGNED

THE FILECODE OF THE DAD OF THE DESCRIPTOR IS UNKNOWN

FILE ALREADY ASG. IN OTHER MACH. WITH NON IDENTICAL DAD FILECODE

NO FCT ENTRY FOR DISC OF DESCRIPTOR

NO FCT ENTRY FOR DAD OF DESCRIPTOR

FILE CODE WAS USED BY RUNNING TRANSAC. FOR ANOTHER EDFM FILE.

Assigning TDFM files, in the ASG command, only the identification of the descriptor file has to be given. In the descriptor file MAS finds references to KEY and DATA files. Most of the above mentioned error messages are caused by the fact that these references call discs (via the volume number), DADs or userids, which are not known or assigned in the Background machine.

NOTE:

If the syntax of the assign command is incorrect, the specified f can be deleted.

Format 1 SCR FCOD=f
Format 2 SCR f

f is the filecode to be removed from the MAS tables.

The SCR command is used to remove a filecode from the MAS filecode table for the current background machine Job. It is only necessary to do this if the current Job is likely to cause an overflow of the filecode table, because it has a large number of filecode assigns, and will exceed the SCL MFC limit, or the system generation default.

It is also used if a new object file is to be created (processors only extend it) or if filecodes have been assigned to a DAD that has to be reassigned. Certain filecodes used by the BCP to process the BCL commands for the Job (for example filecodes /E0, /O2) cannot be specified in the SCR command.

If a magnetic tape or cassette was assigned by a BCL REQ command, it should be scratched by a BCL REL command in order to:

- Scratch the filecode
- Tell the operator to dismount the tape
- Release the device attached by the REQ command.

LKM 54 may be used instead of the REQ and REL commands, if preferred.

The following error messages may be output by the BCP if the SCR command is rejected:

FILE CODE NOT ALLOWED

FILE CODE NOT FOUND

COM. EXECUTED, BUT F.C. WAS USED BY STILL RUNNING TRANSAC FOR A TDFM FILE

BCL Tape and Cassette Commands

The background user may use these devices by performing the following sequence of commands:

- Give a BCL REQ command (to attach the device, assign a filecode to it, and instruct the operator to mount it).
- Give commands to position the tape or cassette, and to label it as required. Commands to either the BCP or LIB may be used; they have the same format.
- Run the program which uses the tape (via LKM 1 or LIB).
- Give commands to the BCP or to LIB, to write EOY or EOF and to reposition the tape as necessary, these can also be done via LKM 1.
- Give a BCL REL command to detach the device, scratch the filecode, and instruct the operator to dismount the tape.

Format 1 FBS FCOD=fc[,NUMB=n]
Format 2 FBS fc[,n]

fc is a filecode assigned to an MT or TK device
n is the number of tape marks to be back-spaced.

The tape will be positioned before the nth tape mark skipped. If the NUMB parameter is omitted, NUMB = 1 is assumed.

One of the following error messages is output if an FBS command is rejected:-
NO TAPE MARK ON M.T. (begin of tape is found)
I/O ERR ON M.T. (FCOD=xy), STATUS=abcd (see Appendix C, LKM 1)

Format 1 FFS FCOD=fc[,NUMB=(ALL | n)]
Format 2 FFS fc[,ALL|n]

fc is a filecode assigned to an MT or TK device.

n is the number of subsequent tape marks to be skipped over. The tape will be positioned after this tape mark.

ALL The tape will be forward-spaced until two consecutive tape marks are found and positioned after the first of these two.

If the NUMB parameter is omitted, NUMB=1 is assumed.

One of the following error messages is output if an FFS command is rejected:

PARAM ERR

NO TAPE MARK ON THE TAPE

KEY PARAM TOO LONG

I/O ERR ON M.T. (FCOD=xy), STATUS=abcd

Format 1 PLB FCOD=fc Format 2 PLB fc

fc is a filecode assigned to an MT or TK device.

The volume label of the MT or TK will be printed on the operator console and the tape will be positioned at the first record of the file following the label. If the tape has no volume label, the message:

NO VOLUME LABEL (first record does not start with 'VOL1')
is output, and the tape is backspaced one block.

The following error message is output if a PLB command is rejected:
I/O ERROR ON M.T. (FCOD=xy), STATUS=abcd

Format 1 RBS FCOD=fc[,NUMB=n]

Format 2 RBS fc[,n]

fc is a filecode assigned to an MT or TK device.

n is the number of records to be skipped.

The tape will be positioned before the nth previous record. If the NUMB parameter is omitted, NUMB = 1 is assumed.

One of the following error messages is output if an RBS command is rejected:

BEGINNING OF TAPE IS ENCOUNTERED

EOF ENCOUNTERED

I/O ERR ON M.T. (FCOD=xy), STATUS=abcd

EOS ENCOUNTERED

REF

Rewind File

REF

Format 1 REF FCOD=fc

Format 2 REF fc

fc is the filecode.

The file is rewound and positioned at the first record.

The following error message may be output:

**** I/O ERROR. (FCOD=xy), STATUS abcd

Format 1 REL FCOD=fc,MESS='m'
Format 2 REL fc,'m'

fc is a background machine filecode mentioned on a previous REQ command.
m is a message to be sent to the operator on filecode /EF of the system machine, maximal 72 charcaters.

The system will detach the device and send the operator the message:
DISMOUNT dn da m ,PLEASE

where

dn = device name code
da = device address
m = message on REL command.

One of the following error messages will be sent to the background machine user on filecode /01 of the background machine, if a REL command is rejected:

DISK DEVICE OR DEVICE NOT ATTACHED TO THE PROGRAM
FILE CODE NOT ASSIGNED
NOT ASSIGNED TO A PHYS. DEV. OR ASSIGN TO NO DEVICE

Format 1 REQ DVCE=d,FCOD=fc,MESS='m'
Format 2 REQ d,fc,'m'

- d is a two-character device-name code, either MT or TK. An operational device of this type will be attached (reserved exclusively) to the Job.
- fc is the filecode to be assigned to the device; two hexadecimal characters, preceded by the / character.
- m is a message to be sent to the operator on filecode /EF of the system machine, maximal 72 characters.

The REQ command is used by the background user to perform the same functions as LKM 14 (to request the exclusive use of a device), or LKM 54 (REQ or REL).

The background user may not know the device addresses, and even if he does he will not necessarily know which devices are being used by the foreground machines at any moment. The ASG command cannot in these cases be used by the background user, and the REQ command is available for such circumstances.

The system will:

- Allocate an operational device (one not mentioned on an 'OF' operator command or an SCL DOF command from the system manager) of the requested type to the batch machine.
- Attach it to the Job.
- Assign the specified filecode to it.
- Send the message:
 MOUNT ON dn da 'm' THEN RS, PLEASE
 to the operator on filecode /EF of the system machine, where:
 dn = device type on REQ command
 m = message on REQ command
 da = the device address allocated.
- Enter the pause state until the operator enters the RS (Restart) command for the background machine.

One of the following error messages will be output to the background user on filecode /01 of the background machine, if a REQ command is rejected:

```
NOT ASSIGNED
DYNAMIC AREA OVERFLOW
INVALID DEVICE NAME (ONLY MT OR TK)
NO FREE AND OPERABLE DEVICE
BAD ASSIGN STATUS = abcd
```

Format 1 REW FCOD=fc
Format 2 REW fc

fc is a filecode assigned to an MT or TK device.

The tape is rewound to the load point.

One of the following error messages is output if a REW command is rejected:

FILE CODE MISSING

FILE CODE ERROR

FILE CODE UNKNOWN

I/O ERR ON M.T. (FCOD=xy), STATUS=abcd

ERR ENCOUNTERED IN REWIND (the LKM 1 order rewind returned a positive status, not equal to load point detected)

Format 1 RFS FCOD=fc[,NUMB=n]

Format 2 RFS fc[,n]

fc is a filecode assigned to an MT or TK device.

n is the number of records to be skipped. If the NUMB parameter is omitted, NUMB = 1 is assumed.

The tape will be positioned after the nth subsequent record.

One of the following error messages is output if an RFS command is rejected:

END OF TAPE IS ENCOUNTERED

EOF ENCOUNTERED

I/O ERR ON M.T. (FCOD=xy), STATUS=abcd

EOS ENCOUNTERED

EOV MARK ENCOUNTERED

NO DATA ON TAPE

Format 1 ULD FCOD=fc

Format 2 ULD fc

fc is a filecode assigned to an MT or TK device.

The tape device is switched to the manual state by sending a 'switch off' command.

One of the following error messages is output if a ULD command is rejected:
THE FUNCTION HAS NOT WORK CORRECTLY (FCOD=xy) STATUS=abcd

Format 1 WEF FCOD=fc[,NUMB=n]

Format 2 WEF fc[,n]

fc is a filecode assigned to an MT or TK device.

n is the number of EOF marks to be written. If the parameter NUMB is omitted, NUMB = 1 is assumed.

The following error message may be output:
I/O ERR ON M.T. (FCOD=xy), STATUS=abcd

Format 1 WES FCOD=fc[,NUMB=n]

Format 2 WES fc[,n]

fc is a filecode assigned to an MT or TK device.

n is the number of EOS marks to be written. If the parameter NUMB is omitted, NUMB = 1 is assumed.

The following error message may be output:
I/O ERR ON M.T (FCOD=xy), STATUS=abcd

WEV

Write EOv on MT or TK

WEV

Format 1 WEV FCOD=fc

Format 2 WEV fc

fc is a filecode assigned to an MT or TK device.

An EOv is written on the tape.

The following error message is output if a WEV command is rejected:
I/O ERR ON M.T. (FCOD=xy), STATUS=abcd

An MT or TK label is 80 ASCII characters, formatted as follows:

CHARS	CONTENTS
0-3	VOL1
4-9	volume serial number
10	security code
11-40	reserved
41-79	owner code.

The tape label is followed by a Tape Mark.

Format 1 WLB FCOD=fc[,SNUM=`sn`][,SCOD=`sc`][,OWNE=`ow`]

Format 2 WLB fc[,['sn'],['sc'],['ow']]

fc is a filecode assigned to an MT or TK device.

sn is the volume serial number (1-6 ASCII characters), default blanks.

sc is the security code (1 ASCII character), default 0.

ow is the owner code (1 or more ASCII characters). If more than 39 characters are entered, only the first 39 will be accepted, default blanks.

One of the following error messages is output if a WLB command is rejected:

PARAM ---- TOO LONG

I/O ERR ON M.T. (FCOD=xy), STATUS=abcd

BCL Interactive Commands

Introduction

Four BCL commands are available which permit interactive intervention; these are the ROI, ERR, PSE and MES commands. The ERR command is described in the section "BCL Scheduling Commands"; it permits the background user to correct interactively a BCL command which has been rejected by the BCP.

MES

Output Message

MES

Format MES n

n is a message to be sent to the operator.

The MES command causes the BCP to output the specified message on the system machine device /EF. Having done this, the BCP reads the next BCL command from the device defined by background machine filecode /EO and continues. The MES command could be used by the background user to inform the operator that he has finished using special paper, for example.

Format PSE n

n is a message to be sent to the operator.

The pause command causes the BCP to output the specified message on the system machine device and then place the background machine into the wait state. The RS operator command must be entered to restart the background machine.

The PSE command could be given, for example, by a background user to request the operator to mount special paper and ready the printer. It complements the facilities available with LKM 6 and LKM 54.

Format 1 ROI FCOD=fc,MESS=`m`
Format 2 ROI fc,`m`

fc is the filecode.

m is a message to be sent to the operator, maximal 72 characters.

The operator is sent the following message:

m ON dn,da THEN RS, PLEASE

where:

`m` is the message as specified on the ROI command

`dn` is the device name

`da` is the device address.

The background machine is placed in the pause state until the RS operator command is entered.

BCL Task Initiation Commands

How the BCP Initiates Tasks

There are 3 BCL commands available for initiating tasks in the background machine. These are:

- Standard Processor Call
- Non-Standard Processor Call
- RUN command.

The processing performed by the BCP on these BCL commands consists of four operations:

1. The BCL command is analysed for correct syntax and rejected if invalid.
2. The location of the program to be executed is calculated. In other words, the DADcode, Userid and the LM file (name, type, version) are identified.
3. The Job Parameter Table is initialised. This table contains the information required by MAS to load the program (calculated in 2 above) and other information about the program (e.g how many lines of print it may output and, for a swappable background machine, for how many seconds it must be resident).
4. The BCP then issues an LKM 3 (Exit) Monitor request with A8 pointing to the JPT (Job Parameter Table). This causes MAS to replace the BCP by the required program which will then be executed.

Task Initiation Commands

Standard Processor Call (See Volume III of this set).

Format 1

pname [SIZE={MAX | n}][,DUMP={ALL | PROG | NO}]

Format 2

pname [MAX|n][,PROG|ALL|NO]

pname is one of the Standard Processor Names. These are the names of the Standard Processors supplied with MAS, for example:
FRT Fortran Compiler
ASM Assembler
RTL RTL/2 compiler
LKE Linkage Editor
UPD Update
LIB Librarian
EDF Extended Disc File Processor
MAC Macroprocessor.

n is only required when the background machine was defined as swappable with SCL DCB command, and when the processor assumes it may use virtual addresses larger than that for the last word in the last page containing the load module (or issues any LKM 4). If both conditions are true, n specifies the number of pages to be allocated additional to those required to contain the load module. The default is 0. If the background machine is memory-resident, the SIZE parameter is ignored, and the Processor is allocated the pages reserved for the background machine by SCL DCB command. If so, it should ensure that it does not issue LKM4 as well as assume it can address pages beyond the end of the load module directly.

MAX MAS will reserve 32K words for the Processor.

DUMP is used to request a postmortem dump in the event of a Processor Abort, or if the Processor issues LKM 3 (Exit) with bit 8 of the exit code set (i.e. Exit with postmortem dump). The parameter is ignored if the postmortem dump option was not selected at SYSGEN.

ALL requests a dump of both the system machine and the background machine.

PROG requests a dump of the background machine only.
If NO is specified, or if the DUMP parameter is omitted, no
postmortem dump will be produced.

For a Standard Processor Call the BCP evaluates the program location as:

DAD = /FO

Userid = the first user in the /FO Catalogue.

The Processors must therefore be contained in the Directory for the first User
in the Catalogue for the DAD /FO.

Fortran Diagnostic Error Messages

To obtain full documentation about Fortran runtime errors, the parameter
D(debug)={YES | FULL} can be specified in the OPT card. With this parameter,
Fortran maintains a chain of blocks to print if need be an error message with
full documentation (error number, statement number, called at).

To print a full error message correctly, all Fortran programs, main program and
subroutines, should be compiled with D=YES or D=FULL.

Problems occur using Fortran routines called by Assembler routines. The
diagnostic error message is incorrect when the main program is an Assembler
program, because Fortran does not know where the start of the chain is. No
problems occur when using Assembler subroutines called from Fortran.

To overcome problems with Assembler main programs, two solutions exist:

1. Let the Assembler program be called by a dummy Fortran main program,
consisting of a CALL of the Assembler program, and compiled with D=YES
or D=FULL.

2. Insert in the Assembler main program the following instructions:

```
EXTRN      W:DGER,W:ERLK  
and  
LDKL      A1,W:DGER  
ST        A1,W:ERLK+2
```

Using one of these alternatives, the full diagnostic error messages are printed
correctly for non re-entrant code. When re-entrant code must be generated,
only the first method can be used.

Non-Standard Processor Call

A Non-Standard Processor Call is used, either when the Processor name is not one of the standard processor names or the Processor is not contained in the Directory for the first user in the Catalogue for the DAD /FO.

Format 1 pname [SIZE={MAX|n}][,DUMP={ALL|PROG|NO}][,USID=u][,DAD=d]
Format 2 pname [[MAX|n],[ALL|PROG|NO],[u],[d]

pname is any name of 1-6 ASCII characters except a BCL command mnemonic. In general BCL commands either have a 3 character name or commence with : (colon). The user can thus easily ensure that no non-standard processor has the same name as a BCL command (which would cause the BCP giving an error if it happened).

DUMP Identical to a Standard Processor Call.
SIZE Identical to the Standard Processor Call.
u the userid (1-8 characters).
d the DAD filecode (/FO - /FF).

For a Non-Standard Processor Call the BCP evaluates the program location according to whether the parameters USID and DAD are, or are not, specified in the call:

IS DAD SPECIFIED?	Y	Y	Y	N	N	N	N
IS USID SPECIFIED?	Y	N	N	Y	N	N	N
WAS USID ON BCL :JOB?		Y	N		Y	Y	N
WAS DAD ON BCL :JOB?					Y	N	
SET DAD TO SPECIFIED VALUE	*	*	*				
SET DAD TO :JOB VALUE			*				
SET DAD TO VALUE COMPUTED FOR JOB						*	
SEARCH ALL DADS FOR USID				*			
SET DAD = /FO							*
SET USID TO SPECIFIED VALUE	*		*				
SET USID TO :JOB VALUE		*		*	*	*	
SET USID TO 1ST USER IN DAD		*				*	

The following error messages may be output by the BCP if a Standard or Non-Standard Processor Call is invalid:

PARAM. NOT VALID
PROCESSOR NOT CATALOGUED
BAD ASSIGN
I/O ERROR ON DISK
SEARCH DIRECTORY NOT POSSIBLE
PARAM.... MISSING
INVALID PARAM=
BAD COMMAND NAME (the command name contained more than 6 characters)

Format 1 RUN optional-keyword-parameters
Format 2 RUN optional positional parameters in the order as occurring below

The optional keyword parameters are:-

KEYWORD	DEFAULT	SPECIFICATION RULES
PROG=	/D6	Up to 6 ASCII chars specifying the name of the program to be executed. It must be the name of an LM file. If omitted, filecode /D6 is assumed to define the program, and is assigned by the LKE to its output load module.
USID=	:JOB USID	Up to 8 ASCII characters, specifying the Directory in which the program is to be located.
DAD=	:JOB DAD	A DAD filecode (/F0 - /FF) defined by an SCL FCD command which specifies the catalogue in which the Directory is located.
VERS=	0	A digit from 0 to 7, specifying the version of the program to be executed. The value must not exceed the value specified on the SMV command to LIB for this Userid.
SIZE=	0	Either the word MAX or a number from 0 to 15, specifying the number of memory pages, additional to those containing the load module, which the program assumes it may address using virtual addresses or by giving one or more LKM 4 requests. This value is only required if the SCL
DUMP=	NO	DCB command is defined a swappable background machine. Only required if one desires a post-mortem dump in the event of the program aborting. Specify ALL to dump the system and the background machine. Specify PROG to dump the background machine only. The parameter is ignored if the post-mortem dump option was not selected at SYSGEN.
PNCH=	1000	Either the maximum number of records which may be punched by this program, or NO if there is no limit.
PRNT=	1000	Either the maximum number of lines which may be printed by this program, or NO if there is no limit.
TIME=	300	Either the maximum number of seconds for which the program may run in the background machine, or NO if there is no limit. The operator should not set the clock off if a value is specified, or it will be ignored.
FR1=,FR3=	0	The values to be placed in the floating point registers FR1 - FR3 when the program is started.

If, during the execution of this program, an attempt is made to exceed one of the three limits PNCH, PRNT or TIME, the program will be aborted.

The default value for PROG is the program output by the last execution of the Linkage Editor (LKE Processor) in the same Job, to the Load Module file (/D6).

The following error message may be output by the BCP if the RUN command is rejected:

```
INVALID PARAM.=....
BAD ASSIGN
USID NAME UNKNOWN
PROG. NAME UNKNOWN
SEARCH CATAL. NOT POSSIBLE
SEARCH DIRECTORY NOT POSSIBLE
PROG. NAME MISSING
I/O ERROR ON DISK
FILE /D6 NOT ASSIGNED
```

The INC command can be used to prepare an Object file for the LKE Processor. There are two types of INC command, depending on whether the modules to be added to the Object file are to be read by the BCP from a non-disc device or from a disc device. The object modules will be output to the disc temporary object library (filecode /D5). If this object library is not assigned, or has been closed (an EOF mark has been written), the BCP will automatically create a new one in the DAD whose catalogue contains a pointer to the Directory for the Userid specified on the BCL :JOB command for this background machine job. The relevant DAD will have been found by the BCP when the BCL :JOB command was processed - either because the DAD was also specified on the BCL :JOB command, or because the BCP searched all the DADs (defined by SCL FCD commands when the background machine was defined) until it found the one whose catalogue contained the relevant Userid. The DADs are searched in the same order they were defined by SCL FCD commands.

INC from a disc device

In this case the INC command has a set of parameters which locate the library to be read by the BCP. It is possible to select an individual module from the library. Filecode /D0 will be used by the BCP to assign the file.

Format 1 INC LIBR=f[,MNAM=n|ALL][,USID=u][,DAD=d]

Format 2 INC f[,n|ALL],[u],[d]

- f is the library name (filename);
- n is the object module name, or ALL if all the object modules in the library are to be read by the BCP. Default is ALL.
- u identifies the Directory to be searched for the filename. If USID is not specified, the Userid from the BCL :JOB command will be assumed.
- d identifies the DAD catalogue to be searched for the user Directory. If it is not specified, the DAD specified on the :JOB will be assumed. (If no DAD was specified on the :JOB, the BCP will have identified it by searching all the DADs defined by SCL FCD commands, until it found the Userid.)

INC from a non-disc device

Format 1 INC [FCOD=fc]

Format 2 INC [fc]

- fc is the filecode which defines the device type (default=/E2) and address from which the BCP is to read the object modules. It can refer to a disc file. The BCP will read everything up to the EOF mark.

The following messages may be output by the BCP if the INC command is rejected:

PARAM xxxx MISSING
BAD ASSIGN
INV. FCOD
.....= UNNECESSARY PARAMETER
NO SPACE IN FILECODE TABLE
NO SPACE FOR FILE DESCRIPTION TABLE
NO SPACE ON DISK
UNKNOWN FILE NAME
DAD UNKNOWN
USID NAME UNKNOWN
UNKNOWN LIBR NAME
ERR. ASSIGN FILE /DO (ST=..)
NOT AN OBJECT FILE
ERROR READ/WRITE (STT=...)FILE=...
UNKNOWN MOD. NAME
ERROR IN DIRECTORY OBJ.
DAD FCOD NOT ASSIGNED
TOO MANY FILECODES

Format NOD nodename[,{rovs | *}[,a]]

nodename is a 6 ASCII character name, conforming to 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 eventually by the Linkage Editor. It defines a branch in an overlay tree.

rovs 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.

* is a signal to the Linkage Editor to implicitly name the segments in this compilation in ascending order, in accordance with the parameters on the LKE option command.

a is an absolute address in hexadecimal and, if present, it forces the loading of the module at this page boundary address.

The BCP merely copies this command into the file defined by filecode /D5 (i.e. the object temporary file). It will be processed later by the Standard Processor LKE.

The following error messages may be output by the BCP if the NOD command is invalid:

BAD ASSIGN
ERROR. ASSIGN FILE /DO (ST=...)
PARAM MISSING
PARAM MUST BEGIN WITH ALPHA CHAR

Format 1 HLP [CMND=c][,FCOD=fc]

Format 2 HLP [[c],[fc]]

c is a BCP command name. All keyword parameters for this command are output to filecode fc . If omitted, the syntax of all BCP commands is printed. The keyword parameters are output in the order of the positional parameter input.

fc is the filecode where the commands are printed. Default is filecode 1.

Error messages:

INVALID COMMAND

PRINT FILECODE NOT VALID

Format 1 SKP [PAGE=n]

Format 2 SKP [n]

n is the number of pages to be skipped on the lineprinter (0-100). The default is 1 page.

The SKP command causes the BCP to output a number of skips to Top-of-Form on filecode 2. A negative number of pages or a number of pages greater than 100 is changed into 100.

BCL Datacommunication commands

Some BCL commands exist to assign, scratch and manage datacom devices. These commands are:

DAS assign a linecode
DDC disconnect a linecode
DDL delete a linecode
DHD halt and disconnect
DHL stop an exchange

DAS Assign a linecode DAS

Format 1 DAS LCOD=l,DVCE=d

Format 2 DAS l,d

l is a linecode from 1 to 255. It remains valid during the Job, just like the ASG command.

d If a binary is supplied, d is assumed to be a linecode and l is assigned to that linecode by equivalence. If not a binary number is supplied, d is assumed to be assumed to be a device-name-address with eventually a linenumber. Its lay-out is then: dn[da[ln]]

dn is a name of a datacom device. See Appendix A

da is its device address. If omitted, the first occurrence of dn is chosen.

ln is the linenumber, only for AMA8 and LSM16. If omitted, zero is assumed.

d may be NO, to assign the linecode to a dummy device.

Error messages

One of the following error messages may be output by the BCL:

DEVICE NAME UNKNOWN
BAD DEVICE ADDRESS
NO DYNAMIC AREA IN THE MACHINE
DEVICE UNKNOWN
SECOND LINECODE NOT ASSIGNED
INVALID LINECODE

Other BCL Datacommunication commands

Format 1 c lcod=1
Format 2 c 1

c is a command mnemonic: DDC, DDL, DHD or DHL.
l is an assigned linecode.

Depending on the command mnemonic , the linecode is deleted (DDL) or disconnected (DDC), there is a halt and disconnect issued on the line (DHD) or the exchange is stopped (DHL))

Error messages

DISCONNECT A LINE ERROR, LC=1, STATUS=abcd
For the DDC command, the disconnect (LKM 8 order /11) returned a status #0.
INVALID LINECODE
For the DDL command, the specified linecode was not assigned or is not a value between 0 and 256.
HALT AND DISCONNECT LINE ERROR LC=1, STATUS=abcd
For the DHD command, one of the issued LKMs-8 returned a status #0.
STOP AN EXCHANGE ERROR LC=1, STATUS=abcd
For the DHL command, one of the issued LKMs-8 returned a status #0.

