9 CCI Control Commands

Control commands start with a mnemonic of three letters, followed by one space, possibly followed by one or more parameters.
The parameters are positional and separated by commas.
No additional spaces are allowed.
When the commands are entered via the typewriter, each command must be terminated with **CR LF**
Some of the parameters have a fixed meaning:
<userid> is the user identification, a string of 1 to 8 alphanumeric characters, not starting with / (slash). The first character must be a letter.
<name> is a file name or object module name of 1 to 6 characters, not starting with / (slash), or a numeric character.
/<disc number> or /FX is one of the file codes F0 to FF (see File Codes).

Each time the system wants a new control command it will type out S: After that, the control command may be typed in by the user.
It is possible to add a comment statement after the last parameter of a command. It must be separated from the command by at least one space. On the following pages the control commands are listed in alphabetical order, according to their mnemonics.
They are followed by the processor calls, which are used in the same manner.

In the syntax descriptions, Backus Normal Form is used for the notation, i.e.:

| means: or

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

⌊ ⌋ means: alternative components; one of the items within these brackets must be selected: ⌊+|–⌋ 426 can mean +426 or –426.

<> means: these brackets contain a syntactic item.

␣ means: one space.

Command	Meaning	Page
ASG	Assign a File Code	82
ASM	Call Assembler	109
BYE	End of Session	83
DCU	Declare User	83
DEB	Call Debugging Package	110
DEL	Delete File	84
DLU	Delete User	84
DUF	Dump File	85
END	End of Catalogued Procedure	85
FBS	Space File Backwards	90
FFS	Space File Forward	90
FOR	Call Full Fortran Compiler	111
HSF	High- Speed Fortran	112
INC	Include an Object Module	86
JOB	Start Batch Processing	86
KPF	Keep File	87
LED	Call Line Editor	114
LIC	List Catalogue	88
LKE	Call Linkage Editor	113
LSD	List Directory	88
LSF	List File Code	88
LST	List File	89
MES	Send Message	91
MOV	Move a File	91
NOD	Define Node	92
OLE	Overlay Linkage Editor	116
PCH	Punch a File	93
PLB	Print label	90
PLD	Punch Load	94
POB	Punch Object	94
POD	Print Object Directory	95
PRC	Print Catalogue	98
PRD	Print Directory	98
PRT	Print File	99
PSE	Pause	100

Command	Meaning	Page
RBS	Space Record Backwards	90
RDA	Read Data	100
RDO	Read Object	101
RDS	Read Source	102
REF	Rewind File	90
REW	Rewind to Load Point	90
RFS	Space Record Forward	90
RSU	Replace Supervisor	103
RUN	Run a Program	103
SCR	Scratch	104
SDM	Save Disc onto Magnetic Tape	105
SEG	Define Segments	106
SKF	Skip Form	107
SVD	Save Disc onto another Disc	107
SVU	Save User Files	108
ULD	Unlock Device	90
UPR	User Processor	116
WES	Write Device	90
WEF	Write EOS	90
WEV	Write End-Of-Volume	90
WLB	Write Label	90

syntax ASG␣/< file code 1> [, /< file code 2> | ,< device name>][,< name> [,<userid>[,<disc number>[,NP][,NP]]]

use: This command is used to assign a file code to a peripheral unit, a disc file or a temporary area on disc.
The parameters have the following meaning:
<file code1>: file code which is to be assigned.
<file code2>: if this parameter is used, the assignment previously made for this file code has to be made for the first one (<file code1>) also. As a result the assignments for the two file codes specified will be equated.
<device name>: if this parameter is used, <file code1> is assigned to the peripheral unit specified here by two characters for the unit type and 2 hexadecimal digits for the address. If the device is the disc, only DK need be specified, without address.
<name>: this parameter is used only when DK is specified for <device name>. It specifies the name of the library file to which the file code must be assigned. If DK is used without this parameter, the file code will be assigned to a temporary disc area.
<userid>: this parameter is used only when <name> is specified. With <disc number>, it allows assigning a file code to a file in another user's library on the disc specified. The file will be set to write-protected, unless the parameter NP is specified, in which case it will not be protected, to allow writing on a file of a different <userid>.

If the file code to be assigned has already been assigned previously, the old assignment is deleted.

Note:
As mentioned previously, file codes 01 to 09 and D0 to FF are reserved for the system or have standard assignments. The following restrictions apply, however:

- file code 01 cannot be assigned
- file codes 02 to 09 can be assigned only to non-disc devices.
- file codes D0 to DF cannot be assigned.
- file codes E0 to EF cannot be assigned to a disc file.
- file codes F0 to FF cannot be assigned.

errors: FILE CODE ERROR (1st parameter)
2nd FILE CODE ERROR
DEVICE UNKNOWN
TOO MANY PARAM
DEVICE NAME MISSING (2nd parameter)
FILE CODE NOT ASSIGNED (2nd file code)
FCT OVERFLOW (file code table overflow)
FILE CODE ABSENT
FILE NAME ERROR
USERID ERROR
INVALID FILE CODE
USERID UNKNOWN
DEVICE NAME ERROR
DEVICE ADDRESS ERROR
I/O ERROR (encountered during a read/write to/from disc)
LFT OVERFLOW (disc logical file description table overflow)
FILE NAME UNKNOWN
DISK OVERFLOW (no free granule available to allocate to the temporary disc file)
TOO MANY FILE CODE EQU (more than 7 file codes have been assigned to the same disc file)

BYE END OF SESSION BYE

syntax: BYE␣[BYE[,<DNDA>]]

use: In *batch processing mode,* this command indicates the end of the job and the system looks for the following job; if the parameter BYE is also specified, the system will switch from batch processing to conversational mode. If, in this case, <DNDA> (device name + device address) is also specified, this becomes the new assignment for file code / E0.

In *conversational mode*, the user must give this command at the end of the session to indicate that he is leaving the system; the system is re-initialized and will again ask for identification in order to start a new session, unless the parameter BYE is also specified, in which case the system will switch to batch processing mode and automatically start reading the job control commands (on the card or punched tape reader).

DCU DECLARE USER DCU

syntax: DCU␣<userid>,/<disc number>

use: This command can only be used in a system session, i.e. when the user gives, at the start of the session, the user identification SYSTEM. Then, through this command, a new user identification is added to the Catalogue of the disc specified. A directory granule is allocated to this user and initialized with /FFFF. An entry for this user is filled in the Catalogue. The allocation table is updated.

errors: INVALID USERID (the user identification does not start with a letter)
USERID ABSENT (no parameter is given in the command)
INVALID FILE CODE (the disc number cannot be a disc file code, for it is not in the range from /F0 to /FF)
DISK FILE CODE ABSENT (the disc file code is not present in the command)
DISK NOT OPERATIONAL (the disc unit is not ready)
USERID ALREADY CATALOGUED (the userid specified has already been catalogued previously on the disc specified)
CATALOG OVERFLOW (too many userids have been catalogued on the disc specified)
DISK I/O ERROR (an I/O error has been detected during a read/write operation to/from disc)
DISK OVERFLOW (no free granule is available to be allocated to the userid directory)
TOO MANY PARAM (an illegal parameter follows the disc number)
COMMAND NOT ALLOWED (the current session is not a SYSTEM session)
DISK FILE CODE UNKNOWN (the specified disc file code has not been declared at SYSGEN or the generated system does not contain the disc specified).

DEL DELETE FILE DEL

syntax: DEL⊔[<name>|/OB][,[/S|/O|/L]]

use: This command is used to delete a file or object module from a library.
<name> indicates the name of the file or module.
/OB indicates that the whole object file of the library must be deleted. If /OB is used, /S, /O or /L may not be specified.
/S, /O and /L specify the type of file: source, object or load. When <name> is used as the first parameter and no second parameter is specified, the type of file is UF (user file). If a /S, /O or /L file is to be deleted, this must be specified in the second parameter.
When <name> is used with /S or /L, a check is made on the types source or load to find the file which is to be deleted.
When <name> is used with /O, <name> is considered as being an object module in the object library.
When DEL /O is given the user object directory OBDIR is regenerated (see POD command).

errors: PARAM ERROR
INVALID PARAMETER
MISSING PARAMETER
FILE NOT CATALOGUED
I/O ERROR
TOO MANY PARAM
ERROR ASSIGN
PROGRAM NOT CATALOGUED

DLU DELETE USER DLU

syntax: DLU⊔<userid>,/<disc number>

use: This command can only be used in a system session, i.e. when the user gives, at the start of the session, the user identification SYSTEM. By means of this command, the user specified is deleted from the disc specified.
<userid> specifies the user to be deleted.
/<disc number> gives the file code of the corresponding disc.
The corresponding entry in the Catalogue, the directory granule and all the granules of the library files are released and the allocation table of the disc is updated.
The DLU command may not be used to delete the first user on the disc with disc number /F0 (SYSTEM).

errors: COMMAND NOT ALLOWED (the current session is not a SYSTEM session)
USERID ERROR (the first parameter is not a userid)
USERID MISSING (no parameter is given)
DISK FILE CODE ERROR (the second parameter is not numeric)
DISK FILE CODE MISSING (no disc address specified in the command)
INVALID DISK FILE CODE (the value of the second parameter is not in the range from /F0 to /FF)
DISK FILE CODE UNKNOWN (the on-line system does not contain the specified disc)
DISK NOT OPERATIONAL (the disc unit is not ready)
TOO MANY PARAM (more than two parameters specified in the command)
USERID NOT CATALOGUED (the specified userid has not been catalogued on the disc specified)
I/O ERROR IN CATALOG (an I/O error has been detected during a read or write operation in the catalogue)
DISK I/O ERROR (an I/O error has been detected during the de-allocation of the user files).

DUF DUMP FILE DUF

syntax: DUF␣[/<file code>|/O|/L|<name>][,<sect.nb1>[,<sect.nb2>]]

use: This command is used to get a hexadecimal dump on the print unit of a library file or a file with a file code or a disc at physical level (/FX)
<file code>: file code of the file which must be dumped.
<name>: name of a library file which must be dumped. This is a user file of the type UF.
/O and /L cause dumping of the object and load files respectively. This is mainly useful for system debugging purposes.
It is also possible to get a selective dump by specifying two sector numbers, <sect.nb1> and <sect.nb2>, as the beginning and ending sectors of the dump.
A dump is made up to an EOF record, up to End-Of-Volume (last granule) or <sect.nb2>. The <sect.nb.> is a disc sector logical address.
Users must be careful about the position of the file after a DUF command. For a sequential file, they must rewind it before using it again. This means that the following command sequence will give an error:

```
ASM /S
DUF /O
LKE
```

After assembly, the sequential /O file is positioned to the next free sector of the file, but when a DUF command is executed, it is positioned to the last dumped sector, in random mode. This means that the EOF record written by the system when the LKE command is encountered does not immediately follow the object code and the result will not be correct. The command sequence should be as follows:

```
ASM /S      or:   ASM /S
LKE               KPF /O
DUF /O            DUF /O
                  LKE
```

The DUF command must normally be performed only after the completion of execution of a job step.

errors:
FILE NAME ERROR
FILE NAME MISSING
FILE CODE ERROR
INPUT FILE ASSIGN ERROR (followed by a message giving the reason for the error)
FILE CODE NOT ASSIGNED
DISK NOT OPERATIONAL
TOO MANY PARAM
PARAM ERROR (error in sector number)
I/O ERROR (an I/O error has been encountered while reading the disc file)
SECTOR DELETED

END END CATALOGUED PROCEDURE END

syntax: END

use: This command must be specified by the user at the end of a catalogued procedure to indicate its termination to the system (see page 23). This implies that the user cannot define a procedure named END, nor can he use any other CCI command name for a catalogued procedure.

INC INCLUDE OBJECT MODULE INC

syntax: INC␣[/OBJCT]<name>][,<userid>[,<disc number>]]

use: By means of this command it is possible to select an object module from the library of the current (<userid>) or another (<userid>,<disc number>) user and to copy it in the temporary file /O.
<name> is the name of the object module which is to be included.
<userid> is to be used only if this object module is to be searched in the library of a user other than the current one.
If /O (temporary object file) exists, but has not been terminated by an EOF yet, this module is written at the end of this file.
If /O exists and has already been terminated with an EOF, this file is lost and a new assignment is made for a new /O.
If /O does not exist, an assignment is made for it and this module will be the first one of this new /O.
When the module has been copied, no EOF is written. Thus, it is possible to use this command several times, mixed with RDO commands and language processor calls to build a /O file.

Note:
- If /OBJCT is specified, the whole object library will be copied onto /O.
- When filling the /O file, no RDO, ASM or FRT command must appear after an INC command.
  It may only be followed by another INC command.

errors:
MISSING PARAMETER
PARAM ERROR
ERROR ASSIGN
UNKNOWN USERID
NO OBJECT LIBRARY
INVALID NAME
PROGRAM NOT CATALOGUED
I/O ERROR

JOB START JOB JOB

syntax: JOB␣[<userid>|/<disc file code>,<userid>]

use: This command is used to define the beginning of a new batch.
When it is encountered, the system will automatically switch to batch processing mode and implicitly close the preceding session, if any.
If <userid> only is specified, the system will scan the catalogue of each on-line disc, starting with disc unit /F0, until it finds the user identification specified.
If /<disc file code>,<userid> is specified, the system will look for the user identification only on the disc of which the file code is specified.

KPF KEEP FILE KPF

syntax: KPF␣[/S|/O|/L|/<file code>][,<name>]

use: This command is used to keep a file or module, which has previously been created as temporary, in a library, i.e. to make this file or object module permanent.

/S, /O, /L specify the type of the file which is to be kept.

<file code> is the file code of the file which is to be kept.

<name> is the name which is to be given to this file in the library and which will be placed in the directory.

If the first parameter is /S, the file will be of the type source. If <name> is not specified, /S is assumed to contain a source program of which the name can be found in its IDENT statement. Otherwise the name specified is taken as the file name. In case a file of the same name and type already exists in the library, it is deleted.

If the first parameter is /L, <name> must be specified. An old file of the same name and type will be deleted, as with /S.

If the first parameter is <file code>, <name> must be specified. An old file with the same file code will be deleted. Of course, <file code> must apply to a file which has previously been created as temporary.

If the first parameter is /O, <name> is optional. If it is not specified, all object modules of the /O file must be kept in a library, otherwise only the module <name> will be kept. Keeping object modules implies copying them from the /O file onto the user object file. If any modules of the same name already are included in the existing object file, they will be deleted by the system setting the 'sector deleted' flag in the sectors containing these modules. If possible, the copying is done in the same physical area which the deleted modules occupied previously.

When KPF /O is given, the system generates a user object directory, the layout of which is described under the CCI command POD.
Note: Any NOD commands in the /O file will be ignored.

errors:
PARAM ERROR
INVALID PARAMETER
MISSING PARAMETER
DIRECTORY OVERFLOW
FILE EMPTY
I/O ERROR
IDENT MISSING
FILE CODE NOT ASSIGNED
FILE ALREADY CATALOGUED
MODULE UNKNOWN
DISK OVERFLOW
FILE OVERFLOW
ASSIGN ERROR

LIC LIST CATALOGUE LIC

syntax: LIC␣/<disc number>

use: This command can only be used in a system session, i.e. when the user gives, at the start of the session, the user identification SYSTEM. It provides for printing out the catalogue of the disc specified on the typewriter.
/<disc number> gives the file code of the disc of which the catalogue must be listed.

errors:
SYSTEM SESSION COMMAND
PARAM MISSING
PARAM ERROR
FILE CODE NOT ASSIGNED

LSD LIST DIRECTORY LSD

syntax: LSD␣[/OB]

use: This command provides for a listing of the directory of the user library on the typewriter.
If /OB is specified, only the names of the modules of the object file are listed.

errors:
PARAM ERROR
NO OBJECT FILE CATALOGUED

LSF LIST FILE CODES LSF

syntax: LSF

use: When this command is given, a list is output on file code 1 of all the assigned file codes and the devices corresponding to them.

LST LIST FILE LST

syntax: LST␣[<file code>|/S|/S,<name>|<name>][,<line nb1>[,<line nb2>]]

use: This command causes a listing of the specified disc file on the operator's typewriter. The file must be sequential and consist of ASCII records. If a record is longer than a print line it will be printed on several lines.
The file can be either:
- a catalogued source file: /S,<name>
- a catalogued user data file: <name>
- a temporary data file: <file code>
- the temporary source file: /S.

<line nb1>,<line nb2> if specified provides for a listing of the file from the first line number to the second line number specified.

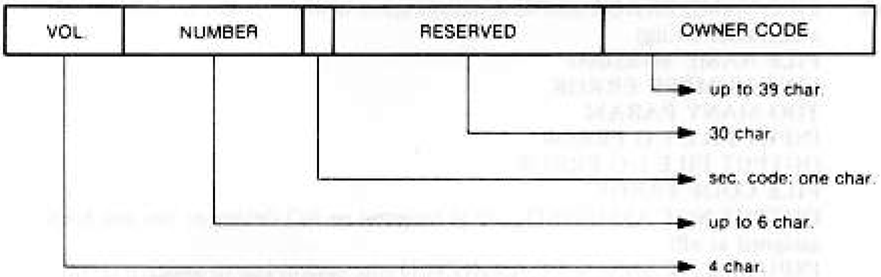
The maximum record size allowed is 80 characters. Records which are longer will be truncated. Non-printable record characters will be replaced by spaces and trailing blanks will be removed. The listing will stop when either <line nb2>, an EOF or End-Of-Volume (the last granule of the file) is reached. End-Of-Volume occurs when no EOF has been written for this file, which normally is the case for temporary files created by a program in the debugging phase. After the listing, the file is positioned at the last record listed.

errors: FILE NAME ERROR (the first parameter is neither /S nor a file code nor a character string)
FILE NAME MISSING
LINE NUMBER ERROR
TOO MANY PARAM
INPUT FILE I/O ERROR
OUTPUT FILE I/O ERROR
FILE CODE ERROR
OUTPUT NOT ASSIGNED (/02 is assigned to NO device or has not been assigned at all)
INPUT FILE CANNOT BE ASSIGNED (the system has to assign a temporary work file to the file which must be listed but this turns out to be impossible. A message will follow explaining the error).
EOV ON INPUT FILE, MOUNT NEW TAPE THEN RESTART (the input file code is assigned to a magnetic or cassette tape and its end-of-volume is encountered before the whole file has been listed. To continue the operation, the operator must mount the next tape or turn over the cassette and restart the program. The EOV mark is not considered as a record, so it is not listed).

MAGNETIC TAPE CONTROL COMMANDS

- REW⊔<file code> is used to rewind the tape on the device to which <file code> has been assigned.
- ULD⊔<file code> is used to switch the device specified by <file code> to manual state by sending an 'unlock' command to the cassette tape unit or a 'switch off' command to the magnetic tape unit.
- FFS⊔<file code>[,<number>|ALL] is used to position the device after a tape mark. <number> if used, indicates the number of tape marks which must be skipped. If ALL is specified, the device will be positioned after two consecutive tape marks. Default value=1.
- FBS⊔<file code>[,<number>] is used to space the tape backwards across the previous tape mark or across a number of tape marks as specified by <number>.
- RFS⊔<file code>[,<number>] is used for spacing forward over a number of records, if <number> is specified. If it is not, a forward spacing is done until immediately after he next physical record.
- RBS⊔<file code>[,<number>] is used for backwards spacing, either until immediately past the previous record, or, if <number> has been specified, past the specified number of previous records.
- WES⊔<file code>[,<number>] is used to write one or a <number> of EOS records onto the device specified by <file code>.
- WEF⊔<file code>[,<number>] is used to write one or a <number> of tape marks onto the device specified by <file code> (EOF records for devices other than tape).
- WLB⊔<file code>,<serial number>,<sec. code>,<owner> is used to write a volume label on a magnetic or cassette tape, indicated by <file code>.

  <serial number> is the volume serial number, up to 6 numeric characters.
  <security code> consists of one hexadecimal character.
  <owner> is a character string of up to 39 characters, where blanks and commas are accepted.
  The label is written as follows, in ASCII:



  The label is followed by a tape mark.
  If <security code> is specified as NO, no volume label but only a tape mark is written.

- PLB⊔<file code> is used to have the volume label of a magnetic or cassette tape printed on the operator's typewriter and the tape positioned at the first record of the file following the label. The CCI does not check the label. If the label is absent and a tape mark is read as the first record on the tape, the message NO LABEL is printed.
- WEV⊔<file code> is used to write an End-Of-Volume mark on magnetic or cassette tape.
- REF⊔<file code> is used to position the file specified at the first record. It can be used for cassette tape, magnetic tape and disc.

The commands REW, FFS, FBS, ULD can be used only with cassette and magnetic tape units.

The following error messages are possible for these commands:
FILE CODE MISSING (no <file code> specified)
FILE CODE ERROR
INVALID FILE CODE
FILE CODE UNKNOWN
PARAM ERROR (error in the second parameter)
PARAM MISSING
TOO MANY PARAM
I/O ERROR

MES SEND MESSAGE MES

syntax: MES␣<message to the operator>

use: This command is used especially in batch processing mode or catalogued procedures to have the message specified typed out to the operator.

MOV MOVE A FILE MOV

syntax: MOV␣<name>,[/S|/L|/<file code>][,<userid>[,<disc number>]]

use: This command is used to move a file from a library - generally of another user (<userid>,<disc number>) to a temporary file /S or /L or to a file indicated by a file code.
<name> is the name of the library file which is to be moved.
/S: the file must be moved to temporary file /S.
/L: the file must be moved to temporary file /L.
<file code> : file code of the temporary file to which the file <name> must be moved.
<userid> : user identification of the user whose file must be moved (if the file belongs to another user than the current one).
The type of the file must be compatible with the type of the receiving temporary file:

- /S: source
- /L: load
- <file code> : undefined.

If /S or /L or <file code> has already been assigned to a temporary file, the latter file is lost.
After a file has been moved, it can be used directly as a temporary file, or it can be kept in a library (KPF command).

errors: FILE TYPE MISSING (the second parameter is absent)
FILE TYPE ERROR (the second parameter is neither a file code, nor /S, nor /L)
USERID ERROR
USERID UNKNOWN
TOO MANY PARAM
INPUT FILE ASSIGN ERROR
OUTPUT FILE ASSIGN ERROR (these messages are followed by another one specifying the cause of the error)
I/O ERROR (an I/O error has been encountered during the read or write operation and the file is not copied. A new MOV command has to be given).

NOD

DEFINE NODE

NOD

syntax: NOD␣<name>

use: By means of this command, the user can define a node in an overlay structure (see Programmer's Guide 2, Vol II: System Software).
<name> must be a string of 6 ASCII characters for acceptance by the OLE. The Control Command Interpreter, however, will copy any number of characters onto the object file.
The command is written as it is encountered, after which an: EOS is written onto the /O file.

errors: UNKNOWN FILE (/O file has not yet been opened)
NO OBJECT FILE (file on which command must be written is not/O)
CLOSED OBJECT FILE (/O file has already been closed)
I/O ERROR (impossible to write on /O file).

PCH PUNCH A FILE PCH

syntax: PCH␣[/<file code>|/S|<name>|<name>,/S] [,<new file code>]

use: This command is used to have a file punched on the punch unit. The file must be sequential with a maximum record length of 132 characters. The file may be of one of the following types, as specified in the command:
<file code>: file code of a temporary user file which must be punched.
/S: the source file must be punched.
<name>: name of a catalogued user data file which must be punched.
<name>,/S: name of a catalogued source program which must be punched. If any records of over 132 characters are encountered in the file, they are truncated. The file must be closed by an EOF record, otherwise the file will be punched up to the End-Of-Volume, where the last records of the last granule may be wrong.
<new file code>: output file code. Default value is /03.

errors: /S EMPTY
FILE NAME ERROR
FILE NAME MISSING
INVALID FILE CODE
FILE CODE NOT ASSIGNED
TOO MANY PARAM
INPUT I/O ERROR
OUTPUT I/O ERROR
FILE TYPE ERROR (the parameter following name is not /S)
INPUT FILE CANNOT BE ASSIGNED (it is impossible for the system to assign a work file code to the input file. A message will follow explaining the cause of the error).
EOV ON OUTPUT FILE, MOUNT NEW TAPE THEN RESTART (an end-of-volume has been encountered on magnetic or cassette tape output device. The operator must mount a new tape or turn over the cassette and restart the program to continue the output operation).
OUTPUT FILE CODE ERROR

PLD PUNCH LOAD PLD

syntax: PLD␣<name>[,/L|,<file code>|,/L,<file code>]

use: This command is used to punch a load file from the library or from the temporary file /L. The file will be punched on the punch unit in object code format.
<name> : name of a load file in a library. If /L is specified, the temporary file /L will be punched with ident record built from the name specified in the command, if this name is different from the IDENT name of the program.
<file code>: output file code. Default value is /03.

errors:
PARAM ERROR
FILE CODE NOT ASSIGNED
PARAM ABSENT
FILE NAME NOT CATALOGUED
NO LOAD MODULE
OUTPUT FILE CODE ERROR

POB PUNCH OBJECT POB

syntax: POB␣[<name>][<name>|<file code>|<name>,<file code>]

use: This command is used to punch an object module from the object file of the library or to punch the whole contents of the temporary object file /O.
<name> is the name of the library object module which must be punched. If no name is specified, the whole temporary object file /O will be punched.
When the temporary file /O must be punched and no EOF has yet been written after it, the EOF mark is first written, before the file is rewound and punched.
<file code>: output file code. Default value is /03.

errors:
OBJECT MODULE NAME ERROR
/O EMPTY
/O CLOSE ERROR (error detected during writing of EOF and rewinding of /O file)
/O INPUT ERROR (error detected during reading of /O file)
OUTPUT I/O ERROR (I/O error encountered during the output operation)
INPUT I/O ERROR (error detected during reading of the module which is to be output)
ILLEGAL EOS IN INPUT FILE (the first record of a module is an EOS)
OBJECT MODULE NOT CATALOGUED
IDENT MISSING (the first record of a module in the object library is not an IDENT)
OBJECT LIBRARY ASSIGN ERROR (it is impossible for the system to assign a work file to the user object library. A following message will explain the cause of the error).
EOV ON OUTPUT FILE, MOUNT NEW TAPE THEN RESTART (an end-of-volume has been encountered on a magnetic or cassette tape output device. The operator must mount a new tape or turn over the cassette and restart the program to continue the output operation).
OUTPUT FILE CODE ERROR

PRINT OBJECT DIRECTORY

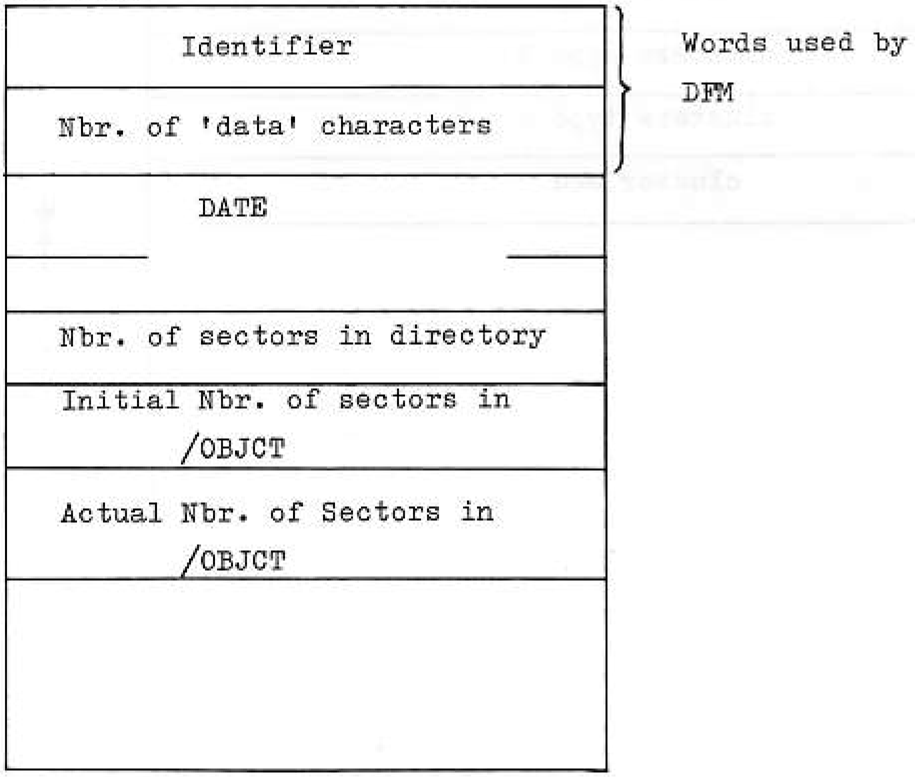
syntax: POD

use: This command is given to obtain a listing of the OBDIR user object directory file which has been created when a KPF /O command was given.
In the print-out, the first line is self explanatory. Under the second line the name of the module will be found, then under STRT, INIT and ACTL the sector number on which the module starts, the initial number of sectors filled by the module and the actual number of sectors filled by the module respectively.

errors: I/O ERROR (incorrect I/O request)
INVALID FILE (OBDIR is not assigned to /D8)
INVALID OBJECT (an EOF has been encountered)
INVALID OUTPUT (there has been an abnormal sequential I/O)
DIRECTORY ERROR (OBDIR format incorrect)
ASSIGN FILE CODE ERROR (/D8 could not be assigned)
DISK NO ASSIGN ERROR (/D8 deletion incorrect)

Structure of Directory

- Sector 0



Identifier

Nbr. of 'data' characters

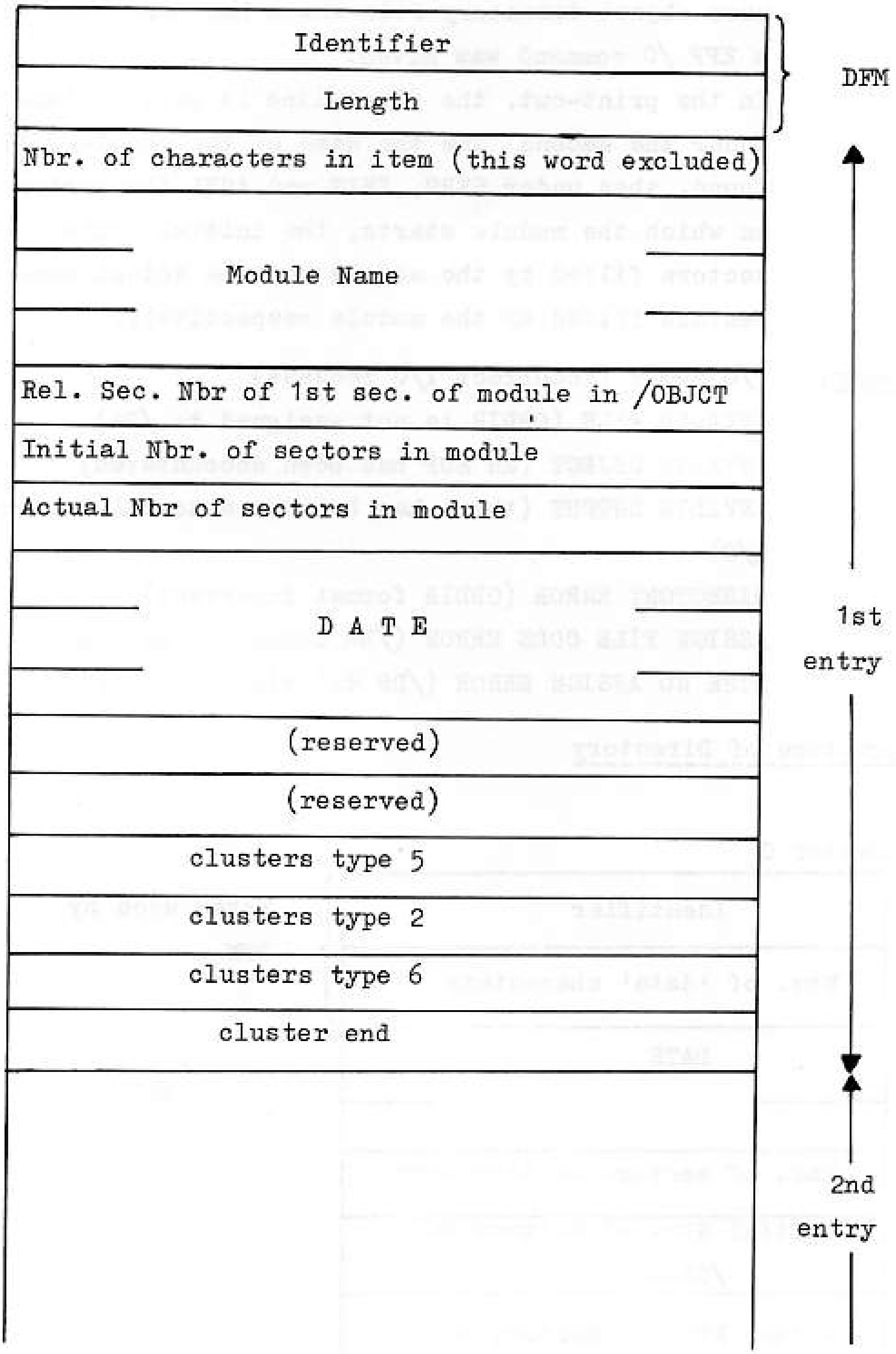
DATE

Nbr. of sectors in directory

Initial Nbr. of sectors in /OBJCT

Actual Nbr. of Sectors in /OBJCT

- Following sectors



Identifier

Length

DFM

Nbr. of characters in item (this word excluded)

Module Name

Rel. Sec. Nbr of 1st sec. of module in /OBJCT

Initial Nbr. of sectors in module

Actual Nbr of sectors in module

D A T E

(reserved)

(reserved)

clusters type 5

clusters type 2

clusters type 6

cluster end

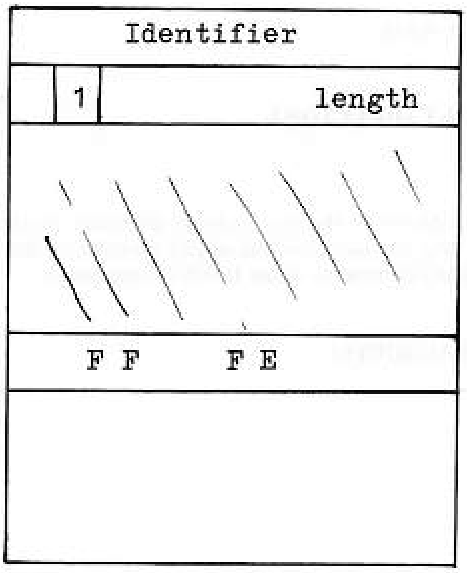
1st entry

2nd entry

Remarks:

- The last 3 words of sectors are not used.
- Bit 1 is set in last sector's 2nd word to indicate the end of directory.

Last sector:



- In this last sector, the last directory's entry is followed by the constant /FFFE.

PRC PRINT CATALOGUE PRC

syntax: PRC␣/<disc number>

use: This command can only be used in a system session, i.e. when the user gives, at the start of the session, the user identification SYSTEM. It causes a print-out on the print unit of the catalogue contained on the disc specified. /<disc number> gives the address of the related disc.

errors: SYSTEM SESSION COMMAND (the current session is not a SYSTEM session)
PARAM MISSING
PARAM ERROR
FILE CODE NOT ASSIGNED.

PRD PRINT DIRECTORY PRD

syntax: PRD␣[/OB]

use: This command causes a print-out of the user's library directory on the print unit. If /OB is specified, only the names of the object modules in the object file are printed, including any comments in the IDENT statements.

errors: PARAM ERROR
NO OBJECT FILE CATALOGUED

PRT PRINT FILE PRT

syntax: PRT␣[< file code > | / S | / S, < name > |< name >][, < line nb 1 >[, < line nb 2 >]]

use: This command causes a listing of the specified disc file on the print unit. The file must be sequential and consist of ASCII records. If a record is longer than a print line it will be printed on several lines.
The file can be either:
- a catalogued source file: /S,<name>
- a catalogued user data file: <name>
- a temporary data file: <file code>
- the temporary source file: /S.

<line nb1>,<line nb2> if specified, provides for a listing of the file from the first line number to the second line number specified.

The maximum record size allowed is 132 characters. Records which are longer will be truncated. Non-printable record characters will be replaced by spaces and trailing blanks will be removed. The listing will stop when either <line nb2>, an EOF or End-Of-Volume (the last granule of the file) is reached. End-Of-Volume occurs when no EOF has been written for this file, which normally is the case for temporary files created by a program in the debugging phase. After the printing, the file is positioned at the last record printed.

errors: FILE NAME ERROR (the first parameter is neither /S nor a file code nor a character string)
FILE NAME MISSING
LINE NUMBER ERROR
TOO MANY PARAM
INPUT FILE I/O ERROR
OUTPUT FILE I/O ERROR
FILE CODE ERROR
OUTPUT NOT ASSIGNED (/02 is assigned to NO device or has not been assigned at all)
INPUT FILE CAN NOT BE ASSIGNED (the system has to assign a temporary work file to the file which must be printed but this turns out to be impossible. A message will follow explaining the error).
EOV ON INPUT FILE, MOUNT NEW TAPE THEN RESTART (the input file code is assigned to a magnetic or cassette tape and its end-of-volume is encountered before the whole file has been printed. To continue the operation, the operator must mount the next tape or turn over the cassette and restart the program. The EOV mark is not considered as a record, so it is not printed).

PSE PAUSE PSE

syntax: PSE␣[<message to the operator>]

use: This command is used to put the machine in pause state and have the specified message typed out for the operator.
The operator must restart the program to have the next command read.

RDA READ DATA RDA

syntax: RDA␣/<disc file code>[,/<input code>]

use: This command is used to read and transfer data to a temporary user file.
/<disc file code>: the temporary user file to which the data are transferred. This file code does not have to be assigned by the user; an implicit assignment will automatically be made by the system.
/<input file code>: file code from which the data are read. Default value is /E1 (source input).
The data will be read until an EOF is encountered.
The file codes specified must be in the range from /01 to /EF.

errors: DISK FILE CODE MISSING
DISK FILE CODE ERROR
INVALID DISK FILE CODE
INVALID PARAMETER (the file code is not a numeric value in the range from /01 to /EF)
FILE CODE NOT ASSIGNED (the file code is assigned to NO device or has not been assigned at all)
TOO MANY PARAM
DISK ASSIGN ERROR (the system is unable to assign a disc temporary work file. A message will follow giving the cause of the error)
INPUT I/O ERROR
OUTPUT I/O ERROR (an error has been detected during the last read operation from the sequential file or during the last write operation to the disc temporary file).

RDO READ OBJECT RDO

syntax: RDO⌴[/<file code>]

use: By means of this command it is possible to copy an object file from the object input unit or from another sequential input unit onto the disc as a /O file or as a complement to the /O file.

<file code> : file code of the input unit from which the object file is to be read. If not specified, the object file is read from the standard object input unit.

No EOF record is written onto the disc.

If there is already a /O file on the disc which has not been closed with an EOF record, the object file is copied after this /O file.

If there is no /O file on the disc or if it has already been closed with an EOF record, a new assignment is done for /O starting at a new granule and the old /O file is lost.

It is possible to give several RDO commands to fill /O with several object files. Mixing RDO commands with language processor calls in order to fill the /O file is allowed also.

Note:

An EOF mark is written after the /O file when one of the following commands is given:

KPF /O[,<name>]

POB

LKE

errors: INVALID PARAMETER (the file code is not a numeric value in the range from /01 to /EF)

FILE CODE NOT ASSIGNED (the file code is assigned to NO device or has not yet been assigned at all)

TOO MANY PARAM

DISK ASSIGN ERROR (the system is unable to assign the disc temporary file. A message will follow giving the cause of the error)

INPUT I/O ERROR

OUTPUT I/O ERROR (an error has been detected during the last read operation from the sequential file or during the last write operation to the disc temporary file).

RDS READ SOURCE RDS

syntax: RDS⊔[/<file code>]

use: By means of this command it is possible to copy a sequential source program file or a sequential data file from the source input unit or from another sequential input unit onto the disc as a /S file.
If a /S file had already been built previously and not been made permanent in the library (KPF command), the old file is lost because a new assignment is made for the /S file starting at a new granule.
Records are 80 characters long and terminated by an EOS or EOF record. The last 8 characters of the input records are replaced by spaces.
When an EOS or EOF is encountered, the system writes the EOS on the /S file, followed by the EOF.

errors: INVALID PARAMETER (the file code is not a numeric value in the range from /01 to /EF)
FILE CODE NOT ASSIGNED (the file code is assigned to NO device or has not yet been assigned at all)
TOO MANY PARAM
DISK ASSIGN ERROR (the system is unable to assign the disc temporary file. A message will follow giving the cause of the error)
INPUT I/O ERROR
OUTPUT I/O ERROR (an error has been found during the last read operation from the sequential file or during the last write operation to the disc temporary file).
EOV ON INPUT FILE, MOUNT NEW TAPE THEN RESTART (the input file is a magnetic or cassette tape and the end-of-volume has been encountered before the end-of-file mark. The operator has to mount the next reel of tape or turn over the cassette and restart the program).

RSU REPLACE SUPERVISOR RSU

syntax: RSU␣/<disc number>

use: This command can only be used in a system session, i.e. when the user gives, at the start of the session, the user identification SYSTEM.
Then this command may be used to change the monitor on the disc, by copying the contents of the /L file onto the disc, starting from absolute sector number 18. The system ensures that <disc number> is a system disc, i.e. contains a load module file starting at address /10 built of consecutive granules and that the file size is ≥ /L.

errors: COMMAND NOT ALLOWED
PARAM ERROR (the parameter must be numeric)
DISK ADDRESS MISSING
INVALID DISK ADDRESS
TOO MANY PARAM
DISK UNKNOWN
DISK NOT OPERATIONAL
/L EMPTY
DISK I/O ERROR (an I/O error has been detected during the reading of the /L file or during the writing onto the specified disc).
/FX NOT A SYSTEM DISC
/L TOO BIG.

RUN RUN A PROGRAM RUN

syntax: RUN␣[<name>][,0]

use: This command must be used to start a program after it has been loaded from a library or from the /L file.
<name>: name of the program, if it is a program which is loaded from library. If no name is specified, the program is loaded from the /L file.
0, is specified to indicate that the program runs in system mode (P856/P857 only).

errors: PROGRAM NAME ERROR
/L EMPTY
PROGRAM NOT CATALOGUED
TOO MANY PARAM

SCR SCRATCH SCR

syntax: SCR␣[[/S|/O|/L|/<file code>]]

use: This command can be used to release previously made user assignments.
It is processed as follows:

- file codes /0A through /DF are deleted
- file codes /01 through /CF and /E1 through /EF are deleted when assigned to a disc file
- file code /E0 is deleted when assigned to a temporary disc file.

If no parameter is specified, all user assignments are released and the pointer for the granule allocation table is reset to the first available granule, i.e. the system is reset to the state it was in at the beginning of the session.
If /S, /O, /L or <file code> is specified, this indicates the file whose assignment must be released. In such a case, the corresponding entry in the file table of the monitor is made available for other assignments. This may be very useful in cases where table overflow problems must be avoided, or it can be done after an error message for table overflow has been received (the number of entries in the file table is defined at system generation time).

Note:

- The file codes from /01 to /09 and from /E0 to /FF can not be scratched, because they are reserved or system file codes.
- SCR may be defined in catalogued procedures only as SCR␣/S or SCR␣/O

errors: INVALID PARAM
INVALID FILE CODE
I/O ERROR (an I/O error has been detected during the loading of the allocation table).

SDM SAVE DISC ONTO MAGNETIC TAPE SDM

syntax: SDM⊔<disc number>,<file code>[,CK]

use: This command is used to save the contents of a disc onto a magnetic tape.
<disc number> is the file code (from /F0 to /FF) of the disc which must be saved.
<file code> is the file code of the magnetic tape on which the disc is saved.
When the SDM command is given, the Control Command Interpreter will write a stand-alone, self-loadable program at the beginning of the tape to IPL the saved contents from MT to DK, then output the contents of the disc onto it, sector by sector. The tape is closed by a tape mark. Thus, the contents of a disc can be saved or duplicated, which is very convenient in configurations with only one disc unit.
CK: If CK is specified, the magnetic tape is rewound and compared to the disc.

errors: PARAM MISSING
DISK FILE CODE ERROR (first parameter is not a file code)
INVALID FILE CODE (Ffirst parameter is not a value from F0 to FF)
FILE CODE NOT ASSIGNED (disc file code is unknown)
2ND FILE CODE ERROR (the second file code is not assigned to a manetic tape or it is not a correct value for a file code)
TOO MANY PARAM
DISK I/O ERROR (an irrecoverable I/O error occurred while reading the disc)
I/O ERROR (an irrecoverable I/O error occurred on the magnetic tape).

SEG DEFINE SEGMENTS SEG

syntax: SEG␣<name list>

use: This command enables the user to build his program in a kind of overlay structure. It defines the library program names of the program parts which will be used as segments by a root program which is started in a following RUN command.

<name list> consists of one or several library program names, separated by commas. The list may not contain more than 15 names.

The segments specified will be called at run time by the root program or by other segments through the Load monitor request (LKM DATA 9). The numbers of the segments are implicitly defined by the order in which they appear in the <name list>, i.e. the first program name specified will be segment number 1.

The following command must necessarily be a RUN command.

Note:

Special care must be taken with respect to the dynamic allocation area when using segmented programs. After the Root segment has been loaded, the start address of the dynamic area is the first address following the root. Now, if another segment were loaded without any provisions the dynamic area would be overwritten, which must be avoided if any Get and Release Buffer requests are used. The user must take care that the start address of the dynamic allocation area is behind the longest segment which the root program is going to load. This may for example be done by giving a request for area reservation (RES) in the Root program to reserve an area behind the root segment into which the other segments will be loaded. The dynamic area will then start behind this reserved area.

errors: SEGMENT NBR. 01 MISSING (no parameter present)

SEGMENT NBR. XX NOT CATALOGUED (the segment number XX is not catalogued or has been declared more than once in the command; all the <names> of the list must be different)

SEGMENT NBR. XX ERROR

TOO MANY PARAM (the number of segments in the command is greater than the maximum number of segments declared at system generation time).

SKF SKIP FORM SKF

syntax: SKF␣[<number>]

use: By means of this command, a number of pages may be skipped on file code /02, i.e. the line printer.
<number> is the number of pages to be skipped. Default value is 1.

SVD SAVE DISC ONTO ANOTHER DISC SVD

syntax: SVD␣/<disc number1>,/<disc number2>

use: This command can only be used in a system session, i.e. when the user gives, at the start of the session, the user identification SYSTEM.
It is used to copy the contents of one disc onto another one.
All allocated granules of the disc specified by <disc number1> are copied onto the disc specified by <disc number2>.
If the capacity of the discs is different, only as many sectors will be duplicated as the smaller disc contains.
The discs are assumed not to contain any defective tracks, for the copying is done sector per sector, sequentially.

Note:
- SVD /FX,/F0 is not allowed.
- The volume label of <disc number2> is not destroyed.

errors: COMMAND NOT ALLOWED (the current session is not a system session)
FIRST FILE CODE MISSING
SECOND FILE CODE MISSING
FIRST FILE CODE UNKNOWN
SECOND FILE CODE UNKNOWN
FIRST FILE CODE ERROR
SECOND FILE CODE ERROR
TOO MANY PARAM
INPUT I/O ERROR
OUTPUT I/O ERROR
INVALID DISK TYPE (the disc to which one of the two file codes is assigned, is not supported by the system).

SVU SAVE USER FILES SVU

syntax: SVU␣<userid>,/<disc number>

use: This command is used to copy all the files of the user specified and contained on the disc specified, into the library of the current session user. This may be the same user as the one specified by <userid>.
<userid>: user identification, a string of up to 8 characters.
/<disc number>: file code of the disc containing the files which must be saved.
The files will be copied one at a time, up to 'end of medium'.
The new file will be kept in the directory of the current user under the same name and type. If the name has already been kept previously in the directory, the old file is scratched and replaced by the new one. All the sectors of the file are copied, one at a time, except for the deleted sectors of the object library file. The files are copied in the same order as they appear in the directory of the user specified by <userid>.

errors: INVALID USERID (the first parameter is not a character string)
DISK FILE CODE ERROR (the second parameter must be a binary value)
USERID MISSING (the command contains no parameter)
DISK FILE CODE MISSING (the second parameter specifying the disc file code is not specified in the command)
INVALID DISK FILE CODE (the second parameter is numeric but not in the range from /F0 to /FF)
DISK FILE CODE UNKNOWN (the file code specified can not be found in the system file code table. It has not been declared at SYSGEN)
DISK NOT OPERATIONAL (the disc to which the file code has been assigned, is not operational. If it has just become ready, a retry is possible, otherwise the error must be corrected)
TOO MANY PARAM
USERID NOT CATALOGUED (the userid given in the command does not exist on the disc specified)
INPUT DISK I/O ERROR (an I/O error has been detected during a read operation from the disc specified in the command)
OUTPUT DISK I/O ERROR (an I/O error has been encountered during a read/write operation from/to the disc used in the current session)
INPUT FILE ASSIGN ERROR
OUTPUT FILE ASSIGN ERROR (in order to save all the files of the user specified, the system assigns a temporary work file code to a file of this user and another one to the disc file of the current session. This may be impossible, in which case a message will follow explaining the cause of the error)
DIRECTORY OVERFLOW ON XXXXXXFT (there is an overflow of the directory of the user of the current session, while the file with the name XXXXXX and the type FT is being catalogued).
The file type is SC for source files, OB for object files, LM for load modules and UF for user data files. This file can not be catalogued. The user may give a PRD command to find out which files have been copied so far, since all files are copied in the order in which they appear in the directory of the user specified by <userid>.

PROCESSOR CALLS

ASM ASSEMBLER ASM

syntax: ASM␣[/S|<name>][,NL]

use: This command must be used to assemble a source program from a library or from the temporary file /S.

/S: the source program must be assembled from the temporary file /S.

<name> : the source program to be assembled can be found in the library. <name> gives the name of this program.

NL: if specified, no listing will be provided of the assembled program. Otherwise, the listing is output on the print unit. Any error messages will be output on the operator's typewriter as well.

The object code is produced on the temporary file /O. If /O does not exist, an assignment will be made for it. If /O does already exist, the object module will be written at the end of this file, unless /O has already been closed with en EOF record. In that case a new assignment is made and the old /O file is lost.

Note:

If a fatal error occurs during assembly, the whole /O file will be deleted and a request for Link Edit will be refused.

errors: FILE NAME ERROR

FILE NAME MISSING

INVALID PARAM

/S EMPTY (no temporary source file exists)

/S ASSIGN ERROR (it is impossible to assign the file code /D4 to the catalogued source file. A message will follow to explain the error)

/O ASSIGN ERROR (an attempt to assign the temporary object file /O is refused. A message will follow to explain the error)

NL OPTION ERROR (NL has been declared more than once in the command)

PROCESSOR NOT CATALOGUED (a segment of the assembler has not been catalogued).

DEB DEBUGGING PACKAGE DEB

syntax: DEB␣[<name>]

use: This command must be used to call the Debugging Package and run a load module under its control.
<name> is the name of the load module if it is to be found in a library. If no name is specified, the temporary file /L is considered to be the load module. The Debugging Package loads the load module into memory by means of the Load monitor request. This implies that it is not possible to debug a segmented program. During the debugging phase a temporary file is created on the user disc for a copy of the Debug Processor on the system disc. This file, together with the user program which must be debugged are considered as a segmented program where the temporary file is the root and the user program is segment number 1.

errors: PROGRAM NAME ERROR (the parameter is not a character string)
/L EMPTY (there is no program in the temporary load file)
PROGRAM NOT CATALOGUED (the program <name> cannot be found in the directory)
TOO MANY PARAM
PROCESSOR NOT CATALOGUED (the debug processor has not been catalogued on the system disc)
SYSTEM DISK I/O ERROR
USER DISK I/O ERROR (an I/O error has been detected during the copying of the debug processor from the system library to a temporary file on the user disc)
ASSIGN ERROR (the system is unable to perform an assignment. A message will follow explaining the reason for this error).

FOR FULL FORTRAN COMPILER FOR

syntax: FOR⊔[/S|<name>][,NL]

use: This command must be used to compile a FORTRAN source program from a library or from the temporary file /S.
/S: the program must be compiled from the temporary file /S.
<name> : the program to be compiled can be found in the library. <name> gives the name of this program.
NL: if specified, no listing will be provided of the compiled program. Otherwise, a listing is output on the print unit. Any error messages will be output on the operator's typewriter as well.
The object code is produced on the temporary file /O. If /O does not exist, an assignment will be made for it. If /O already exists, the object module will be written at the end of this file, unless /O has already been closed with an EOF record. In that case a new assignment is made and the old /O file is lost.

Note:
If a fatal error occurs during compilation, the whole /O file will be deleted and a request for link edit will be refused.

errors: FILE NAME ERROR
FILE NAME MISSING
INVALID PARAM
/S EMPTY (no temporary source file exists)
/S ASSIGN ERROR (it is impossible to assign the file code /D4 to the catalogued source file. A message will follow to explain the error)
NL OPTION ERROR (NL has been declared more than once in the command)
PROCESSOR NOT CATALOGUED (a segment of the compiler has not been catalogued)
/O ASSIGN ERROR (an attempt to assign the temporary object file /O is refused. A message will follow to explain the error).

HSF HIGH-SPEED FORTRAN HSF

syntax: HSF␣⟦/S|<name>⟧[,NL]

use: This command is used to load the high-speed FORTRAN compiler into memory to start compilation of a source module from the temporary file /S or from the user library.

/S: the program must be compiled from the temporary file /S.

<name>: the program to be compiled can be found in the library.

<name> is the name of this program in its directory.

NL: if specified, no listing will be provided of the compiled program. Otherwise,a listing is output on the print unit. Any error messages will be output on the operator's typewriter as well.

The object code is produced on the temporary file /O. If /O does not exist, an assignment will be made for it. If /O does already exist, the object module will be written at the end of this file, unless /O has already been closed with an EOF record. In that case a new assignment is made and the old /O file is lost.

errors: FILE NAME ERROR
FILE NAME MISSING
INVALID PARAM
/S EMPTY
/S ASSIGN ERROR
/O ASSIGN ERROR
NL OPTION ERROR
PROCESSOR NOT CATALOGUED

LKE — LINKAGE EDITOR — LKE

syntax: LKE␣[N|S|U][,M][,[DE|DS]][,/<address>][,<start address>]

use: This command must be used to call the linkage editor. The parameters may be given in any order.

N: no library scanning is desired.

S: only the standard library has to be scanned.

U: only the user library has to be scanned.

Default value: both libraries will be scanned, the user library first.

M: a print-out of the map is wanted. Default: no map will be printed.

DE: all entry points must be preserved for the Debugging Package.

DS: only the pointers (implicit entry points) to the symbol tables must be preserved for the Debugging Package. The symbols have been preserved at assembly time.

Default value: no debugging.

/<address> : hexadecimal displacement value of the blank common from the beginning of the load module. The address must be specified in characters. If this address is an even number, the blank common address is assumed to be relocatable; if it is odd, the address is assumed to be absolute. This option may be used for communication between several load modules making use of the same blank common. In such a case the load modules must be called by the Load monitor request (LKM DATA 9) and have been defined previously in a SEG command.

The address of the blank common must be defined in such a way that it will not be destroyed when a load module is loaded. The address must be defined at link-edit time for each one of the load modules of the overlay structure. Default value: the blank common, if any, will be located at the end of the load module.

<start address>: name of the start address defined as an entry in one of the modules in the /O file. The name must be different from the option parameters used in the LKE command: N, S, U, M, DE and DS.

Default value: the last start address encountered in the /O file.

Before starting LKE, an EOF record is written on the /O file if it has not already been closed with an EOF previously. Moreover, an assignment is made for the /L file onto which the Linkage Editor will write the load module.

Note:

If a fatal error occurs during the link-edit operation, the /L file is deleted and a RUN command will be refused.

errors: INVALID PARAMETER
COMMON VALUE REDUNDANT
LIBRARY OPTION REDUNDANT
DEBUG OPTION REDUNDANT
MAP OPTION REDUNDANT
START ADDR. REDUNDANT
USER LIB ASSIGN ERROR:
SYSTEM LIB ASSIGN ERROR
/L ASSIGN ERROR:
/O EMPTY
/O CLOSE ERROR
PROCESSOR NOT CATALOGUED

LED LINE EDITOR LED

syntax: LED␣< name > [[, < file code 1 > [, < file code 2 >]]] [, / S [, < file code 2 >]] [, xx]

use: This command is used to call the Line Editor and update a source file located in a library.
<name is the name of the source file.
<file code1> is the output file code. If specified, the file is written as a UF file.
<file code2> specifies the input command file code from which the control commands are entered. Default value: / E0. If this file code is assigned to a typewriter, the Line Editor prints **L**: before reading a record.
xx are two characters, specified if the user wants command control characters other than !! (see below).
The updated output file is written as a temporary /S file. If /S already exists, a new assignment is made and the old /S file is lost.

errors: FILE NAME ERROR
FILE NAME MISSING
INPUT FILE CANNOT BE ASSIGNED (followed by message explaining the error)
/S CANNOT BE ASSIGNED (followed by message explaining the error)
INVALID FILE CODE
FILE CODE NOT ASSIGN
TOO MANY PARAM
DSK INPUT ERR, UPD ABORTED
DSK OUTPUT ERR, UPD ABORTED
UNKNOWN COMMAND, TRY AGAIN
I/O ERR ON LAST RECORD, TRY AGAIN
SEQUENCE ERR, TRY AGAIN
SYNTAX ERR, TRY AGAIN
EOF, UPD TERMINATED (the EOF mark has been encountered on the input source file before reaching the specified line, thus terminating the update process)
AUX. INPUT CANNOT BE ASSIGNED, TRY AGAIN (auxiliary file used in JN command cannot be assigned)
CMND NOT ALLOWED IN EXE MODE, TRY AGAIN (definition mode command)
TABLE O'FLOW, TRY AGAIN (character string table is overflowing)
EOF IN AUXI INPUT (JN command is terminated but operation continues).

When the message TRY AGAIN is printed, the user may correct the erroneous command or data record from the device with file code 01. If he types only **CR**, input is resumed from the device with the normal Input File Code.

With the Line Editor, there are two phases of operation: definition and execution. The user first defines the modifications to be made to the file. The commands given are not executed immediately but recorded until the updated file is created. When the Line Editor is called, it is flagged as being in definition mode, to be closed implicitly when an execution mode command is encountered.
Execution mode commands are executed immediately, i.e. creation of the updated file is started. After executing the execution mode commands, the Line Editor will look for any pre-recorded definition mode commands. Once execution mode has been started, it is impossible to return to definition mode.

Definition mode Commands:

!!CH␣ $$** <character string1> **$$** <character string2> **$$

is used to have <character string1> replaced by <character string2> wherever it appears in the program, no matter how many times it is encountered. To replace only one line, see !!RE below. This command is pre-stored in memory but not yet executed.

!!LS␣$$ <character string> $$

causes a listing of the lines containing the <character string> specified to be printed immediately.

Note: When giving a !!CH command, take care that with a !!RE command <character string1> could be affected by the !!CH command.

Execution mode Commands:

!!JN␣⌈<line number>⌉,<name>,<line1>,<line2>

is used to insert the lines from <line1> to <line2> inclusive of the module named <name> after the <line number> of the current input. If <line number> is not specified, the lines are inserted behind the comment line of the main input.

!!RE␣<line number>,$$ <character string1> $$<character string2>$$

is used to have <character string1> replaced by <character string 2> in the line of which the <line number> is specified.

Note: After !!RE immediate insertion of new lines is possible. An !!IL on the same line is not possible.

Further LED commands are:

!!DL␣<line1>⌈,<line2>⌉

which is used to delete the specified line or the lines form <line1> to<line2>inclusive.

!!IL␣⌈<line number>⌉

which is used to insert one or more lines after the specified line number or, if no parameter is specified, after the current statement. It is possible to insert a line before the first line of a coding sequence, e.g. if an IDENT statement has been forgotten.

!!AB is used to abort a line edit operation

!!EN which is used to terminate the line edit session.

Note:

- The commands must be given in ascending order of line numbers.
- Only sequential input files consisting of records of up to 80 characters can be handled.
- The definition commands are pre-stored in memory, not on disc.

OLE OVERLAY LINKAGE EDITOR (P857 only) OLE

syntax: OLE␣⌈N|S|U⌉⌈,M⌉⌈,⌊DE|DS⌋⌉⌈,/<address>⌉⌈,<start address>⌉

use: See Linkage Editor (LKE) command

errors: same as for LKE, plus:
USER OBDIR ASSIGN ERR:
SYST DBDIR ASSIGN ERR:
(error detected when system attempts to assign a file code to the system or user object directory. A following message will indicate the cause of the error).

UPR USER PROCESSOR UPR

syntax: UPR␣<processor name>,⌊/S|<file name>⌋[,NL⌋

use: This command allows for calling a user-made processor from the user library. <processor name> is the name it has as a load module file under the user directory. The other parameters have the same meaning as under ASM.
This feature is useful in conjunction with the GEN directive of the DOS Assembler, allowing the use of assemblers with predefined mnemonics (FORM) or symbols (equivalence tables) without affecting the standard Assembler in the system area.

example: – creating a special Assembler:

```
USERID: USER
MOV     ASM,/L,SAG
ASG     /20,DK
PLD     ASM,/L,/20
REF     /20
RDO     /20 (store assembler on /O file)
ASM     GEN (assemble GEN section with FORM directives and symbol
        definition)
LKE     N,M
KPF     /L,ASMSPE
```

– use of special assembler:

```
USERID: USER
UPR     ASMSPE, SOURCE (SOURCE is module to be assembled)
```

11 Operator Control Messages

There are a number of control messages which the operator can enter via the typewriter. To do so, he must first press the interrupt button on the control panel of the CPU (INT). The system then types out M: and the operator can type a control message.
Such a message consists of two characters identifying the message, followed by a space, possibly followed by one or more parameters, followed by CR - LF. The parameters are separated by spaces or commas.

Note:

- The operator communication package is an optional module. If this module is not included in the monitor, the operator must take care not to press the interrupt button (INT) on the control panel, because then the running program will be aborted.
- Every numeric value in an operator control message must be a hexadecimal value, specified with or without a slash (/). Between parameters, blanks or commas are allowed.
- If a message contains an error, the system types out a message ER. The operator may then press the INT button and type in a correct message.

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

| means: or

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

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

␣ means: space.

<> means: these brackets contain a syntactic item.

In the following paragraphs the operator control messages are specified in alphabetical order according to their mnemonics.

Table of Operator Control Messages

AB ABORT A PROGRAM AB

syntax: AB

use: This message definitively stops a running program. A message is sent to the operator by the system, specifying the program location where the abort occurred and the reason for the abort, in this case: OPERATOR ABORT. If a PU error message is given with retry possibility (RY) and the retry works, the abort will not be effectuated.

AS ASSIGN FILE CODE AS

syntax: AS␣<file code>␣<device address>

use: By means of this message a new file code can be created or the assignment of a previously created file code can be modified.
<file code> is the file code to be assigned, consisting of two hexadecimal characters from 01 to CF or from E0 to FF, except for F0 which is always the system disc.
<device address> is the physical address of the device to which the assignment is made, consisting of device name and number.
The following device names can be used:

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

example: AS 08 LP0D
File code 08 is assigned to the line printer with physical address 0D.

DM DUMP MEMORY DM

syntax: DM␣<address1>␣<address2>

use: This message causes a memory dump on the print file, which is usually the line printer. The dump is made in full lines.
<address1> is the beginning address of the memory dump (up to 4 hexadecimal characters).
<address2> is the ending address of the memory dump (up to 4 hexadecimal characters).

example: DM 0002 0004
The contents of memory will be dumped from hexadecimal address 0000 to address 000E.

HD HALT DUMP HD

syntax: HD

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

MC MANUAL DEVICE CONTROL MC

syntax: MC␣<device address>␣<order>[␣<repeat factor>]

use: This message can be given if the operator wants to do a manual operation on a magnetic tape device.
<device address> is the physical address of the tape unit, consisting of two hexadecimal characters.
<order> must be one of the following hexadecimal numbers, each one of which indicates a specific operation:
16: skip forward to EOF mark
22: write EOF mark
24: write EOV mark
26: write EOS mark
31: rewind to load point
33: backspace one block
34: skip forward one block (not allowed on cassette)
36: skip back to EOF mark

38: Unlock
<repeat factor> allows the operator to have the required function performed as many times as specified here, with only one MC message.

example: MC 15 34
The device with physical address 15 is to skip forward one block.

PS PAUSE PS

syntax: PS

use: This message causes the running program to be stopped temporarily. In order to restart the program the operator must give the message RS (see below).

RD RELEASE DEVICE RD

syntax: RD⌴<device address>

use: When the monitor has typed out a PU error message (peripheral unit failure) after an I/O operation and thus requests operator intervention, the operator may type in this message if he wants to release the operation on the device which resulted in the error message (see System Messages).
<device address> gives the address of the device which the operator wants to release and consists of two hexadecimal characters, as specified at system generation time.
After release of the device, control is returned to the user with the error status in word 4 of the Event Control Block.

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

RS RESTART A PROGRAM RS

syntax: RS[⌴<value>]

use: This message causes a program which has been stopped temporarily by a Pause monitor request or PS operator message, to be restarted.
<value> is a hexadecimal value of up to 4 characters which, if specified, will be loaded into the A7 register by the monitor.
This new value for the A7 register may be specified only if the program has been stopped by the **monitor request** Pause.

RY RETRY AN I/O OPERATION RY

syntax: RY⊔<device address>

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

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

If the operation succeeds now, control is returned to the user with the status in word 4 of the Event Control Block.

If it still does not succeed, a new error message may be output by the monitor if another retry is possible, or the operator may release the device (RD operator message).

example: RY 02

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

WM WRITE INTO MEMORY WM

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

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

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

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

<value1> is put in location <address>

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

example: WM 4FE 44F 3FE4

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

The value 3FE4 is placed in memory location 500.

13 Monitor Requests

The user program can request the monitor to perform certain functions. A request takes the form of an LKM (Link to Monitor) instruction followed by a DATA directive.
The directive has a number as operand, which specifies the function to be executed. If this number is negative, the user is scheduling a label on completion of the request.
Preceding a request, certain parameters may need to be loaded into the A7 and A8 registers.
After the monitor has processed the request it loads a return code in the A7 register. If the requested service module is not available, A7 will always contain the value -1.

Note: It is possible to use scheduled labels in conjunction with a monitor request. See chapter 4.

I/O REQUEST

Calling Sequence

```
LDK      A7,CODE
LDKL     A8,ECBADR
LKM
DATA 1
```

Use

The user can ask the system to start a particular I/O operation on a peripheral device.
Processed at level 48 for physical I/O requests, at level 49 for logical I/O requests (Data Management).

Register A7 is loaded with a CODE specifying the details of the I/O function, as follows:

bit		7	8	9	10 15	
	.		W	R	ORDER	A7

W and R specify the mode of operation:

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

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

R = 1: the program itself will process any abnormal condition concerning the requested operation (possible only with Basic Read/Write). The system will return the hardware status in ECB word 4. No retry is possible.

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

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

01: Basic Read

05: Basic Write.

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

02: Standard Read

This order can be used to input standard object code records in 4 x 4 (4 rows of 4-bits: 4 track tape) or 8 + 8 (2 rows of 8 bits: 8-track tape) format, as well as ASCII character strings.

06: Standard Write

Standard (ASCII) I/O requests provide, by means of standard conversions, for special features such as error control characters, conversion from external code to internal ASCII and vice versa.

07: Object Write (4+4+4+4 tape format)

08: Object Write (8+8 tape format)

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

0A: Random Read

0B: Random Write

14: Skip forward to EOS mark

16: Skip forward to EOF mark

22: Write EOF mark

24: Write EOV mark

26: Write EOS mark

30: Get information about a file code

31: Rewind file.

33: Backspace one block

34: Space one block forward (not allowed for cassette)

36: Skip backward to EOF mark

38: Unlock.

(Physical disc access on sector level is possible with orders /11 (Read) and /15 (Write). The Disc File Management module is not used in these cases. The file code in ECB0 must be one of the disc file codes F0 to FF. ECB5 must contain the absolute sector number of the sector which is to be accessed. Therefore, when a disc is shared by Disc File Management and user physical disc access, extreme caution must be exercised).

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

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

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

X: these words must be filled by the user
Y: these words are filled by the monitor

where:

WORD 0: event character:
bit 0 = 1: end of operation has occurred for the ECB.
The other bits remain unused.

WORD 1: address of the user buffer

WORD 2: requested length to be read or written, in words (basic read on card reader) or characters (other devices). The first character is always the character given by the buffer address. For standard write on typewriter or line printer, two characters must be added, at the beginning of the buffer.

WORD 3: effective length which has been transmitted, in words (basic read on card reader) or characters (others). Stored here by the monitor upon completion of the I/O operation.

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

- For Basic orders, this word will be filled with the hardware status by the control unit. However, if the monitor detects an error in the calling sequence, bit 0 will be set to 1 and the other bits will contain the software status of page 128. Hardware status: Appendix D).
- For the other orders, the software status will be returned:

bit 0 = 0: the operation has been successfully completed:
bit 7 = 1: no data (tape cassette)
bit 8 = 1: End-Of-Volume } cassette or
bit 9 = 1: End-Of-Tape } magnetic tape
bit 10 = 1: beginning of tape encountered.
bit 11 = 1: end of input medium (disc only).
bit 12 = 1: requested length is incorrect.
bit 13 = 1: illegal character code.
bit 14 = 1: an EOS mark has been read.
bit 15 = 1: an EOF mark has been read.

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

When the monitor has detected an error in the calling sequence, bits 0 and 1 are both set to 1 and bits 2 to 15 have the following significance:

bit 10 = 1: file is write-protected.

bit 11 = 1: function is unknown or not compatible with the device or bit 9 in A7 set for a DFM request

bit 12 = 1: illegal buffer size or address

bit 13 = 1: illegal ECB address

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

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

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

- the relative sector number to be exchanged in the case of a random disc file (see Data Management).
- the tabulation table address in the case of a standard read operation on ASR or punched tape equipment. This tabulation table has the following format.

Number of Tackets	First Tacket
Second Tacket	Third Tacket

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

Example:

3	10
20	30

Input line: LABEL \OPER \OPERAND \COMMENT

Line in buffer:

```
LABEL␣␣␣␣OPER␣␣␣␣␣␣OPERAND␣␣␣COMMENT
1        10        20        30
```

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

<u>Note</u>: If word 5 is used for tabulation, the required length in word 2 must contain both the characters <u>and</u> the blanks in the tackets, i.e. for the above example word 2 must be filled with 36.

WAIT FOR AN EVENT

Calling Sequence

```
LDKL  A8,ECBADR
LKM
DATA  2
```

where:

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

Use

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

Note:

It is recommended not to use a Wait request inside a scheduled label routine, as this causes the whole program to be blocked temporarily.

EXIT

Calling Sequence

```
LKM
DATA 3
```

Use

This request is used to specify the end of a program. The program exit is effected after completion of all I/O operations and after all labels, if any, have been scheduled. A scheduled label exit passes control to the next scheduled label, if one is present, otherwise control passes to the main program.

In batch processing, register A7 contains an exit code in its right character (see under each monitor request).
If bit 0 of A7 is set to 1, the current job will be aborted up to the following JOB or BYE␣BYE command.

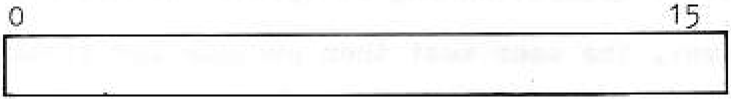
GET BUFFER REQUEST

Calling Sequence

```
LDK    A7, LENGTH
LKM
DATA 4
```

where:

LENGTH is the length, in characters, to be allocated to the buffer area (maximum 32k characters):

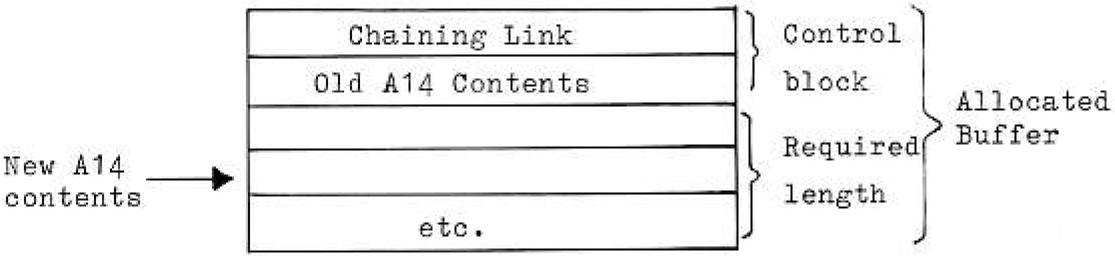


If 0 is loaded into A7, the monitor will return the highest address of memory in A7, except when the memory size is 32k, in which case 0 will be returned in A7.

Use

By means of this request, the user can allocate a memory area for temporary use, in the dynamic memory allocation area.

When the allocation is made, a control block is created by the system at the beginning of the allocated area. This block will contain a chaining link and the old contents of the A14 register:



The user must not destroy this control block

Upon completion of the request, the system responds as follows:

A7 = 0:	the buffer is allocated
= 1:	there is no memory space available (bit 0 in LENGTH = 0)
A14:	contains the address of the fourth word of the allocated buffer, so that, as soon as the buffer is allocated, the user may give a Call Function instruction with the A14 register without having to update the A14 register first.

However, the user must then provide for stack handling.

RELEASE BUFFER REQUEST

Calling Sequence

```
LDKL   A14, BUFADR
LKM
DATA 5
```

where:

BUFADR points to the second word in the buffer as given in register A14 after the Get Buffer request.

Use

To release the memory space previously reserved by a Get Buffer request. The A14 register is reloaded with the value it contained before the Get Buffer request was made.
The system responds as follows:

```
A7 = 0:   the memory space is released.
    -1:   the memory space has already been released.
```

If the A14 pointer is incorrect, or if the buffer area has been destroyed, the system issues a Halt.

PAUSE

Calling Sequence

```
LDK    A7,MESLGT
LDKL   A8,MESBLK
LKM
DATA 6
```

where:

MESLGT is a constant, specifying the length of a message which the program may output on being put in Pause state (in characters).
MESBLK is the address of a message block containing the message to be output.

Use

This request causes a temporary halt of the running program. It is put in wait state and, if specified, a message is printed out on the operator's typewriter. The program can be restarted only by an operator control message RS. When the program is restarted, it may be given an additional parameter in register A7 (see RS operator control message).

Notes:

- It is very useful if the user mentions, in the output message, that the program now is in pause state.
- The message must start with a control character.

KEEP CONTROL ON ABORT CONDITION

Calling Sequence:

```
LDKL A7,PARAM
LDKL A8,LABADR
LKM
DATA 7
```

where:

LABADR is the address of a user label to be scheduled on abort. PARAM is the address of a 3-word block which will receive the parameters of the abort condition. It has the following format:

	Abort Code	0
Aborted PSW		1
Aborted Address (A0)		2

where:

- the abort code has the same meaning as in the regular AB abort message.
- word 1 gives the PSW of the aborted program.
- word 2 gives the address where the abort took place (register A0 contents).

Use

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

The abort code returned in the PARAM block may be one of the following:

01: simulated routine save area overflow

02: non-available instruction used

04: buffer area destroyed or block was bigger than 16k words

05: label could not be scheduled

07: buffer overflow

08: disc overflow

09: disc queue overflow

0A: memory overflow during loading phase.

<u>Note</u>:

An operator abort (AB) can never be blocked.

LOAD A SEGMENT

Calling Sequence:

```
LDK    A7,NUMBER
LDKL   A8,ADDRESS
LKM
DATA 9
```

where:

NUMBER is the number of the segment which is to be loaded, as specified in the SEG control command. ADDRESS is the loading address of this segment.

Use

If a program is built up in segments and to be executed in an overlay structure, this request is used in the root segment or any segment, to load another segment into memory. For more details, see the section Programming and CCI control command SEG. The system response to the request can be found in A7:

A7 = 0: loading has been completed
A7 = 1: parameter error, e.g. segment number error
A7 = 2: memory overflow (segment too large)
A7 = 3: error in loading address.