PHILIPS

Module M11

PTS 6000 TERMINAL SYSTEM

USER LIBRARY

DOS6800 System Software
Programmer's Reference Manual

Data
Systems

Date : July 1977

Code : 5122 993 38631

MANUAL STATUS SURVEY

Module M11 "DOS6800 SYSTEM SOFTWARE"

This issue comprises following updates :

- U1.38631.0778 (July 1978; complete revision for Release 3.2).

PREFACE

The following software is supplied with the PTS6000 Terminal System:

- DOS6800 System Software
- TOSS System Software

This Manual describes the use of DOS6800 System Software (release 3.2).

The use of those parts of DOS6800 System Software designed specifically for CREDIT or Assembler program development are not described in this Manual. Information on these topics can be found in the Assembler PRM (M06) and in the CREDIT PRM (M04).

Information about TOSS System Software which is needed by programmers is included in the Assembler PRM and in the CREDIT PRM.

LIST OF CONTENTS

6. CONTROL COMMANDS

1. INTRODUCTION

1.1 General

DOS6800 System Software is a single user conversational operating system. It provides all the facilities needed for the creation, updating, compiling, assembling and linking of application programs. However, testing of application programs must be done under TOSS System Software (which is not discussed in this Manual). TOSS System Software comprises the following components:

- TOSS Monitor
- CREDIT Configurator
- CREDIT Interpreter
- CREDIT Debugging Program
- TOSS Utility Programs
- Additional Functions

The TOSS Monitor must be generated for specific PTS6000 configurations. This is done via the DOS6800 utility SYSGEN.

DOS6800 System Software comprises the following components:

- DOS6800 Monitor
- Processors
- Utilities
- Catalogued Procedures

These components are discussed briefly in the following sections.

1.2 DOS6800 Monitor

The main functions of the Monitor are:

- To communicate with the user (normally via the console typewriter).
- To maintain a library of files for each user.
  These files may be of various types and are held on disk.
- To provide temporary file space on disk for the current user of the System. This file space is used for temporary work files.
- To load and run the various processors and utilities.

The Monitor runs in a series of "sessions". A session is opened when the user signs-on by keying-in his identification in reply to the prompt USERID: typed on the console typewriter. This prompt is output after the Monitor has been loaded or after the previous session has been closed.

Having opened a session the user communicates with the Monitor by keying-in "control commands". With these commands the user may create temporary files, manipulate library files, call processors and utilities etc.

At the end of a session the user must sign-off by keying-in BYE. The Monitor will then scratch all temporary files and type out the prompt USERID: so that the next session may be started.

The user identification (userid) is a unique alphanumeric code specified by the user. This code is defined in appendix A; it may be upto eight characters long. The userid is used by the Monitor to identify the library maintained for each user. A user may of course have more than one userid (and therefore more than one library).

The memory resident part of the Monitor is called the Supervisor. The non-resident part is called the Control Command Interpreter (CCI). The CCI is responsible for interpreting the control commands keyed-in by the user. When the user calls a processor or utility into execution the Monitor loads it into memory, overwriting the CCI.

During execution the user cannot communicate with the Monitor through control commands. However, the user can input certain messages directly to the Supervisor. These are called control messages. Control messages allow the user to request a few Monitor services which are sometimes needed during program execution e.g. abort the program, retry an I/O operation.

1.3 DOS6800 Processors

The processors are as follows:

- Assembler (language processor)
- CREDIT Translator (language processor)
- CREDIT Linker (pre linkage edit processor)
- Line Editor (interactive text editor)
- Linkage Editor (load module builder)

The processors are held in the system library and are called into execution by the control commands ASM, TRA, TLK, LED and LKE.

The Linkage Editor and Line Editor are described in this Manual. The Assembler is described in the Assembler PRM (M06 Part 3). The CREDIT Translator and CREDIT Linker are described in the CREDIT PRM (M04).

1.4 DOS6800 Utilities

The following utilities are available:

CPLGEN	– writes IPL segments onto a cassette.
DMPGEF	– generates a memory dump program (DUMPFD) on a flexible disk.
DMPGEN	– generates a memory dump program (DUMPER) on a cassette.
DUMPA	– lists cassette or magnetic tape blocks or flexible disk sectors, on the line printer.
JESPER	– copies programs from disk load modules to a program loading cassette.
OBX	– produces a cross-reference listing of the references between a specified number of object modules
PM6800	– formats a disk pack.
PRDUMP	– lists a cassette produced by DUMPER or a flexible disk created by DUMPFD.
RUM	– restores a userid or complete disk from magnetic tape.
SUM	– saves a userid or complete disk on magnetic tape.
SYSGEN	– generates control commands which are used to create a TOSS Monitor.
XRF	– produces a cross-reference listing of a source module.

The utilities are held in the system library; for information on running the utilities, refer to section 9.

1.5 DOS6800 Catalogued Procedures

The following catalogued procedure is supplied with DOS6800 System Software:

- $PCAS – uses CPLGEN and JESPER to create a cassette containing an IPL and a program.

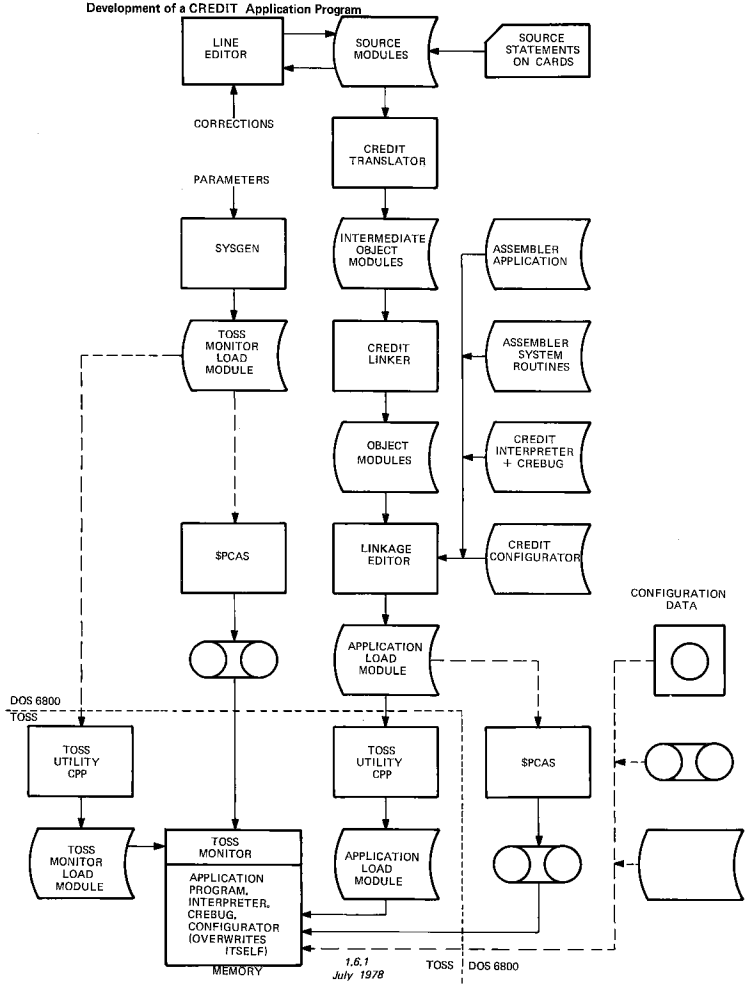
A catalogued procedure is a sequence of control commands stored as a file on disk. These control commands can be invoked by simply keying-in the procedure name on the console typewriter.

The catalogued procedure $PCAS is held in the system library. The user can easily create his own procedures which can be held in his own library.

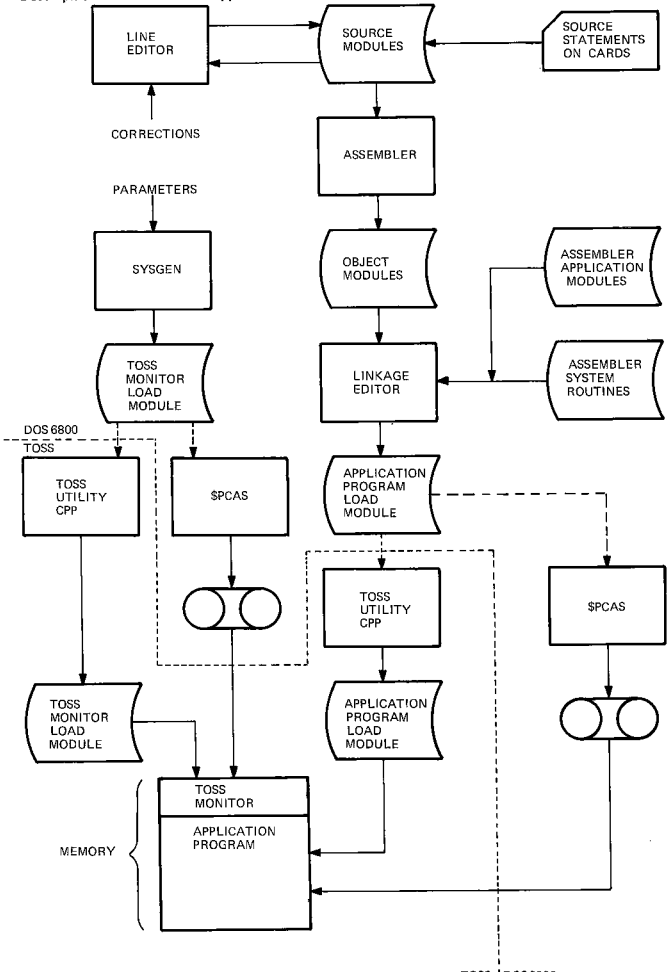
1.6 Application Program Development

The development of a CREDIT and an Assembler application program under DOS6800 System Software is shown in the following diagrams.

Development of a CREDIT Application Program



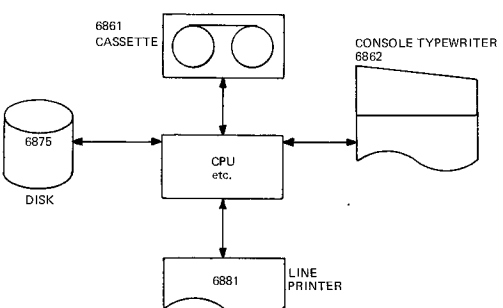
Development of an Assembler Application Program



1.7 Minimum Configuration

The minimum hardware configuration on which DOS6800 System Software can be run is as follows:

QUANTITY	TYPE NUMBER	DESCRIPTION
1	6811-001	Terminal computer
1	6815-001	System operators panel
1	6816-001	Text panel
1	6823-001	Core memory 16K
1	6861-001	Cassette unit
1	6862-001	Console typewriter
1	6875/6876	Disk unit 2 x 2.7M bytes/2 x 2.5M bytes
1	6844-001	Channel unit for disk
1	6881	Line printer 200 lpm
1	6847-001	Channel unit for line printer and card reader
1	6827	Multiplex processor
1	6833-001	Channel unit for magnetic tape cassette device

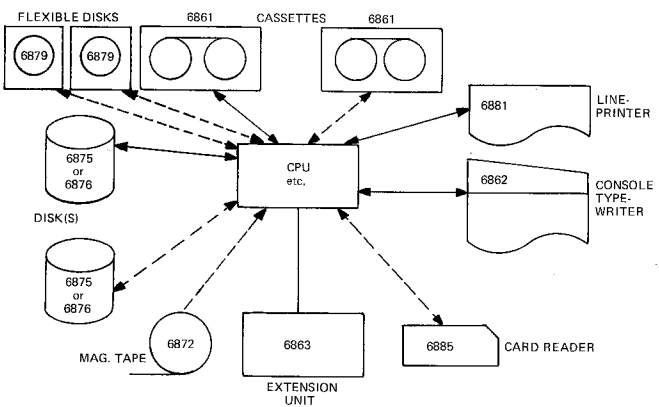


Note: It is possible to replace the cassette unit with a flexible disk drive, if required.

1.8 Maximum Configuration

The maximum hardware configuration which can be controlled by DOS6800 System Software includes the items listed in the previous section plus the following items:

QUANTITY	TYPE NUMBER	DESCRIPTION
1	6861-001	Cassette unit
1	6823-001	Core memory 16K
1	6815-002	Computer full panel
2	6875	Disk unit 2 x 2.7M bytes
1	6872-001	Magnetic tape unit
1	6842-001	Channel unit for magnetic tape
1	6885	Card reader
1	6863	Extension Unit
2	6876	Disk unit 2 x 2.5M bytes
2	6879	Flexible disk



Note: The 6875 and 6876 disk units may be used in a mixed configuration, to a maximum of two drives in total.

1.9 Notation Conventions

The following symbols (Backus Normal Form) are used to define the syntax of control commands, control messages, Line Editor commands and SYSGEN responses:

: : =	is defined as
␣	space
[ ]	the syntactic item between the square brackets may be omitted.
{ }	select one of the items between the braces.
a|b	select either a or b. This has the same meaning as braces. It is used with long item strings.
||	concatenate the syntactic items at either side of this symbol.
. . . .	elipsis indicates that the last syntactic item may be repeated.

These symbols are used throughout the Manual. They are also used in the parameter syntax definition in appendix A.

Note that the following conventions are used in this Manual when it is necessary to differentiate between alphabetic and numeric 0.

Alphabetic – Ø
Numeric – 0

1.10 Special Key-in

The user communicates with the Monitor and certain processors or utilities via the console typewriter. The following keys have a special meaning:

Normal PTS6862 Symbol	Hexadecimal code	Meaning
Λ	/5E	Erase current line. The line of information being keyed-in is discarded and the user may re-key the complete line.
_ (upper case zero)	/5F	Erase last character. The last character keyed-in is discarded. "N" consecutive erase characters may be keyed-in to erase the last "N" characters.
(LF)	/0A	Line feed. The paper is fed one line.
(CR)	/0D	Carriage return The next character will be typed in column one.

Notes:

- The above symbols may be different if a non standard keyboard is used on the PTS6862 console typewriter. However, the hexadecimal codes for those symbols will be the ones shown above.
- Each line of input keyed-in by the user on the console typewriter must be followed by a (CR). The (CR) may optionally be preceded by a (LF). This ensures that any error messages typed out by DOS6800 System Software appear on a separate line. If the (LF) is not keyed-in error messages may be unreadable, because they will be typed over the input line.

2. DOS6800 SYSTEM START

2.1 Introduction

System start is the initialization process which prepares a PTS6000 Terminal Computer for application program development. It comprises the following steps:

- Load the DOS6800 Monitor into memory.
- Key-in the date and time on the console typewriter.
- Select page size for line printer.
- Select conversational mode.
- Key-in the userid.

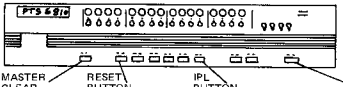
2.2 The System Disk

A system disk contains a copy of all DOS6800 System Software, held in the correct format for starting and running the System.

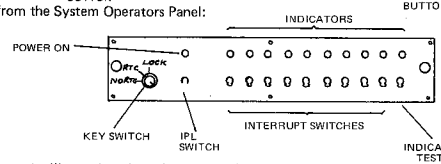
There must be at least one system disk on disk drive 1. This is the disk from which the Monitor will be loaded. In fact, all of the disks in use may be system disks. However, only the system disk from which the Monitor is loaded is recognized as **the** system disk while the System is running.
A system disk may also contain user files.

2.3 Loading the Monitor

The Monitor is loaded from the Computer Full Panel, or the System Operators Panel (SOP) if the particular computer has no Full Panel.



or from the System Operators Panel:



Note: the illustration above is an example of one type of Full Panel; exact layout may differ slightly between models.

The procedure for loading the Monitor is as follows:

- Ensure that the power is switched on at the Terminal Computer and at each peripheral device.
- Insert a cartridge in the disk drive, press the START button and wait for the READY indicator to light.

- If the Terminal Computer has a Full Panel **and** a System Operators Panel (SOP), press the Reset (RST) button, then the Master Clear (MC) button, then the Initial Program Load (IPL) button on the Full Panel.
- If the Terminal Computer has only a SOP, press the IPL switch on the SOP.
- If the system disk is the cartridge press SOP switch 2 (the leftmost switch is counted as zero). If the system disk is the fixed disk press SOP switch 3.
- The Monitor will then be read into Memory.

2.4 Initializing the Monitor

As soon as the Monitor is loaded it will type out the prompt DATE: The user may reply with either

DD MM YY (CR)
YY MM DD (CR)
or simply (CR)

Where DD, MM and YY are 2 digits giving day, month and year, separated by a delimiter, which may be any character on the keyboard.

The Monitor then types out the prompt TIME: The user must key-in the time as follows:

HH MM (CR)
or simply (CR)

Where HH, MM are decimal digits specifying hour and minute. The SOP key must be in the RTC position.

The Monitor then types out the question LINES/PAGE ON LP: The user must key-in a decimal number indicating the number of lines per page to be printed on the line printer. The default value is 40.

The Monitor then types out the question BATCH PROCESSING: The user must key-in N or (CR) which imply conversational mode.

The Monitor then types out the prompt USERID: The user must key-in his userid in one of the following ways:

{/disk-number, userid
{userid

/disk-number is the hexadecimal identification number of the system disk or user disk (see appendix A for the permitted values). Note that the system disk may be the fixed or the cartridge disk, so the disk numbers /F0, /F1, /F2 and /F3 need not always refer to the same physical disk.

If the first of the above formats is keyed-in the Monitor will look for the userid only on the disk with the specified disk number.

If the second format is keyed-in the Monitor will look at both disks starting with disk /F0.

If the userid SYSTEM is keyed-in (with no disk number) a "system session" is started. During a system session the user may key-in certain administrative commands in addition to the commands permitted in a "user session". These commands are:

DCU	Declare a new userid
DLU	Delete a userid
LIC	List catalogue of userids
PRC	Print catalogue of userids
RSU	Replace Supervisor

It is recommended that the DCU command be used with discretion. A confusingly large number of userids in a System should be avoided. The existence of unused userids may lead to human errors and possibly to wasted disk space. Two particular uses of the userid should be avoided. These are:

- declaring the same userid on two or more disks;
- declaring the same userid as that used by the Monitor.

The declaration of the same userid on two or more disks may be made by specifying the disk number in each DCU for this userid. The reply to the USERID: prompts must then include the relevant disk numbers from this userid.

The userid used by the System is SAG. However, when the user wishes to reference files in the system library, he must specify SYSTEM as the userid. That is, SYSTEM is the **external** system userid, which must be specified in certain control commands; SAG is the **internal** sytem userid, which is not normally referred to by the user. It is recommended that neither SAG nor SYSTEM be declared by the user on any disk.

An example of the initialization dialogue is shown below:

```
DOS6800 REL. 3.3
DATE  :  01 01 77
TIME  :  09 30
LINES/PAGE ON LP: 66
BATCH PROCESSING  ?  N
USERID:  BVD
S :
```

2.5 Error Reports

In cases of errors, the following messages may be output:

- INPUT COMMAND I/O ERROR

An I/O error has been detected during the reading of the user identification. The user must type in a new userid on the typewriter.

- I/O ERROR

An I/O error has been detected during the loading of the disk allocation table from the disk into memory. The user must type his userid again on the typewriter.

- USERID UNKNOWN

The userid specified has not been found on any of the disks. The user must type in a new userid on the typewriter.

3. DISK FILES

3.1 User Files

3.1.1 *Temporary and permanent files*

The Monitor maintains a library of disk files for each declared userid. A single library is not permitted to span more than one disk. It is possible to declare the same userid on more than one disk and thus create two libraries with the same userid. However, these will be treated by the Monitor as entirely separate libraries.

Library files are "permanent". That is they remain in existence until explicitly deleted by user. A separate set of "temporary" disk files may also be used for the duration of a session. Files created during a session are always considered temporary, i.e. they will be scratched at the end of the session. However, the contents of temporary files can be made part of the library at any stage during a session through the control command KPF (keep file). The contents of temporary files may also be deleted at any time through the control command SCR (scratch).
Temporary files are always held on the same disk as the user library.

3.1.2 *File types*

Files are classified as "source", "object", "load" or "undefined". These file types are explained in the following paragraphs.

A source file contains a single module of CREDIT or Assembler source language. It may be created from any input device by keying-in the control command RDS (read source). It may be updated by the Line Editor.

An object file contains one or more object modules produced by the CREDIT Translator, the CREDIT Linker or the Assembler.

A load file contains a single load module generated by the Linkage Editor. A load module is a program which is ready for execution.

An undefined file contains data which is usually neither a program nor part of a program (though it may contain a source, object or load module). It may be created from any input device by keying-in the control command RDA (read data). It may be updated by the Line Editor.

Temporary files which are made part of the library (KPF) retain their original file types.

3.1.3 *File assignment*

The user may assign temporary disk files of undefined type during a session. This is done via the control command ASG (assign). These files are empty until the user writes data to them (see section 6.6 for more details).

In addition, certain control commands automatically assign temporary disk files. For example, if the user keys-in RDS (read source) a temporary source file will be automatically assigned. Data will then be read from the source input device and will be written to the temporary source file (see sections 4.1 and 6.4 for more details). These files are given the names /S, /Ø or /L. The /S file is a temporary source file. The /Ø file is a temporary object file. The /L file is a temporary load file.

Note that the mnemonics /S, /Ø and /L may have a different meaning in control commands DEL, LST, PCH and PRT. In these commands the mnemonics may be used together with a file name to specify a source, object or load library file. This difference is important and should be noted. Apart from the description of the above commands in section 6.12 the mnemonics /S, /Ø and /L will be used to refer to the temporary files only.

At any time there is only one /S file, one /Ø file and/or one /L file in existence. This means, for example, that if two RDS commands are keyed-in the second RDS command overwrites the data written by the first RDS command.

The automatic assignment of /S, /Ø and /L files means that the user normally does not have to explicitly assign temporary disk files. However, in certain commands the user may, if he wishes, specify that a file other than the /S, /Ø or /L file must be assigned.

3.1.4 *Keeping files*

As mentioned above, the control command KPF (keep file) is used to incorporate temporary files into the users library. When a file is made permanent it may be given the file name specified in the KPF command. The format of this name is defined in appendix A; it may be upto six characters long. Alternately, if the file being kept is a source or load module, the module name may be automatically taken as the file name. The format of a module name is also defined in appendix A; it may be upto six characters long.
The relationship between temporary files and permanent files is discussed in the following paragraph.

In the case of temporary user (i.e. undefined) files and the /S and /L files the KPF control command registers each file as an individual permanent file, deleting any file of the same name that was previously in the library. This is not the case for the /Ø file: only one permanent object file exists for each userid. If a /Ø file is made permanent the contents of the file are added to the users permanent object file, deleting any module of the same name that was previously in the permanent file.

The permanent object file must be specified in certain control commands. The following mnemonics are used for this purpose:

/ØB	denotes the complete object file of the current userid (control commands DEL, LSD and PRD)
/ØBJCT	denotes the complete object file of the specified userid (control command INC)

In the remainder of this Manual the mnemonic /ØBJCT will be used to refer to the permanent object file.

3.2 System Files

The system library has the same structure as a user library. It contains the DOS6800 Monitor, processors, utilities and catalogued procedures. Control commands may be used to interrogate or modify the system library.

However, when referring to the system library in a control command either the current userid must be SYSTEM (a system session) or the userid SAG must be specified in the control command.

In addition to the system library a system disk normally contains several TOSS System Software libraries. These libraries are used for generating TOSS Monitors and link editing application programs. They have the same structure as user libraries. The userids of these libraries are TOSSUTIL, INT:PROD and TOSSWORK.

4. FILE CODES, DEVICE NAMES AND DISK NUMBERS

4.1 File Codes & Device Names

In order to maintain flexibility DOS6800 processors, utilities and control commands refer to I/O devices by using a logical file code rather than a physical device name. The relationship between file codes and devices is indicated in a file code table maintained within the Monitor. The contents of this table can be altered by the user explicitly or implicitly by the use of control commands.

A file code is a hexadecimal number ranging from /1 to /EF. The / (oblique) indicates that this code is hexadecimal. This convention is used for all hexadecimal numbers which appear in this Manual. Note that there are the following exceptions to this rule:

/S	denotes a source file
/Ø	denotes an object file
/L	denotes a load file
/ØB	denotes the object library
/ØBJCT	denotes the object library

A device name comprises a unit type and unit address.
The unit type comprises two alphabetic characters. The unit address comprises two hexadecimal digits. The device names used in the file code table are shown below:

Unit Type	Unit Address	Description
CR	0D	Card reader
LP	0F	Line printer
MT	0C	Magnetic tape unit
TK	0E	Cassette unit 1
TK	1E	Cassette unit 2
TY	10	Console typewriter

Note that the file code table contains device names for peripheral devices (i.e. non-disk devices) only.
When a file code is assigned to a disk file, a disk-number is entered against the file code indicating the disk on which the file is held. There is a separate table of files maintained by the Monitor for each disk.

The initial contents of the file code table (i.e. at system start) are shown below:

File Code	Use	Device Name
/1	Response from CCI (S:)	TY10
/2	Print output	LP0F
/3	Cassette output	TK0E
/4	Not assigned	– – –
/5	Undefined	TY10
/6–/DF	Not assigned	– – –
/E0	Control command and Line Editor input	TY10
/E1	Source input	TK0E
/E2	Object input	TK0E
/E3	Not assigned	– – –
/E4	Magnetic tape I/O	MT0C
/E5	Card input	CR0D
/E6–/EE	Not assigned	– – –
/EF	Response from Supervisor following an interrupt (M:)	TY10

In this table certain file codes have been "assigned" to peripheral units; other file codes have not been assigned. Those file codes which have been assigned are known as "standard file codes". They are the file codes assumed by the CCI if the user does not specify a file code in certain control commands.

The user may explicitly assign a file code to a device by using the control command ASG. This command will override any previous assignment for the specified file code, including any assignment for a standard file code (see section 6.6 for examples).

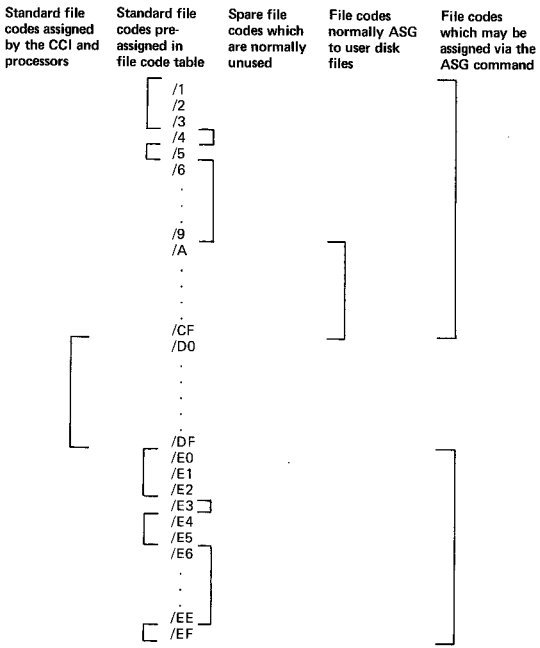
The ASG command may only be used with file codes /1—/CF and /E0—/EF. In this range file codes /4, /6—/9, /E3 and /E6—/EE will normally not be used. /1—/3, /5, /E0—/E2, /E4, /E5 and /EF are standard file codes. The remaining file codes /A—/CF will normally be assigned by the user to any temporary disk files which are required.

File codes /D0—/DF are also "standard file codes". However, they are reserved exclusively for use by the System.
File codes /D0—/DF are assigned automatically by the CCI and by the processors, as a result of the user keying-in certain control commands. These file codes are assigned in the following way:

File Code	Use	Device
/D0	Catalogued procedure input	Disk
/D1—/D2	Not used	— —
/D3	Undefined	— —
/D4	/S file	Disk
/D5	/Ø file	Disk
/D6	/L file	Disk
/D7	System /ØBJCT file	Disk
/D8	User /ØBJCT file	Disk
/D9—/DF	Undefined	— —

File code /EE is the only other file code which is assigned automatically by the System. This file code is used by the CCI when processing a catalogued procedure invokation (see section 10.2). Although this file code may be assigned explicitly by the user, this should be avoided. The contents of any disk file with file code /EE will be lost whenever a catalogued procedure is invoked.

The way in which file codes are used is summarised in the following table.


Standard file codes assigned by the CCI and processors
Standard file codes pre-assigned in file code table
Spare file codes which are normally unused
File codes normally ASG to user disk files
File codes which may be assigned via the ASG command
/1
/2
/3
/4
/5
/6
/9
/A
/CF
/D0
/DF
/E0
/E1
/E2
/E3
/E4
/E5
/E6
/EE
/EF

4.2 Disk Numbers

Values /F0 to /F3 should not be used as file codes. These values are "disk numbers". They identify entire disks rather than individual files. The use of disk numbers /F0 and /F1 depend upon which disk was used to load the Monitor during system start. The disk selected on the SOP when the Monitor is loaded is designated /F0. The other disk on the same spindle is designated /F1. Disk numbers are used as follows:

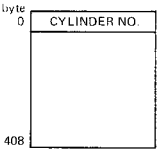
Disk Number	Use		Device Name	Description
/F0	System disk	or vice	DK28	Fixed disk 1
/F1	User disk 1	versa	DK08	Cartridge disk 1
/F2	User disk 2		DK38	Fixed disk 2
/F3	User disk 3		DK18	Cartridge disk 2
/F8	User flexible disk 1		FL09	Flexible disk 0
/F9	User flexible disk 2		FL19	Flexible disk 1
/FA	User flexible disk 3		FL29	Flexible disk 2
/FB	User flexible disk 4		FL39	Flexible disk 3

Values /F4—/F7 and /FC—/FF are reserved for future use.

5. DISK STRUCTURE

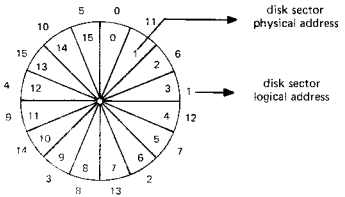
5.1 Sectors

Each disk track is divided by the hardware into 16 sectors of 205 words each. The basic unit in the structure of a disk is the sector. All I/O operations are performed on whole sectors. All sectors on the disk are numbered relative to the start of the disk (sector 0). A sector has the following format:



A cylinder number is written in the first word of every sector when the disk is initially formatted by the utility PM6800. It contains the number of the cylinder in which the sector is located. It is used by the Monitor to check if a seek operation has been successful or not.

There are two sector addressing systems used by the Monitor : physical addressing and logical addressing. Using physical addresses, sectors are numbered according to their physical sequence on the disk. Using logical addresses, sector numbers are interlaced on a "factor minus three" basis. This is done to give programs enough time to process the current sector before reading or writing the next sector. The following diagram illustrates this point:



Physical addresses are used by the Monitor when reporting disk errors on the console typewriter. In all other cases the logical address is used. For example, all data management functions refer to logical addresses and when a disk dump is made the sectors are printed in their logical sequence.

5.2 Granules

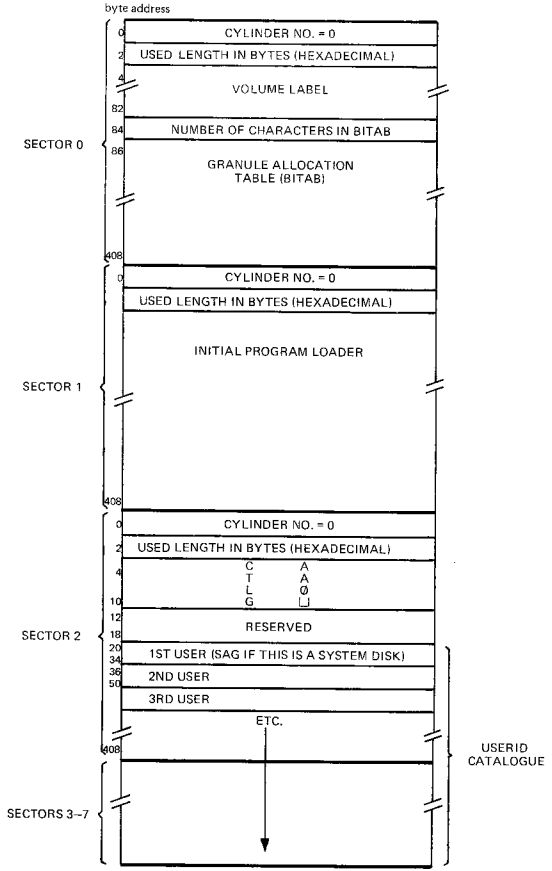
Disk space is allocated in discrete granules. Each granule comprises eight logically consecutive sectors. There are two granules per track.

The first three granules on every disk are reserved by the Monitor. The use of these granules is described in the following sections.

5.3 Granule Zero

5.3.1 *General*

The layout of this granule is shown in the following diagram:


byte address
0
CYLINDER NO. = 0
2
USED LENGTH IN BYTES (HEXADECIMAL)
4
VOLUME LABEL
82
84
NUMBER OF CHARACTERS IN BITAB
SECTOR 0
86
GRANULE ALLOCATION TABLE (BITAB)
408
0
CYLINDER NO. = 0
USED LENGTH IN BYTES (HEXADECIMAL)
INITIAL PROGRAM LOADER
SECTOR 1
408
0
CYLINDER NO. = 0
2
USED LENGTH IN BYTES (HEXADECIMAL)
4
C A
T A
L Ø
G L
10
12
RESERVED
SECTOR 2
18
20
1ST USER (SAG IF THIS IS A SYSTEM DISK)
34
36
2ND USER
50
3RD USER
ETC.
408
USERID CATALOGUE
SECTORS 3–7

The significant entries in this granule are:
- Volume label
- Granule allocation table
- Initial program loader
- Userid catalogue

The volume label and initial program loader (IPL) are written on the disk when it is premarked by the utility PM6800. The IPL is used to load the Monitor into memory from the disk during system start.
The granule allocation table and userid catalogue are described in the following sections.

5.3.2 *Granule allocation table*

This table is held in sector 0 of granule 0. The status of each granule on the disk is recorded in this table. Each bit in the table corresponds to one disk granule. Bits set to 0 indicate that the corresponding granules are in use. Bits set to 1 indicate that the granules are free. The granule allocation table is named BITAB.

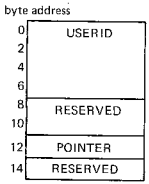
This table is copied into memory at system start. It is updated in memory. It is written back to disk after a KPF (keep file) or DEL (delete file) command is issued.

At the start of a session disk space allocation begins from the first available granule. New granules are allocated in ascending order. No backward search is done to take into account any granules which may have been released as a result of file deletion. However, the SCR (scratch) command will result in granule allocation starting again at the first available granule.

A library file (i.e. a file that has been made permanent via the KPF command) cannot be extended. That is the number of granules allocated to the file cannot be increased. Only temporary files may be extended. The user must therefore create a temporary file each time he updates a library file. This is done automatically by the Line Editor processor. The new temporary file may then be kept (KPF command) in place of or in addition to the original library file.

5.3.3 *Userid catalogue*

The catalogue is held in sectors 2 to 7 of granule 0. It contains an entry for each userid on the disk, including SAG if the disk is a system disk. Each entry contains the userid and a pointer to the file directory for that userid. Entries have the following format:



The pointer contains the sector number of the granule containing the file directory. If the userid is SAG the pointer will point to sector 8 (the first sector of granule 1). This is because the file directory for the SAG library is always held in granule 1.

Whenever a new user is declared via the DCU command an entry is made in the catalogue. It remains in the catalogue until the userid is deleted via the DLU command.

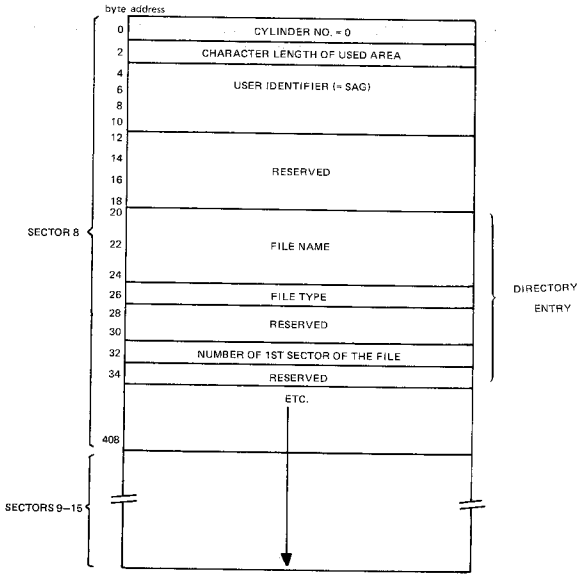
Entries are placed in the sequence in which DCU commands are keyed-in, starting at byte 20 of sector 2.
A catalogue may contain a maximum of 149 entries (i.e. the capacity of 6 sectors less 20 bytes for header).

The last entry in the catalogue is followed by a word containing /FFFF. When a userid is deleted the first word of the entry is set to /0000.

5.4 Granule One

If the disk is a system disk granule 1 contains the file directory for the SAG library. If the disk is not a system disk granule 1 is not used.

The format of this granule (when used) is as follows:



There is one eight word entry in the directory for each file in the library of userid SAG. Each entry contains the file name, the file type and the number of the first sector of the file. The format of the entry is shown in the above diagram (bytes 20 to 35).

The file type can have the following values:

- SC meaning source file
- ØB meaning object file
- LM meaning load file
- UF meaning undefined file type.

There is also a file directory for each non system userid declared on a disk. These user directories may be located anywhere on disk. Their locations are indicated by the pointers in the userid catalogue. The directories for user libraries have exactly the same format as the system file directory, except for cylinder no., sector no. and userid. Each directory occupies one complete granule.

Whenever a disk file is made permanent via the KPF command an entry is made in the directory. It remains in the directory until the file is deleted via the DEL command.

Entries are placed in the sequence in which KPF commands are keyed-in, starting at byte 20 of the granule. A directory may contain a maximum of 199 entries (i.e. the capacity of eight sectors less 20 bytes for the header).

The last entry in a directory is followed by a word containing /FFFF. When an entry is deleted the first word of the entry is set to /0000.

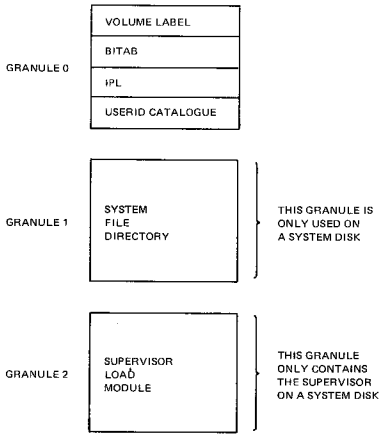
The granule for a directory is allocated at the time a new userid is declared.

5.5 Granule Two

On a system disk, granule 2 and subsequent contiguous granules hold the Supervisor load module which is read into memory by the IPL at system start.

5.6 Summary of Granules Zero, One and Two

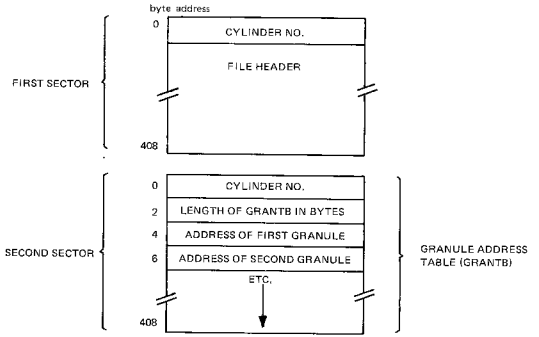
In summary, the contents of granules 0, 1 and 2 of a disk can be depicted thus:



5.7 Files

A file is held in an integral number of granules, so one granule cannot be shared by two or more files. The Monitor allocates as many granules as are needed during the creation of a temporary file. These granules need not be contiguous. They are chained and the address of each granule is entered in a granule address table called GRANTB.

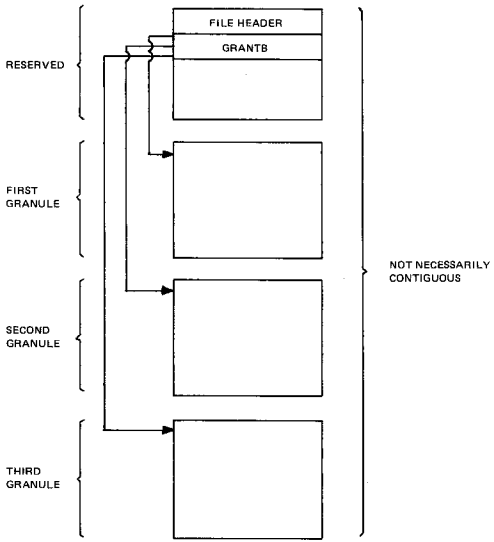
The first two sectors of the first granule of each file are reserved by the Monitor. The first sector contains a file header. The second sector contains the table GRANTB.



A GRANTB is set up each time a disk file is assigned. As the file is created the address of each granule used by the file is entered in the table. The last address in GRANTB is followed by a word containing /0000.
The granule address is the sector number, relative to the start of the disk, of the first sector in the granule.

Because GRANTB is limited to the length of one sector the maximum length of any file is 320K words (i.e. the capacity of 200 granules).

The structure of a file is therefore:


FILE HEADER
GRANTB
RESERVED
FIRST GRANULE
SECOND GRANULE
THIRD GRANULE
NOT NECESSARILY CONTIGUOUS

5.8 System Files

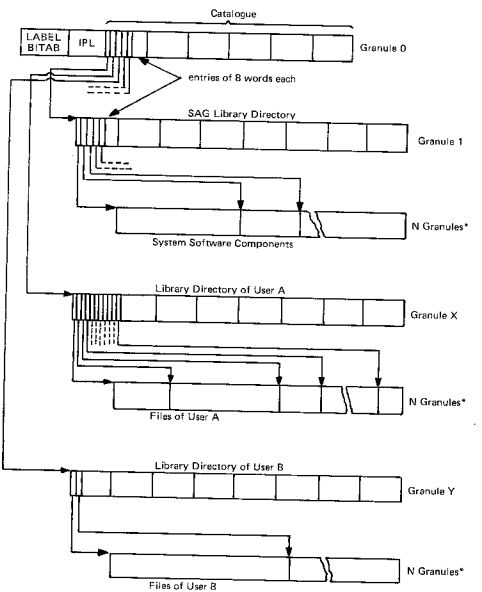
DOS6800 System Software components are held in disk files in the same manner as user programs and data.

There is one constraint on the storage of system files: the first file in the library of user SAG must start in granule 2 of the disk and must occupy contiguous granules. This file is the Supervisor load module. However, the format of the Supervisor is that of a normal load module and apart from the above restriction it is held in a normal file.

Other DOS6800 System Software components are also held in the library of the user SAG but the granules they occupy need not be contiguous.

5.9 Summary of Disk Structure

The structure of the disk is summarized in the following diagram.



*These granules are not necessarily adjacent.

6. CONTROL COMMANDS

6.1 Introduction

There are two ways in which the user can communicate with the Monitor: control messages and control commands.

Control messages are only used during the execution of a processor or utility (see chapter 7). Control commands are used at all other times.

Most control commands are used either to handle files and I/O devices or to call into execution processors or utilities. The remainder of the commands are used for general administration e.g. declare new userid (DCU), list catalogue (LIC).

Control commands may be input to the Monitor on any input device. The file code used for control command input is /E0. The default device associated with this file code is TY10, the console typewriter (see file code table in section 4.1). However, file code /E0 may be reassigned to any other input device via the ASG command. For example /E0 could be assigned to a library disk file containing a sequence of control commands. When such an assignment is made control passes immediately from the console typewriter to the assigned device. The last record read on the assigned device must be a reassignment of file code /E0 to the typewriter. When this reassignment occurs control is passed back to the console typewriter and further control commands can be keyed-in.

A string of control commands can also be held on disk in a special format known as a "catalogued procedure". A catalogued procedure can be invoked by simply keying-in the procedure name on the console typewriter. At the end of the procedure, control is automatically handed back to the typewriter. This is a more flexible method of invoking control command because the basic control commands in the catalogued procedure can be automatically modified according to parameters keyed-in by the user. Catalogued procedures are discussed in chapter 10.

Control commands are processed by the Control Command Interpreter (CCI). At the start of each session the CCI types out the prompt S: on the console typewriter. The user may then key-in a control command. If the user calls into execution a processor or utility the copy of the CCI in memory is overwritten. When the processor or utility terminates, the CCI is loaded into memory from disk and again types out the prompt S:. The user can then key-in further control commands.

The general format of a control command is as follows:

command␣[parameter [,parameter] . . . . .]

"command" is a three letter mnemonic which specifies the basic command to be obeyed. This mnemonic may be followed by one or more "parameters". The significance of individual parameters depends upon the command mnemonic used.

Each command entered via the console typewriter must be followed by a carriage return (CR). It is possible to follow a control command with a comment. This facility will normally only be used in catalogued procedures. The last parameter must be separated from the comment by at least one space.
The syntax of each parameter used in control commands is defined in appendix A.

6.2 Summary of Commands

This section comprises a complete list of control commands. Related commands are grouped under generic headings. These groups of commands are discussed in later sections.

- Input commands:
  - RDA – Read data from the specified input device.
  - RDO – Read an object module from the object input device.
  - RDS – Read a source module from the source input device.
- Output commands:
  - PCH – Write a source or undefined module to the specified output device.
  - PLD – Write a load module to the specified output device.
  - POB – Write an object module to the specified output device.
- Userid commands:
  - DCU – Declare new userid.
  - DLU – Delete a userid.
  - LIC – List catalogue of userids.
  - PRC – Print catalogue of userids.
- File code commands:
  - ASG – Assign a file code to an I/O device.
  - LSF – List all assigned file codes and I/O devices.
  - SCR – Scratch files.
- Disk commands:
  - DEL – Delete a file.
  - DUF – Dump a file onto the printer.
  - INC – Include an object module.
  - KPF – Keep file.
  - LSD – List directory.
  - LST – List a file (may also be used for a tape file).
  - MOV – Move a file.
  - PRD – Print directory.
  - PRT – Print a file (may also be used for a tape file).
  - SVD – Save disk onto another disk.
  - SVU – Save user files.
- Magnetic Tape and Cassette Commands:
  - FBS – File backward space.
  - FFS – File forward space.
  - PLB – Print label.
  - RBS – Record backward space.
  - REF – Rewind file (may also be used for a disk file).
  - REW – Rewind tape.
  - RFS – Record forward space.
  - ULD – Switch device to manual.
  - WEF – Write end of file mark (may also be used for a disk file).
  - WEV – Write end of volume mark.
  - WLB – Write label.

- Processor and utility commands :

  ASM – Assemble a source module.
  LED – Update a source module.
  LKE – Link edit object modules.
  RUN – Run a utility.
  TLK – Convert CREDIT intermediate object code into object code.
  TRA – Translate a source CREDIT module into intermediate object code.
  UPR – Call a special version of the Assembler processor.
  XRF – Run the XRF utility

- Non Typewriter commands :

  MES – Send a message to the console typewriter
  PSE – Send a message to the console typewriter and enter a pause state.

- Miscellaneous commands :

  BYE – End a session
  RSU – Replace supervisor
  SKF – Skip forms on the line printer

6.3 Input Commands

RDA, RDO and RDS are used to read data from an input device and write the data to disk. The standard input devices are listed in the file code table in section 4.1. They are:

File Code	Use	Device Name
/E1	Source input	TK0E
/E2	Object input	TK0E

/E1 is the default input file code used for the RDA and RDS commands. /E2 is the default input file code used for the RDO command. However, the user can specify a different input file code in any of these commands if the default value is not suitable.

Alternatively the devices associated with /E1 and /E2 can be changed to any other input device via the ASG command. Normally file code /E1 would be assigned by the user to the card reader, so that the RDS command would read from cards. However, it is also likely that the user would assign file code /E1 to the console typewriter so that a source program could be keyed-in directly.

Source input records must be no more than 80 characters long. Source input records longer than 80 characters are truncated, whereas shorter records are augmented by spaces up to the 80th character. If the input device is the console typewriter each line must be terminated by a (CR). If a source program is being keyed-in on the console typewriter all field separators in a source statement must be keyed-in as back slash (\) instead of space.

The last record in the input file must be an end of file record. That is the first four characters of the last input record must be :EOF. If the file is being input on the typewriter the last line of input must be :EOF (CR). If the file is being input on the card reader the last card must contain :EOF in columns 1 to 4. And so on.

The RDO and RDS commands always write the input data to the /Ø and /S files respectively. The RDA command writes the input data to the disk file specified by the user.

When the :EOF record is encountered during an RDA or RDS command an end of file mark is written at the end of the disk file. A subsequent RDS command or a subsequent RDA command specifying the same disk file code will overwrite the data read during the previous RDS or RDA command.

A subsequent RDO command, however, does not necessarily overwrite data read in by a previous RDO command. When an :EOF record is encountered during an RDO command an end of file mark is **not** written at the end of the /Ø file. The file is left open so that further RDO commands can be issued to add object modules to the end of the /Ø file. An end of file mark is only written to the /Ø file when one of the commands WEF, KPF /Ø, POB or LKE is issued. When this is done a subsequent RDO command will overwrite the contents of the /Ø file.

The above rules for writing to the /S and /Ø files may also apply when a processor writes to these files. If the output file of the Line Editor is the /S file the previous contents of this file will be overwritten. Similarly the Assembler and CREDIT Translator will overwrite the /Ø file if this file has been closed. If the file is not closed the output of the Assembler and CREDIT Translator will be added to the end of the /Ø file.

The above rules for writing to the /Ø file also apply to the INC command. This command selects an object module from a library and includes it in the /Ø file. Thus the /Ø file can be built up by a mixture of RDO and INC commands and processor calls. The output of the CREDIT Linker will always overwrite the /Ø file.

6.4 Output Commands

The commands PCH, PLD and POB are used to write a module or file from disk to a specified output device. The default output file code is /3 (standard for cassette).

The resulting module or file can then be read back to disk using the RDA, RDO or RDS command.

These commands are often used when files must be transported between PTS installations. The files are written to cassette in one installation, the cassette is carried to another installation and is then copied back to disk.

The permitted combinations of disk file types and commands is shown below:

File type to be output	Output Control Command	Input Control Command	File type recreated on disk
Object	POB	RDO	Object
Load	PLD	RDO	Object
Source	PCH	RDS/RDA	Source/undefined
Undefined	PCH	RDA/RDS	Undefined/source

Note that a **load** module must be copied back to disk via the RDO command. This means that the resulting object module must be re-link edited to regenerate the load module.

6.5 Userid Commands

DCU and DLU are used to declare and delete userids respectively. LIC and PRC are used to list the catalogue of userids.

These commands may only be issued during a system session.

6.6 File Code Commands

These commands are ASG, LSF and SCR.

The ASG command is used to alter the contents of the file code table (described in section 4.1). The command may be used to:

- assign one of the file codes /1—/CF or /E0—/EF to a peripheral unit or disk file
- equate two file codes so that they refer to the same peripheral unit or disk file.

This command will override any previous assignment for the specified file code, including any standard assignments in the file code table.

The user may not assign file codes /D0—DF and is strongly dissuaded from assigning file code /EE (see section 4.1). The only other restrictions on the use of file codes are ones that are obvious to the user e.g. the line printer should **not** be assigned as an **input** file.

An example of a "standard file code" assignment is as follows:

```
ASG   /E1,CR0D
```

This command will change the peripheral unit, associated with file code /E1 from TK0E (cassette unit 1) to CR0D (card reader). /E1 is the source input file code. The result of this change is that when the user issues an RDS (read source) command the source program will be read from cards rather than from cassette.

An example of a "spare file code" assignment is as follows:

```
ASG   /A,DK
```

This command will associate a free area of disk on the disk containing the users library with the file code /A. The user may then write a temporary file to this disk area. A later KPF (keep file) command will incorporate this file into the users library.

Note that a device name is not specified when assigning a disk file. Only the characters DK are used. The Monitor enters the disk number of the disk containing the user's library in the file code table. This points to an additional table of files maintained by the Monitor for each disk.
The actual device names used for disk are shown in the table of disk numbers described in section 4.2.

Before writing to the /S, /Ø, or /L file or to a temporary user disk file a file code must be assigned. However, such an assignment need not be made explicitly by the user. When the user issues a command which results in the creation of a temporary disk file the System automatically assigns the required file code if it is not already assigned. The commands to which this applies are listed below.
The files which are assigned are also shown. Note that the /S, /Ø and /L files have file codes /D4, /D5 and /D6 respectively; temporary user files may have file codes /1—/CF and /E0—/EF.

Type of file created	Command	File assigned
Source	RDS	/S file
Source	LED	/S or temporary user file
Object	RDO	/Ø file
Object	INC	/Ø file
Object	ASM	/Ø file
Object	TRA	/Ø file
Object	TLK	/Ø file

Type of file created	Command	File assigned
Load	LKE	/L file
Undefined	RDA	temporary user file
Source/load /undefined	MOV	/S, /L or temporary user file

The ASG command may also be used to equate two file codes. One of these file codes must be assigned to a device. The entry in the file code table for the other equated file code will point to the assigned file code. This means that the file codes may be used interchangeably. A reference to either file code will result in a reference to the same I/O device. A maximum of seven file codes can be assigned to a disk file in this way.

Temporary files may also be "scratched" at any time during a session via the SCR control command. The effect of this command, basically, is to nullify the specified disk file code assignments. That is, the relevant entries in the file code table are removed and the associated disk files are lost (unless they have been made part of the library). Individual files may be scratched by specifying /S, /Ø, /L or a file code in the SCR command. If an individual file is not specified then all temporary disk files are scratched. The way in which files are scratched is described in the following table:

File Code	Use	Effect of SCR Command
/A—/CF	Normally assigned to user disk files	File codes are de-assigned. That is the file code is no longer associated with the physical device. The contents of the file are lost and the file must be re-assigned if it is to be used again.
/D0—/DF	Standard file codes assigned by CCI and processors (including /S, /Ø and /L)	
/1, /2, /3 /5, /E0, /E1, /E2, /E4, /E5, /EF	Standard file codes pre-assigned in the file code table	If the user has not assigned the file code to disk the SCR command has no effect.
/4, /6—/9, /E3, /E6—/EE	Spare file codes which are normally unused	If the file code has been assigned to disk it is de-assigned as described above. An exception to this rule is made for file code /E0. If this file code has been assigned to disk it is equated to file code /1.

Note that the above file code groupings are the same as those used in the summary table at the end of section 4.1.

At the end of a session all temporary files are automatically scratched (BYE command).

In addition to the ASG and SCR commands the LSF (list file codes) command may be issued at any time during a session. This produces a listing of all assigned file codes and their associated devices.

6.7 Disk Commands

The disk commands enable the user to perform the following disk file maintenance functions:

- Make temporary files permanent (KPF)
- Copy library files (INC and MOV)
- Save library files or complete disks (SVU and SVD)
- Print the contents of files or file directories. (DUF, LSD, LST, PRD and PRT)
- Delete files (DEL)

KPF is the only command in the above list that warrants further explanation in this section. To understand the effect of the KPF command it is necessary to understand the way in which library files are used and the relationship between temporary files and library files. These topics are discussed in the remainder of this section.

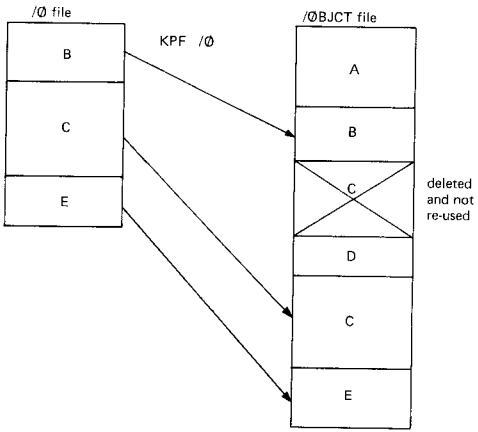
In the case of temporary user files and the /S and /L files the KPF command makes the temporary file permanent by entering the file name in the users file directory. After the KPF command has been carried out the temporary file may still be referenced by using the file code or the /S or /L mnemonics. It may also be referenced by using the file name. These types of reference will both point to the same physical area of disk. This is because the file is not physically moved during the KPF process.

However, as soon as an attempt is made to write to the /S or /L file (e.g. by an RDS command) a new area of free disk space will be assigned. Thus a new temporary file may be created without interfering with the file which has been made permanent.

If the file is a temporary user file it may not be written to after a KPF command. If a SCR command is issued for such a file the file code is simply deassigned (explained in section 6.5) and the permanent file is unaffected.

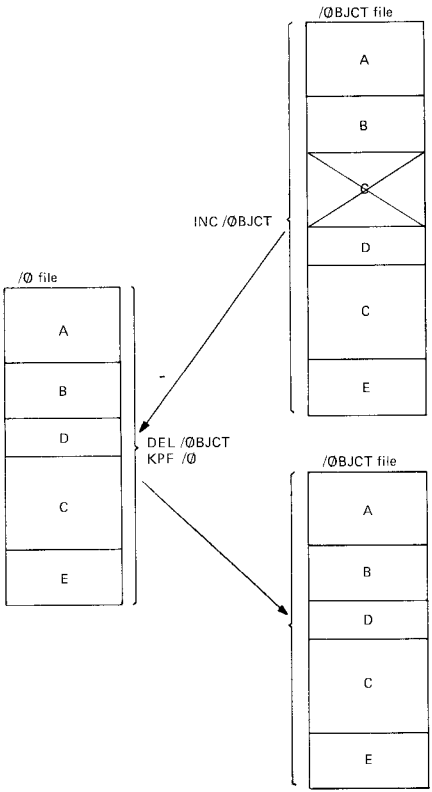
It can be seen from the previous paragraph that **one** permanent file exists for **each** KPF command which specifies a temporary user file or the /S or /L file. This is not the case for the /Ø file. Only one permanent object file (/ØBJCT) exists for each userid. When the /Ø file is made permanent via a KPF command the object module(s) in the file are physically copied to the users /ØBJCT library file. If the /ØBJCT file contained object modules prior to the KPF command the new modules will be added to the file after the existing modules. If one of the existing modules has the same name as one of the new modules the existing module is deleted. If possible, the new module is copied into the space that was occupied by the deleted module. If the new module is too large it will be added to the end of the file with the other new modules.

It can be seen that the above process can result in the /ØBJCT file containing an increasing amount of unused file space due to the replacement of object modules with larger ones. This situation is depicted in the following diagram:



In the above example the new module B is the same size as the old module B and is thus written over the old module. The new module C is larger than the old module C so the old module is deleted and the new module is written after the last module in the file. The new module E did not exist in the /ØBJCT file prior to the KPF. It is therefore added at the end of the file.

Periodically the user should reorganise his /ØBJCT file to remove unused space. This can be done by moving the contents of the /ØBJCT file to the /Ø file (INC command), by deleting the /ØBJCT file (DEL command) and by copying the /Ø file back to the /ØBJCT file (KPF command). This is depicted in the following diagram:


/ØBJCT file
A
B
C
D
INC /ØBJCT
D
C
E
/Ø file
A
B
D
C
E
DEL /ØBJCT
KPF /Ø
/ØBJCT file
A
B
D
C
E

6.8 Magnetic Tape and Cassette Commands

In this section the term "tape" will be used to refer to both half inch magnetic tape and magnetic tape cassettes.

Commands FBS, FFS, RBS, REF, REW and RFS are used to position the tape. Command REF can also be used with a disk file in which case the "next record pointer" is moved to the beginning of the specified disk file. The file can then be read from the beginning or re-written. The other commands may only be used with tape.

Commands WLB and PLB are used respectively to write and print tape labels. If a label is written to a tape it must be printed and checked manually when the tape is read. The CCI does not perform any checks. The label is written as a normal file, terminated by a tape mark. After the PLB command has read the label the tape is positioned at the start of the next file. The layout of the tape label is as follows:

VOL1	Volume serial number	Security code	Reserved	Owner identification

The remaining commands are WEF, WEV and ULD.
WEF and WEV are used respectively to write an end of file mark and an end of volume mark. ULD is used to switch a tape unit to manual. WEF can also be used with a disk file.

Before reading from a cassette (RDA, RDO and RDS) the tape should be rewound (REW) to ensure that reading will begin from the start of the tape.

6.9 Processor and Utility Commands

The Line Editor (LED command) is used to update a source file held in the user's library. The updated file is written to the /S file or to a temporary user disk file.

The Assembler (ASM command) is used to convert source assembly language into object code. The CREDIT Translator (TRA command) is used to convert source CREDIT language into intermediate object code. The Translator and Assembler read source language modules either from the /S file or from a user library file. The output is written to the /Ø file.

The CREDIT Linker (TLK command) converts intermediate object code into object code. Part of the intermediate code is read from the /Ø file. All output object code is written back to the /Ø file overwriting the input code.

The Linkage Editor (LKE) generates a load module from a number of input object modules. Part of the object code input is read from the /Ø file. All of the output load module is written to the /L file.

The RUN command is used to call a utility into execution.
The UPR command is used to call into execution a special version of the Assembler processor.

6.10 Non Typewriter Commands

These commands are used in catalogued procedures and other types of "non typewriter" control command files to communicate with the user. The MES command simply types a message on the console typewriter. The PSE command types a message and enters a pause state. To continue processing the user must issue a restart (RS) control message (see section 7.5).

6.11 Examples

6.11.1 *Introduction*

The following examples have been selected because they are typical of the kind of control command sequences used during program development. The commands are not described in detail. The objective of these examples is merely to give the user an impression of the type of control command sequences which he will find useful. A full explanation of each type of command is given in section 6.12.

Each example comprises a control command sequence, an explanation and a flow diagram.

6.11.2 *Development of a CREDIT load module*

Control commands:

```
1     USERID:ABC
2     ASG     /E1,CR0D
3     RDS
4     KPF     /S
5     TRA     /S
      LED     MØD3,/S
6     !!DL    6
      !!EN
7     KPF     /S
8     TRA     /S
9     KPF     /Ø
10    SCR     /Ø
11    TRA     MØD1
12    KPF     /Ø
13    TLK     U,M,X
14    LKE     N,M
15    KPF     /L,LØAD
16    $PCAS   A=LØAD
17    BYE
```

Explanation:

1 Sign-on

2 Assign source input file code (/E1) to card reader (CR0D).

3 Read source module into /S file.

4 Keep source module using the file name specified in the program IDENT (say MØD2).

5 Translate source module held in /S file and write object module to /Ø file.

6 Edit the next source module, which is held in library file MØD3 and write updated module to /S file. Updates are keyed-in on the typewriter.

7 Keep source module using file name MØD3.

8 Translate source module in /S file and add object module to /Ø file.

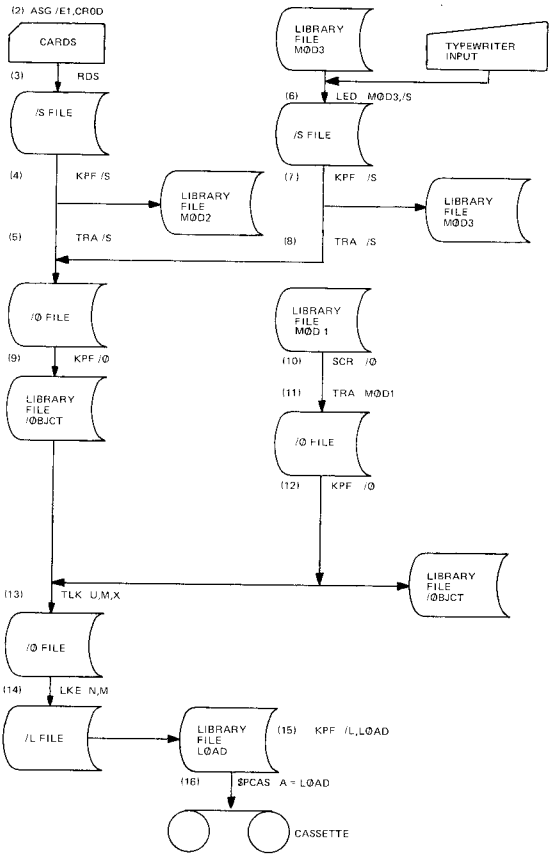
9 Keep object modules MØD2 and MØD3 in ØBJCT file.

10 Scratch the contents of /Ø file.

11 Translate the next source module, which is held in library file MØD1 and write object module to /Ø file. This is the main module of the program which contains the data division.

12 Keep object module MØD1 in /ØBJCT file (Note MØD1 has been translated and kept as a separate module to ensure that the Linker processes the modules in the correct sequence).
13 Link the intermediate code modules held in the /Ø and /ØBJCT files and write the linked modules to the /Ø file.
14 Link edit the modules in the /Ø file and write the resulting load module to the /L file.
15 Keep the load module using file name LØAD.
16 Copy the load module to cassette.
17 Sign off.

Flow diagram:



6.11.3 *Catalogued procedure for Link Editing*

Control commands:

```
1     LED     M:PRØC,/S /file-code
2     !!IL
3     $LINK
4     SCR     /Ø
5     INC     M1,XYZ
      INC     M2,XYZ
      INC     M3,XYZ
      INC     M4,XYZ
6     LKE     N,M
7     KPF     /L,PRØG
8     END
9     !!EN
10    KPF     /file-code, M:PROC
11    $LINK
```

Explanation:

1 Edit catalogued procedure file M:PRØC and write updated file to a temporary user file. Procedure is keyed-in on the typewriter.

2 Insert the following lines at the start of the file.

3 Name of the catalogued procedure. Note: The following control commands on lines 4 to 7 are used to link edit an assembler program comprising 4 object modules M1, M2, M3 and M4. The control commands will not be obeyed until the catalogued procedure is invoked (line 11). They are merely treated as data by the Line Editor.

4 Scratch the contents of the /Ø file.

5 Copy the object modules M1, M2, M3 and M4 from the /ØBJCT file of userid XYZ to the /Ø file.

6 Link edit the modules in the /Ø file and write the resulting load module into the /L file.

7 Keep the load module using the file name PRØG.

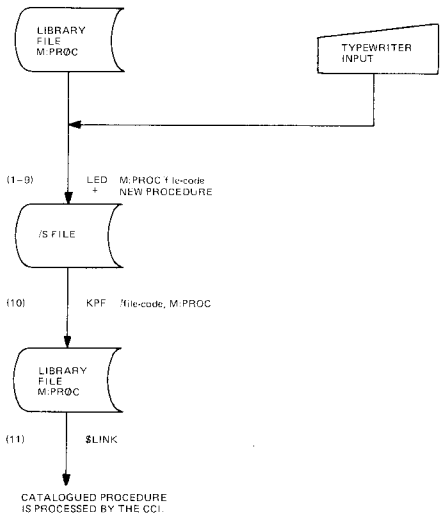
8 End of catalogued procedure.

9 Terminate the Line Editor.

10 Keep the new file of catalogued procedures from the temporary file using file name M:PROC (thus overwriting the previous version of M:PROC).

11 Invoke the new catalogued procedure $LINK.

Flow diagram:



6.11.4 *Creation of a Data Cassette from Typewriter*

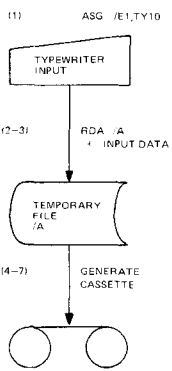
Control commands:

```
1    ASG     /E1,TY10
2    RDA     /A
3   { data
    { :EOF
4    WEF     /3
5    PCH     /A
6    WEF     /3,2
7    ULD     /3
```

Explanation:

1 Make console typewriter the source input device by assigning file code /E1 to the typewriter (overriding the default device).
2 Read data to disk file /A (automatically assigned).
Input is taken from the typewriter.
3 Lines of data, terminated by an :EOF.
4 Write end of file mark on cassette.
5 Write file /A to cassette.
6 Write two end of file marks on cassette.
7 Rewind cassette and switch cassette drive to manual.

Flow diagram:



6.11.5 *Copying Object Files from Disk to Cassette*

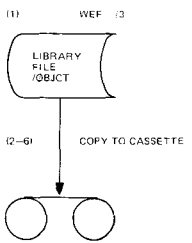
Control commands:

```
1    WEF    /3
2    PØB    ØB1
3    WEF    /3
4    PØB    ØB2
5    WEF    /3,2
6    ULD    /3
```

Explanation:

1 Write end of file mark on cassette (cassette is default device for file code /3).
2 Write object module ØB1 to cassette.
3 Write end of file mark on cassette.
4 Write object module ØB2 to cassette.
5 Write two end of file marks to cassette.
6 Rewind cassette and switch cassette drive to manual.

Flow diagram:



6.11.6 *Copying Object Files from Cassette to Disk*

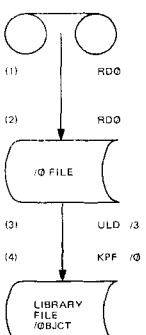
Control commands:

```
1    RDØ
2    RDØ
3    ULD     /3
4    KPF     /Ø
```

Explanation:

1 Read first object module into /Ø file.
2 Read second object module into /Ø file.
3 Rewind cassette and switch cassette drive to manual.
4 Keep /Ø file (i.e. incorporate in /ØBJCT file).

Flow diagram:



6.11.7 *Creating a special version of the Assembler Processor*

Control commands:

```
1    MØV    ASM,/L,SAG
2    ASG    /20,DK
3    PLD    ASM,/L,/20
4    REF    /20
5    RDØ    /20
6    ASM    GENMØD
7    LKE    N,M
8    KPF    /L,ASM
9    UPR    ASM,APPMØD
```

Explanation:

1 Copy the load module ASM (i.e. the Assembler processor) from the library with userid SAG (system library) into the /L file.

2 Assign file code /20 to disk.

3 Copy the load module in the /L file to the disk file /20 converting the file to a single module object file.

4 Move next record pointer in file /20 to the beginning of the file.

5 Read the object module in file /20 into the /Ø file.

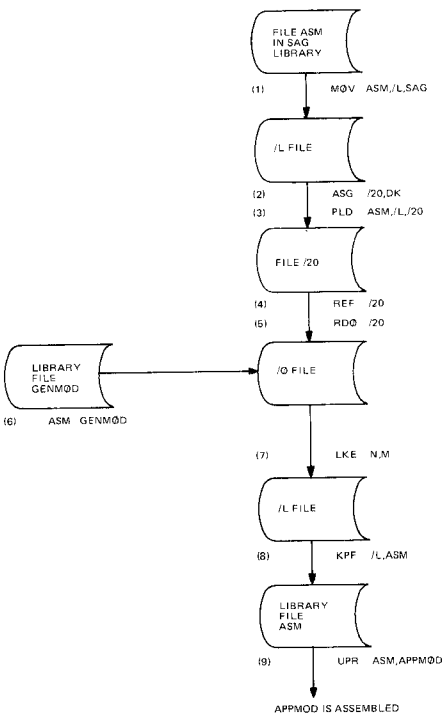
6 Assemble the module GENMØD which defines new Assembler instructions (i.e. extending the instruction set, see Assembler PRM module M06 GEN directive). The object module produced by the Assembler is added to the /Ø file.

7 Link Edit the Assembler processor containing the new module. The load module produced by the Link Editor is written to the /L file.

8 Keep the load module in the /L file (i.e. the new Assembler processor) in the users library.

9 Execute the new Assembler processor for the application source module APPMØD.

Flow diagram:



6.12 Control Command Reference

This section describes the syntax and use of each control command. The syntax for each parameter in these commands is given in appendix A. The notation conventions are described in section 1.9.

Error reports are listed for each control command.
Further error reports may be found in appendix B.

ASG

ASSIGN A FILE CODE

ASG

Syntax : ASG␣/file-code-1,{NO / /file-code-2 / device name / DK}[,file-name[,userid[,/disk-number]][,NP]]

Use : This command is used to assign a file code to a peripheral unit or a disk file or to equate two file codes so that they refer to the same peripheral unit or disk file. Further information concerning the ASG command may be found in section 6.6.

/file-code-1 : *File code which is to be assigned. Only file codes /1–/CF and /E0–/EF may* be used in an ASG command.

NO : If this parameter is used, no device will be assigned and I/O operations on this file will be ignored by the Monitor.

/file-code-2 : If this parameter is used, it is equated to file-code-1. This means that the file *codes may be used interchangeably. A reference to either file code will result* in a reference to the same peripheral unit or disk file. A maximum of seven file codes can be assigned to a disk file in this way.

device-name : If this parameter is used, file-code-1 is assigned to the peripheral unit specified here by two alphabetic characters for the unit type and two hexadecimal digits for the unit address. If the device is the disk, only DK need be specified, without address.

file-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. (Such an assignment is made if CCI input is to be taken from a disk file.) If DK is used without this parameter, the file code will be assigned to a temporary disk area.

userid, disk-number, NP : These parameters are used only when name is specified. They enable a user to assign a file code to a file in another user's library on the disk 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.

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 disk)
LFT OVERFLOW (disk logical file description table overflow)
FILE NAME UNKNOWN
DISK OVERFLOW (no free granule available to allocate to temporary disk file)
TOO MANY FILE CODE EQU (more than 7 file codes have been assigned to the same disk file)

ASM

ASSEMBLER PROCESSOR

ASM

Syntax : ASM␣ { /S / file name } [,NL]

Use : This command is used to assemble a source program from the /S file or from a library file. The object code is written to the /Ø file. If a fatal error occurs during assembly the /Ø file will be deleted. Further information concerning the Assembler may be found in the Assembler PRM (M06).

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

file-name : The source program to be assembled can be found in the library file specified by file-name.

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 console typewriter as well.

Errors : FILE NAME ERROR
FILE NAME MISSING
/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)
/Ø ASSIGN ERROR (an attempt to assign the temporary object file /Ø 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 assembler has not been catalogued)
INVALID PARAM

BYE

TERMINATE SESSION

BYE

Syntax : BYE

Use : This command is used to terminate a session. All temporary files will be automatically scratched and the CCI will type out USERID: so that a new session may be started.

DCU

DECLARE NEW USERID

DCU

Syntax : DCU␣userid,/disk-number

Use : The DCU command enables a user to declare a new userid. This command may only be used during a system session. The new userid is added to the userid catalogue of the disk specified. The format of the userid is defined in appendix A. It may be up to eight characters long. A directory granule is allocated to this user and initialized with /FFFF (the last entry in a directory is always followed by a word containing the value /FFFF) and the granule 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 disk number is not in the range from /F0 to /FF).
DISK FILE CODE ABSENT (the disk number is not present in the command)
DISK NOT OPERATIONAL (the disk is not ready)
USERID ALREADY CATALOGUED (the userid specified has already been catalogued previously on the disk specified)
CATALOG OVERFLOW (too many userids have been catalogued on the disk specified)
DISK I/O ERROR (an I/O error has been detected during a read/write operation to/from disk)
DISK OVERFLOW (no free granule is available to be allocated to the userid directory)
TOO MANY PARAM (an illegal parameter follows the disk number)
COMMAND NOT ALLOWED (the current session is not a system session)
DISK FILE CODE UNKNOWN

DEL

DELETE LIBRARY FILE

DEL

Syntax : DELL␣ { file-name / module-name / /ØB } [, { /S / /Ø / /L }]

Use : This command is used to delete a file or object module from a library.

file-name
module-name : Indicate the name of the file or module

/ØB : Indicates that the whole object file must be deleted. If /ØB is used, /S, /Ø or /L may not be specified.

/S, /Ø, /L : Specify the type of library file (source, object or load). When file-name or module-name is used as the first parameter and no second parameter is specified, the file type is assumed to be un-defined.

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

DLU — DELETE USERID — DLU

Syntax : DLU[ ]userid,/disk-number

Use : This command can only be used in a system session. It is used to delete the specified userid from the userid catalogue. The corresponding file directory granule and all the granules of the library files are released and the allocation table of the disk is updated. The DLU command may not be used to delete the first user on the disk with disk number /F0 (SAG).

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 disk 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 disk)
DISK NOT OPERATIONAL (the disk 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 disk 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 / /Ø / /L / /disk-number / file-name } [,/decimal-number-1 [,/decimal-number-2]]

Use : This command is used to get a hexadecimal dump, on the print unit, of a library file, a temporary file or a whole disk.
Users must be careful about the position of the file after a DUF command. They must rewind it before using it again. This means that the following command sequence will give an error.

```
ASM ␣ /S
DUF ␣ /Ø
LKE
```

After assembly, the /Ø 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. 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 ␣ /Ø
DUF ␣ /Ø      DUF ␣ /Ø
              LKE
```

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

/file-code : Specifies the file which must be dumped.

file-name : Name of a library file which must be dumped. This is a file of undefined type.

/Ø, /L : Cause dumping of the /Ø and /L files respectively. This is mainly useful for system debugging purposes.

/decimal-number-1 /decimal-number-2 : It is also possible to get a selective dump by specifying two decimal sector numbers for the beginning and ending sectors of the dump. A dump is made up to an EOF mark, up to end-of-volume (last granule) or up to decimal-number-2. The sector number is a disk sector logical address.

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 disk file)
SECTOR DELETED

FBS

FILE BACKWARD SPACE

FBS

Syntax : FBS[,/file-code[,decimal-number]]

Use : This command may only be used with magnetic tape or cassette units. It is used to move the tape backwards across the specified number of tape marks.
The default number of tape marks is one.

Errors : 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

FFS

FILE FORWARD SPACE

FFS

Syntax : FFS␣/file-code [,{decimal-number / ALL}]

Use : This command may only be used with magnetic tape or cassette units. It is used to move the tape forwards across the specified number of tape marks. The default number of tape marks on one. If ALL is specified the tape will be positioned after two consecutive tape marks.

Errors : 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

INC

INCLUDE OBJECT MODULE

INC

Syntax : INC ␣ { /ØBJCT / module-name } [,userid [,/disk-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 and to copy it into the /Ø file. After an INC command has been issued no command other than INC may be used to write to the current /Ø file.

module-name : Is the name of the object module which is to be included.

/ØBJCT : If this is specified, the whole object library will be copied into the /Ø file.

userid : Is to be used only if this object module is held in the library of a user other than the current one.

/disk-number : Disk on which the specified userid is present.

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

KPF

KEEP FILE

KPF

Syntax : KPF [,] { /S / /Ø / /L / /file-code } [, { file-name / module-name }]

Use : This command is used to incorporate into the library a file or module, which has been created as a temporary file, i.e. to make this file or object module permanent. Further information concerning the KPF command may be found in section 6.7.

/S, /Ø, /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.

file-name
module-name : Are the names which are to be given to this file or module in the library and which will be placed in the file directory. If the first parameter is /S, the file will be of the type source. If file-name is not specified, /S is assumed to contain a source program of which the file-name can be found in its IDENT statement. Otherwise the name specified is taken as the file name. If the first parameter is /L or file-code, file-name must be specified. If the first parameter is /Ø, module-name is optional. If it is not specified, all object modules of the /Ø file are kept in the user library, otherwise only the module named will be kept.

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

LED

LINE EDITOR PROCESSOR

LED

Syntax : LED ␣ file-name [, { /file-code-1 / /S } [,/file-code-2] [,XX]]

Use : This command is used to line edit a source file held in the library. The updated source file is written to the /S file or to the file specified by file-code. A detailed description of the Line Editor is given in section 8.2.

file-name : Is the name of the source file.

/file-code-1 : Is the output file code. If specified the file is written as a file of undefined type.

/file-code-2 : Specifies the input command file 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 alphabetic characters, specified if the user wants command control characters other than !!

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 ASSIGNED
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).

LIC

LIST CATALOGUE

LIC

Syntax : LIC␣/disk-number

Use : This command can only be used in a system session. It enables the user to print out the userid catalogue of the specified disk on the typewriter.

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

LKE

LINKAGE EDITOR PROCESSOR

LKE

Syntax : LKE␣ [N|S|U] [,M] [,label]

Use : This command is used to link edit a set of object modules. The object modules may be held in the /Ø file, the system /ØBJCT file and the user /ØBJCT file. The output load module is written to the /L file. If a fatal error occurs during link editing the /L file will be deleted. Further information concerning the Linkage Editor may be found in section 8.3.

N : The system or user /ØBJCT file need not be scanned.

S : Only the system /ØBJCT file has to be scanned.

U : Only the user /ØBJCT file has to be scanned.
Default value: both /ØBJCT files will be scanned.
The user /ØBJCT file will be scanned first, then the system /ØBJCT file and then the user /ØBJCT file again.

M : A print-out of the map and symbol table is required.

Label : This label identifies the start point of the program. It must be defined as an entry point in one of the modules in the /Ø file; it may be upto eight characters long. The label must not be N, S, U, M, DE or DS. If label is not specified the last start address encountered in the /Ø file will be used.

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:
/Ø EMPTY
/Ø CLOSE ERROR
PROCESSOR NOT CATALOGUED
INV. IDT (IDENT name larger than 6 characters)

LSD	LIST DIRECTORY	LSD

Syntax : LSD ␣ [/ØB]

Use : This command provides for a listing of the file directory of the user library on the typewriter. If /ØB 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. The disk numbers /F0—/F3 are also listed.

LST **LIST FILE** LST

Syntax : LST ⊔ { /file-code / /S / /S, file-name / file-name } [,line-number-1 [,line-number-2]]

Use : This command causes a listing of the specified disk or tape file on the console typewriter. If a record is longer than a print line it will be printed on several lines. 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-number-2, 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. After the listing, the file is positioned at the last record listed.

/S, file-name : a library source file.

file-name : a library undefined file.

/file-code : a temporary undefined file.

/S : the /S file.

line-number-1 : if specified these provide for a listing of the file from line-number-1
line-number-2 : to line-number-2.

Errors : FILE NAME ERROR (the first parameter is neither /S nor 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 (/1 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 EOF mark is not considered as a record, so it is not listed).

MES — SEND MESSAGE — MES

Syntax : MES␣ alphanumeric-character . . . .

Use : This command is used, especially in catalogued procedures, to have the message specified typed out to the operator.

MOV

MOVE A FILE

MOV

Syntax : MOV _ file-name, { /S | /L | /file-code } [,userid [,/disk-number]]

Use : This command is used to copy a file from a library (generally of another user) to the /S or /L file or to a file indicated by a file code. Only source or load files respectively may be moved to the /S and /L file. Any type of file may be moved to an undefined file specified by file-code.

file-name : Is the name of the library file which is to be moved.

/S : The file must be moved to the /S file.

/L : The file must be moved to the /L file.

/file-code : File code of the temporary file to which the file must be moved.

userid : User identification of the user whose file must be moved (if the file belongs to a user other than the current one).

/disk-number : Disk on which the userid is present.

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).

PCH

WRITE A SOURCE OR UNDEFINED FILE

PCH

Syntax : PCH _: { /file-code-1 | /S | file-name | file-name, /S } [,/file-code-2]

Use : This command is used to have a disk source or undefined file written to the specified output device. The maximum record length must be 132 characters. 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 record of the last granule may be wrong. Further information concerning the PCH command may be found in section 6.8.

/file-code-1 : File code of a temporary user file which must be output.

/S : The temporary source file must be output.

file-name : Name of a catalogued user data file which must be output.

file-name,/S : Name of a catalogued source program which must be output.

/file-code-2 : Output file code. Default value is /3.

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 file-name is not /S)
INPUT FILE CANNOT BE ASSIGNED (it is impossible for the sytem 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

PLB **PRINT LABEL** **PLB**

Syntax : PLB␣/file-code

Use : This command is used to print a magnetic tape or cassette label on the console typewriter. The tape is positioned at the first record of the file following the label. If the label is absent and a tape mark is encountered at the start of the tape the message NO LABEL is printed. Further information concerning tape labels may be found in section 6.8.

Errors : FILE CODE MISSING (no file code specified)
FILE CODE ERROR
INVALID FILE CODE
FILE CODE UNKNOWN
PARAM MISSING
TOO MANY PARAM
I/O ERROR

PLD

WRITE A LOAD FILE

PLD

Syntax : PLD␣ file-name [, { /L / /file-code / /L,/file-code }]

Use : This command is used to have a disk load file written to the specified output device. The file will be output in object code format and must be read back to disk via the RDO command. Further information concerning the PLD command may be found in section 6.8.

file-name : Name of a load file in a library. If /L is specified the /L file will be
/L : 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 /3.

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

POB

WRITE AN OBJECT MODULE

POB

Syntax : POB␣ [{ *module-name* / /file-code / module-name,/file-code }]

Use : This command is used to have a disk object module or file written to the specified output device. Further information concerning the POB command may be found in section 6.8.

module-name : Is the name of the library object module which must be output. If no name is specified, the whole /Ø file will be punched. When the /Ø file must be output and no EOF has yet been written after it, the EOF mark is written before the file is rewound and output.

/file-code : Output file code. Default value is /3.

Errors : OBJECT MODULE NAME ERROR
/Ø EMPTY
/Ø CLOSE ERROR (error detected during writing of EOF and rewinding of /Ø file)
/Ø INPUT ERROR (error detected during reading of /Ø file)
OUTPUT I/O ERROR (I/O error encountered during the output operation)
INPUT I/O ERROR (error detected during reading of 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.

PRC

PRINT CATALOGUE

PRC

Syntax : PRC LI /disk-number

Use : This command can only be used in a system session. It enables the user to print out on the print unit the userid catalogue contained on the disk specified.

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 :␣ [/ØB]

Use : This command causes a print-out of the user's file directory on the print unit. If /ØB 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,file-name / file-name } [,line-number-1[,line-number-2]]

Use : This command causes a listing of the specified disk or tape file on the print unit. If a record is longer than a print line it will be printed on several lines. 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-number-2, 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. After the printing, the file is positioned at the last record printed.

/S,file-name : A library source file

file-name : A library undefined file

/file-code : A temporary undefined file

/S : The /S file

line-number-1 : If specified, these provide for a listing of the file from line-number-1
line-number-2 : to line-number-2.

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 (/2 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 ␣ [alphanumeric-character . . . .]

Use : This command is used, especially in catalogued procedures, to put the machine in a pause state and have the specified message typed out for the operator. The operator must restart the program to have the next command read, via the control message RS.

RBS — RECORD BACKWARD SPACE — RBS

Syntax : RBS ␣ /file-code[,decimal-number]

Use : This command may only be used on magnetic tape or cassette units. It is used to move the tape backwards over the specified number of records. The default number of records is one.

Errors : 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

RDA — READ DATA — RDA

Syntax : RDA ␣/file-code-1[,/file-code-2]

Use : This command is used to read data from an input device and write the data to disk. The data will be read until an :EOF is encountered. Further information concerning the RDA command may be found in sections 6.3 and 6.8.

/file-code-1 : Specifies the disk file to which the data must be written. This file code does not have to be assigned by the user; an assignment will automatically be made by the CCI.

/file-code-2 : File code from which the data are read. Default value is /E1. The standard assignment for this file code is to TK0E (cassette).

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 /1 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 disk 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 disk temporary file).

RDO	READ OBJECT	RDO

Syntax : RDO [/file-code]

Use : This command is used to read an object module from an input device and write the module to the /Ø file. Further information concerning the RDO command may be found in sections 6.3 and 6.8.

/file-code : File code from which the module will be read. Default value is /E2. The standard assignment for this file code is TK0E (cassette).

Errors : INVALID PARAMETER (the file code is not a numeric value in the range from /1 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 disk 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 disk temporary file).

RDS	READ SOURCE	RDS

Syntax : RDS␣ [/file-code]

Use : This command is used to read a source module from an input device and write the module to the /S file. Further information concerning the RDS command may be found in sections 6.3 and 6.8.

/file-code : File code from which the module will be read. Default value is /E1. The standard assignment for this file-code is TK0E (cassette).

Errors : INVALID PARAMETER (the file code is not a numeric value in the range from /1 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 disk temporary file. A message will follow giving the cause of the error)
INPUT 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 disk 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).

REF — REWIND FILE — REF

Syntax : REF ␣ /file-code

Use : This command is used to position the specified file at the first record in the file. It may be used with magnetic tape, cassette and disk.

Errors :
FILE CODE MISSING (no file code specified)
FILE CODE ERROR
INVALID FILE CODE
FILE CODE UNKNOWN
PARAM MISSING
TOO MANY PARAM
I/O ERROR

REW — REWIND TAPE — REW

Syntax : REW ␣ /file-code

Use : This command may only be used with magnetic tape or cassette units. It is used to rewind the tape on the device to which the specified file code has been assigned.

Errors : FILE CODE MISSING (no file code specified)
FILE CODE ERROR
INVALID FILE CODE
FILE CODE UNKNOWN
PARAM MISSING
TOO MANY PARAM
I/O ERROR

RFS

RECORD FORWARD SPACE

RFS

Syntax : RFS␣/file-code [,decimal-number]

Use : This command may only be used on magnetic tape or cassette units. It is used to move the device indicated by file-code forward over the specified number of records. The default number of records is one.

Errors : 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

RSU

REPLACE SUPERVISOR

RSU

Syntax : RSU ␣ /disk-number

Use : This command may only be used during a system session. The command is used to replace the Supervisor (resident part of the Monitor) on a system disk. The load file containing the new Supervisor is taken from the /L file. It is copied to the disk specified by /disk-number starting from sector number 18 (sector 16 and 17 are occupied by the file header and GRANTB).

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 to the specified disk).
/FX NOT A SYSTEM DISK
/L TOO BIG

RUN	RUN A UTILITY	RUN

Syntax : RUN ␣ utility-name

Use : This command is used to call into execution a DOS6800 utility.

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

SCR

SCRATCH FILES

SCR

Syntax : SCR ␣ [{ /S / /Ø / /L / /file-code }]

Use : The effect of this command, basically, is to nullify the specified disk file code assignments. That is, the relevant entries in the file code table are removed and the associated disk files are lost (unless they have been made part of the library). The SCR command works in the following way:

File codes	Effect of SCR command
/A–/DF	File codes are de-assigned. That is the file code is no longer associated with the physical device. The contents of the file are lost and the file must be re-assigned if it is to be used again.
/1–/9 and /E0–/EF	If the user has not assigned the file code to disk the SCR command has no effect. If the file code has been assigned to disk it is de-assigned as described above. An exception to this rule is made for file code /E0. If this file code has been assigned to disk it is equated to file code /1.

Further information concerning the SCR command may be found in section 6.6.

/S, /Ø, /L, /file-code : If specified this parameter indicates the file to be scratched. If not specified all temporary disk files are scratched.

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

SKF

SKIP FORM

SKF

Syntax : SKF [,] [decimal-number]

Use : By means of this command, the specified number of pages may be skipped on file code /2. The standard assignment for this file code is LPOF (line printer). The default number of pages is one.

SVD

SAVE DISK ONTO ANOTHER DISK

SVD

Syntax : SVD ␣ /disk-number-1, /disk-number-2 ,

Use : This command can only be used in a system session. It is used to copy the contents of one disk onto another one. All allocated granules of the disk specified by disk-number-1 are copied onto the disk specified by disk-number-2. If the capacity of the disks is different, only as many sectors will be duplicated as the smaller disk contains. The disks are assumed not to contain any defective tracks, for the copying is done sector per sector, sequentially. Disk-number-2 may not be the current sytem disk (i.e. /F0). The volume label of disk-number-2 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 disk to which one of the two file codes is assigned, is not supported by the system).

SVU

SAVE USER LIBRARY

SVU

Syntax : SVU ␣ userid,/disk-number

Use : This command is used to copy all the files of the user library specified, contained on the disk specified, into the library of the current session user. This may be the same user as the one specified by userid.
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 disk 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
DISK NOT OPERATIONAL (the disk 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 disk specified)
INPUT DISK I/O ERROR (an I/O error has been detected during a read operation from the disk specified in the command)
OUTPUT DISK I/O ERROR (an I/O error has been encountered during a read/write operation from or to the disk 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 disk 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, ØB for object files, LM for load modules and UF for undefined files. This file cannot 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.

TLK — CREDIT LINKER PROCESSOR — TLK

Syntax : TLK␣ [N|S|U] [,M] [,X]

Use : This command must be used to call the CREDIT Linker. It converts the intermediate object code to the object code which can be processed by the Linkage Editor. Further information concerning the CREDIT Linker may be found in the CREDIT PRM (M04).

N : The system or user /ØBJCT files do not need to be scanned.

S : Only the system /ØBJCT file has to be scanned.

U : Only the user /ØBJCT file has to be scanned.
Default value: Both /ØBJCT files will be scanned. The user /ØBJCT file will be scanned first, then the system /ØBJCT file and then the user /ØBJCT file again.

M : The listing of the map, which consists of a listing of the module names and their loading address, and an alphabetical list of all entry points and common blocks together with their value, must be printed.

X : Indicates that a cross reference listing is required.

Errors :
INVALID PARAMETER
COMMON VALUE REDUNDANT
LIBRARY OPTION REDUNDANT
MAP OPTION REDUNDANT
START ADR. REDUNDANT
USER LIB. ASSIGN ERROR
/Ø EMPTY
/Ø CLOSE ERROR
PROCESSOR NOT CATALOGUED

TRA

CREDIT TRANSLATOR PROCESSOR

TRA

Syntax : TRA ␣ { /S / file-name } [,NL]

Use : This command is used to translate a source module from a library or from the /S file. The intermediate object code is written to the /Ø file. If a fatal error occurs during translation the /Ø file will be deleted. Further information concerning the CREDIT Translator may be found in the CREDIT PRM (M04).

/S : The source program must be translated from the /S file.

file-name : Indicates the name of a library source module or program to be translated.

NL : If this parameter is specified the Translator will produce no listing of the translated program. If NL is omitted a listing is produced on the print unit. Error messages, however, will always be listed.

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).
/Ø ASSIGN ERROR (an attempt to assign the /Ø file 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 Translator has not been catalogued).

ULD — SWITCH TAPE UNIT TO MANUAL — ULD

Syntax : ULD ␣ /file-code

Use : This command may only be used with magnetic tape or cassette units. It is used to switch the device indicated by file-code to manual. The device is automatically rewound.

Errors :
FILE CODE MISSING (no file code specified)
FILE CODE ERROR
INVALID FILE CODE
FILE CODE UNKNOWN
PARAM MISSING
TOO MANY PARAM
I/O ERROR

UPR

CALL USER PROCESSOR

UPR

Syntax : UPR ␣ module-name, $\left\{\begin{matrix}/S \\ file\text{-}name\end{matrix}\right\}$ [,NL]

Use : This command enables the user to call into execution a special version of the Assembler processor from the user library. Module-name is the name of the load module containing the special processor. The other parameters have the same meaning as under ASM. This command is useful in conjunction with the Assembler directive GEN which enables the creation of special Assemblers with predefined mnemonics (FORM) or symbols (equivalence tables) without affecting the standard Assembler in the System. Further information concerning the GEN directive can be found in the Assembler PRM (M06). An example of the control command sequence needed to generate a special Assembler is given in section 6.11.7.

WEF — WRITE END OF FILE — WEF

Syntax : WEF ␣ /file-code [,decimal-number]

Use : This command is used to write the specified number of end of file marks on the device specified by file-code. The default number of file marks is one. This command may be used on magnetic tape, cassette or disk.

Errors : 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

WEV **WRITE END OF VOLUME** **WEV**

Syntax : WEV ␣ /file-code

Use : This command may only be used with magnetic tape or cassette units. It is used to write an end of volume mark on the device indicated by file-code.

Errors : FILE CODE MISSING (no file code specified)
FILE CODE ERROR
INVALID FILE CODE
FILE CODE UNKNOWN
PARAM MISSING
TOO MANY PARAM
I/O ERROR

WLB — WRITE LABEL — WLB

Syntax : WLB␣/file-code, decimal-number,hexadecimal-digit,alphanumeric-character . . . .

Use : This command may only be used with magnetic tape or cassette units. It is used to write a volume label on the device indicated by file-code. The label is followed by a tape mark. Further information concerning tape labels may be found in section 6.8.

decimal-number: This is the volume serial number comprising upto six digits.

hexadecimal-digit : This is the security code. If the value of this code is "NO", no volume label will be written. A tape mark will be written instead.

alphanumeric-character : This is the owner. It comprises upto 39 characters including blanks and commas.

Errors :
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

7. CONTROL MESSAGES

7.1 Introduction

Control messages are the means by which the user communicates with the Monitor during the execution of a processor or utility. At all other times the user must communicate via the CCI using control **commands** (see chapter 6).

Control messages are entered by the user on the device with file code /EF. The default device associated with this file code is TY10, the console typewriter (see file code table in section 4.1). Before keying-in a control message the user must press the interrupt button (marked INT) on the Computer Full Panel.

The Monitor will suspend execution of the processor or utility and will respond by typing out the prompt M: on the console typewriter. The user may then key-in a control message. If the control message contains an error the Monitor will type-out ER. The user must then press the INT button and re-enter the message.

7.2 Format of Control Messages

The general format of a control message is as follows:

message ␣ [parameter[,parameter] . . .]

"message" is a two letter mnemonic which specifies the basic message. It may be followed by one or more parameters. The significance of individual parameters depends upon the message mnemonic used. Each message must be terminated by a carriage return (CR).

7.3 Summary of Messages

The following control messages may be used:

AB – Abort program
AS – Assign a file code
RD – Release device
RS – Restart
RY – Retry an I/O operation

7.4 Control Message Reference

This section describes the syntax and use of each control message. The syntax for each parameter in these messages is given in appendix A. The notation conventions are described in section 1.9.

AB

ABORT

AB

Syntax : AB

Use : This message is used to terminate the execution of a processor or utility. The Supervisor responds by typing out the address of the memory location at which the program stopped.

AS

ASSIGN A FILE CODE

AS

Syntax : AS␣ file-code, { device-name / NO }

Use : This message enables the user to assign a new file-code or to change an assignment made previously. This message may be of use when, for example, a printer breaks down. In this case the print file can be re-assigned to the typewriter. If "NO" is specified instead of a device name no device will be assigned and I/O operations on this file will be ignored by the Monitor.

RD

RELEASE A DEVICE

RD

Syntax : RD ␣ unit-address

Use : This message may be used when the Monitor types out a peripheral unit failure report (see appendix B) resulting from an I/O failure. The message will abandon the I/O operation on the specified device.

This message will usually result in the abortion of the control command, processor or utility performing the I/O operation.

RS

RESTART

RS

Syntax : RS

Use : This message is used to restart the System when it has been halted by, for example, an end of volume condition or a PSE control command.

RY

RETRY AN I/O OPERATION

RY

Syntax : RY ␣ unit-address

Use : This message may be used when the Monitor types out a peripheral unit failure report (see appendix B) resulting from an I/O failure. The message will cause the I/O operation to be re-tried.

If the re-try is unsuccessful the Monitor may type out another peripheral unit failure report. The user may then key-in another RY message.

If the I/O operation continues to fail the user may key-in an RD message.

8. DOS6800 PROCESSORS

8.1 Introduction

The DOS6800 processors are as follows:

- Assembler — language processor
- CREDIT Translator — language processor
- CREDIT Linker — pre linkage edit processor
- Line Editor — interactive text editor
- Linkage Editor — load module builder

The processors are held in the system library and are called into execution by the control commands ASM, TRA, TLK, LED and LKE as described in section 6.12.

The Linkage Editor and Line Editor are described in the following sections. The Assembler is described in the Assembler PRM (M06). The CREDIT Translator and CREDIT Linker are described in the CREDIT PRM (M04).

8.2 The Line Editor

8.2.1 *Introduction*

The Line Editor is an interactive text editor used for updating source program modules or data files. The following types of update may be performed:

- Search text for character string and list all occurrences (command mnemonic !!LS).
- Search text for character string and replace by new character string (command mnemonic !!CH).
- Insert text from another file into the current file (command mnemonic !!JN).
- Replace character string in specified line (command mnemonic !!RE).
- Delete specified line or lines (command mnemonic !!DL).
- Insert text after specified line (command mnemonic !!IL).

Having entered the necessary updates the user may either terminate the Line Editor normally or may abort. If the Line Editor is aborted the output file is scratched.

The Line Editor is called into execution by the control command LED. Updates are normally keyed-in on the console typewriter. The updates are applied to the library file specified in the LED command and an updated output file is created. This is written to the /S file or to the temporary file specified in the LED command. The maximum line length of these file is 80 characters.

The LED command is described in detail in section 6.12. After this command has been keyed-in the Line Editor will, if the console typewriter is being used for updating, respond with the prompt L:. The user may then key-in the required update commands.

8.2.2 *Editing phases*

The Line Editor performs all updating during a single pass of the input file. For this reason the user must key-in all commands which apply to the whole file before the update pass begins. These commands are !!LS (search for character string and list all occurrences) and !!CH (search for character string and replace). !!LS commands are executed as soon as they are keyed-in because they do not update the file. !!CH commands are stored in memory until the update pass begins.

When the Line Editor begins execution it is said to be in the "definition phase". During this phase several !!LS and/or !!CH commands may be keyed-in. When a command other than !!LS or !!CH is keyed-in the Line Editor immediately starts the update pass. This is known as the "execution phase". During the execution phase the user may key-in text insertion commands (!!JN and !!IL), text replacement commands (!!RE) and text deletion commands (!!DL). Because the Line Editor carries out a single sequential update pass the line numbers specified in these commands must be in ascending order.

No !!LS or !!CH commands may be keyed-in during the execution phase. However, any !!CH commands keyed-in during the definition phase are obeyed during the execution phase as each line of the input file is scanned.

During either the definition or execution phase the Line Editor may be terminated (!!EN) or aborted (!!AB).

Note that a KPF control command must be issued after termination of the Line Editor if the temporary output file is to be retained.

8.2.3 *Update command reference*

This section describes the syntax and use of each update command. The syntax for each parameter in these commands is given in appendix A. The notation conventions are described in section 1.9.

!!CH

CHANGE STRING

!!CH

Syntax : !! CH␣ $$ character-string-1$$ character-string-2$$

Use : Search the file for character-string-1 and replace by character-string-2. Every occurrence of character-string-1 in the file will be replaced. This command may only be used in the definition phase.

!!DL

DELETE LINES

!!DL

Syntax : !!DL ␣ line-number-1 [,line-number-2]

Use : All lines from line-number-1 to line-number-2 inclusive are deleted. If line-number-2 is omitted only line-number-1 is deleted. The user may key-in several lines of text following the !!DL command. This text will be inserted in place of the deleted text until a line beginning with !! is encountered. All deleted lines are listed on the device with file code /2 (unchanged by any !!CH command). This command may only be used in the execution phase.

!!EN

END EDITOR

!!EN

Syntax : !!EN

Use : This command terminates updating. The remainder of the input file is copied to the output file whilst any !!CH updates are applied. The Line Editor will then hand control back to the CCI. This command may be used in either the definition or execution phase.

!!IL

INSERT LINES

!!IL

Syntax : !!IL␣ [line-number]

Use : This command enables the user to insert one or more lines of text after the specified line-number. If line-number is omitted the text is insert after the current line. Lines of text are inserted until a line beginning with !! is encountered. Inserted lines are listed on the device with file code /2. They are not changed by any !!CH command. This command may only be used in the execution phase.

!!JN

JOIN AUXILIARY FILE

!!JN

Syntax : !!JN ␣[line-number-1] ,file-name,line-number-2[,line-number-3]

Use : Text from the auxiliary file identified by file-name is inserted into the file being updated after line-number-1. The file name may refer to the file currently being edited. If line-number-1 is omitted the text is inserted after the current line. The text to be inserted is specified by line-number-2 through line-number-3 inclusive. If line-number-3 is omitted only the line specified by line-number-2 is inserted. This command may only be used in the execution phase.

!!LS

LIST LINES

!!LS

Syntax : !!LS␣!$$character-string$$

Use : This command searches for all occurrences of character-string in the input file. Each line containing character-string is listed on the device with file code /2. This command may only be used during the definition phase.

!!RE

REPLACE STRING

!!RE

Syntax : !!RE␣line-number,$$character-string-1$$character-string-2$$

Use : This command replaces character-string-1 by character-string-2 in the line specified by line-number. If this replacement changes the length of the line it will be truncated or space filled on the right. Characters will not be shifted from one line to another during this process. This command may only be used during the execution phase.

8.2.4 *Error reports*

The following error reports may be printed during the execution of the Line Editor:

Message	Meaning
FILE NAME ERROR	The specified name in the update command contained an error.
FILE NAME MISSING	The specified name cannot be found on this disk.
INPUT FILE CANNOT BE ASSIGNED	This error report is followed by a report explaining the error.
/S CANNOT BE ASSIGNED	This error report is followed by a report explaining the error.
INVALID FILE CODE	The file code specified does not belong to the input device from which the update commands are input.
FILE CODE NOT ASSIGNED	The file code of the command input device must have been specified beforehand by means of ASG.
TOO MANY PARAMETERS	Too many parameters specified.
DSK INPUT ERR. UPD ABORTED	Line Editor cannot read from disk.
DSK OUTPUT ERR. UPD ABORTED	Line Editor cannot write onto disk.
UNKNOWN COMMAND. TRY AGAIN	The update command is not accepted.
I/O ERR ON LAST RECORD, TRY AGAIN	An I/O error occurred.
SEQUENCE ERR. TRY AGAIN	The line numbers in the update command are not in ascending order.
SYNTAX ERR. TRY AGAIN	The update command or the newly typed in line, contained a syntax error.
AUX INPUT CANNOT BE ASSIGNED. TRY AGAIN	The auxiliary file used in !!JN command cannot be assigned.
CMND NOT ALLOWED IN EXE MODE. TRY AGAIN	This command cannot be used in the execution phase.
TABLE O'FLOW, TRY AGAIN	The character string table is overflowing.
EOF, UPD TERMINATED	The :EOF mark has been encountered on the input source file before reaching the specified line, thus terminating the update process.

Message	Meaning
EOF IN AUXI INPUT	The :EOF mark has been encountered from the Auxiliary Input. The !!JN command is terminated but the operator continues.

When the message TRY AGAIN is printed the user has the possibility of correcting the previous command or data record from the device with file code /1. If (CR) is typed in, the input is resumed from the normal input command file.

8.3 The Linkage Editor

8.3.1 *Introduction*

The function of the Linkage Editor is to build an executable load module from a specified set of object modules.

The Linkage Editor is called into execution by the control command LKE. The input object modules are taken from the /Ø file the users /ØBJCT file and/or the system /ØBJCT file. The output load module is written to the /L file.

The LKE command is described in detail in section 6.12.

8.3.2 *Processing*

The Linkage Editor builds the load module by incorporating the relevant object modules one by one. The first object module to be incorporated is the one which contains the application program start point. This module must be held in the /Ø file and must contain as an entry point the label which is to be used as the program start point. If this label is not specified in the LKE control command the last start point found in the /Ø file will be used. This module will normally contain references to other modules. These modules may contain references to further modules, and so on. The Linkage Editor builds the load module as a hierarchy of object modules connected by entry point references.

The referenced object modules must be held in the /Ø file, the user /ØBJCT file or the system /ØBJCT file. The Linkage Editor first incorporates any referenced modules which are held in the /Ø file. If any unsatisfied references exist after this the Linkage Editor will scan the /ØBJCT file of the current userid. If there are still any unsatisfied references the Linkage Editor will scan the system /ØBJCT file and then the user /ØBJCT file again. Scanning of the user and/or system /ØBJCT files can be suppressed by including the U, S or N parameter in the LKE control command.

A number of error conditions can arise during linkage editing. These conditions and the corresponding error reports are listed below.

8.3.3 *Output listings*

An example of the listing produced by the Linkage Editor is shown below. A new page is thrown at the start of each section.
Heading information:

```
LKE U,M
DATE   /  /          TIME  00H-11M-23S-
LABEL = SAVORED      DATE = 31 08 76      PACK NBR = 000   GEERH
```

Memory map:

```
OUTP03   0008
SWRL00   0020
WRITE    0154
FUNCTN   0490
WRTLST   0718
```

Symbol table:

```
                    *** SYMBOL TABLE ***

CTLTAB 0718 R   PICTAB 0718 R   STB    0052 R   SWRL00 00F2 R
T:ADD  04AC R   T:ADDC 04A8 R   T:CMP  06A6 R   T:CMPC 06A2 R
T:CPA  069E R   T:CPAC 069A R   T:EDT  01B4 R   T:EDTZ 01B0 R
T:MOU  0602 R   T:MOVC 06FE R   T:OP1  049C R   T:OPA  0594 R
T:OPS  05D0 R   T:SUB  04A4 R   T:SUBC 04A0 R   T:WRT  0106 R
T:WRTZ 0102 R   TAWRL  0722 R   WB1    0054 R
                      ******

                START = 000C    LENGTH = 079C    REGION = 0477

:EOF
PROG ELAPSED TIME: 00H-00M-08S-700MS-
```

The heading information contains the Monitor date and time and the label, date and number of the system disk.

The listing of the memory map and symbol table is optional. It is produced if the M parameter is included in the LKE control command.

The memory map contains the name (including any comments on the IDENT line) of each object module built into the load module together with the address at which it occurs within the load module. The modules are listed in the order in which they occur within the load module.

The symbol table comprises an alphabetic list of entry point labels together with the addresses at which these labels occur within the load module. Labels which are referenced but which do not exist in the load module are included in the symbol table with **** for the address. A letter is also printed after each address indicating the type of address. These letters are:

A = Absolute
R = Relocatable
S = Internal symbol table
U = Undefined

8.3.4 *Error reports*

An error report is output on the printer whenever an error condition is detected by the Linkage Editor. Errors may be fatal or non fatal. Fatal errors are reported on the console typewriter (T), as well as the printer (P). The possible error reports are as follows:

Report	Output Unit	Abort?	Meaning
IO ERROR file ssss	T	Yes	Irrecoverable I/O error on file 'file' with status 'ssss' of the I/O unit.
BLK.COM.	P	No	Wrong optional blank common address.
BLK.DAT.name	P	No	'name' is an unknown common block name used in a Block Data Subprogram.
DBL.DEF.name	P	No	'name' is defined more than once as an entry point or as the name of a common block.
INV.LGH.name	P	No	'name' is a common block name whose length exceeds the maximum length allowed.
UNS.EXT.	P	No	There are one or more unsatisfied external references. The load module may be executable when no references are made to its externals. The externals are listed in the map.
ABS.STR.	P	No	Absolute start address (ignored).
ERR.MOD.	P	No	A link-edited module has received an error flag from assembly or compilation.
NO STRT.	P	No	Wrong (or no) start address.
INV.IDT.	T&P	Yes	Invalid IDENT record.
PRG.OVL.	T&P	Yes	Generated load module exceeds 32K words.
TBL.OVL.	T&P	Yes	Not enough space to link-edit these modules.
IDT.MIS.	T&P	Yes	IDENT record missing.
END MIS.	T&P	Yes	END cluster missing.
ERR.LKE.	T&P	No	A non-fatal error has occurred during this link-edit run.
ABS.ADR.	T&P	Yes	An absolute address was read. The Linkage Editor does not accept absolute addresses.

9. DOS6800 UTILITIES

9.1 Introduction

The DOS6800 Utilities are as follows:

CPLGEN	– writes IPL segments onto a cassette
DMPGEF	– generates a memory dump program (DUMPFD) on a flexible disk
DMPGEN	– generates a memory dump program (DUMPER) on cassette
DUMPA	– lists cassette or magnetic tape blocks. or flexible disk sectors on the line printer
JESPER	– copies programs from disk load-modules to a program loading cassette
OBX	– produces a cross-reference listing of the references between a specified number of object modules, on the line printer
PM6800	– formats a disk pack
PRDUMP	– lists a cassette produced by DUMPER or a flexible disk produced by DUMPFD
RUM	– restores a userid or disk from magnetic tape
SUM	– saves a userid or disk
SYSGEN	– generates control commands which are used to create a TOSS monitor
XRF	– produces a cross-reference listing of a source module, on the line printer

The utilities are held in the system library and are called into execution by the control command RUN, with the exception of XRF, which is called by the control command XRF. The commands may be used during a user or system session. The RUN command is described in detail in section 6.1.

CPLGEN and JESPER may be executed one after the other by invoking the DOS6800 catalogued procedure $PCAS. The remainder must be executed by keying in the RUN command, except XRF (see above), or by invoking a user-written catalogued procedure.

9.2 CPLGEN

CPLGEN is used to write a TOSS IPL onto cassette, prior to the writing of a load module from disk.

The creation of Monitor and/or Application program cassettes is normally carried out by executing the $PCAS catalogued procedure, but CPLGEN may be executed separately, if required.

The following steps must be taken to execute the CPLGEN utility:

Insert the cassette to be written to in the left-hand drive

Key in the control command RUN CPLGEN xxxxxx, where xxxxxx is the name of the Monitor that is to be copied after the IPL.

9.3 DMPGEF

DMPGEF is used to generate a memory dump program (DUMPFD) on a flexible disk, that can be used to dump the contents of memory onto another flexible disk. The flexible disk may then be printed using the PRDUMP utility.

No IPL is written to the flexible disk, because the dump program is loaded and started directly by bootstrap.

Because the dump program is loaded into memory and runs as a free standing program, it may be used even when the memory resident Monitor is inoperable. However, the dump program will be loaded into memory locations /0 – /21A if the flexible disk drive is connected on a multiplex channel, or locations /0 – /27E if it is connected on a programmed channel. The original contents of this part of memory will therefore be overwritten.

The following steps must be taken to produce a memory dump using the DMPGEF utility:

Insert the flexible disk to be written by DMPGEF in a flexible disk drive

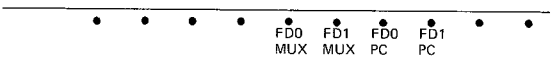
Assign file code /3 to the output device

Key in the control command RUN DMPGEF

If the Terminal Computer has a Full Panel and a SOP, press the RST button, then the MC button, and then the IPL button (see diagram in section 2.3)

If the Terminal Computer has a SOP only, press the IPL button

Press the appropriate SOP switch to load the program from flexible disk, depending on the disk drive on which the disk is mounted, and whether the drive is connected on a multiplex or programmed channel (see diagram below)



The dump program will then be loaded into memory, after which the leftmost SOP light will be illuminated

Insert a blank flexible disk in a drive and close the door

Select the disk drive being used for the dump, and press the appropriate SOP switch, where SOP switches 1 to 4 correspond to flexible disk drives 0 to 3

The contents of memory will then be dumped onto the flexible disk, after which the leftmost SOP light will be illuminated. If an error occurs, the second SOP light from the left will be illuminated, and the dumping must be repeated.

Run the PRDUMP utility to obtain a listing of the memory dump from the flexible disk (see section 9.9)

9.5 DUMPA

This utility is used to list on the device with file code /2 (standard for printer) the contents of the input device with file code /E1 (standard for cassette).

The following questions should be answered on the device with code /1 (standard for console typewriter).

- ASCII or EBCDIC : Specify the character representation on the tape by keying-in A for ASCII or E for EBCDIC

If the answer to the first question is A(ASCII), the following qeustion is output:

- ASCII or HEXA/ASCII : Specify the format of the dump by keying-in A for ASCII or H for HEXA/ASCII

If the answer to the first question is E(EBCDIC), the following question is output:

- EBCDIC or HEXA/EBCDIC : Specify the format of the dump by keying in E for EBCDIC or H for HEXA/EBCDIC

After this question/answer sequence DUMPA types out the prompt C: and starts printing. The user may then key-in one of the following commands:

- S Stop dump
- SX Skip X blocks and print
- S–X Skip to block X and print

An example tape dump follows:

```
                TAPE/CASSETTE DUMP
                DATE:          : :
                TIME:          00:05:35
                FORMAT:        ASCII

BLOCK NO:  1.1.1        SIZE:  32
1 S L 0 2 L O T I N P _ _ 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

END OF FILE

BLOCK NO:  2.1.1        SIZE:  240
1 2 2 0 0 4 0 1 8 0 2 1 _ _ _ _ _ _ _ 1 5 0 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
1 2 2 0 0 4 0 2 7 9 0 1 _ 3 0 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

BLOCK NO:  2.1.2        SIZE:  240
1 2 2 0 0 4 0 2 9 8 8 1 _ _ _ _ _ _ _ 3 0 0 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
1 2 2 0 0 4 0 3 0 8 7 1 _ _ _ _ _ _ _ 1 5 0 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
1 2 2 0 0 4 0 3 1 8 6 1 _ _ _ _ _ _ _ 1 5 0 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

BLOCK NO:  2.1.3        SIZE:  240
1 2 2 0 0 4 0 6 5 5 2 1 _ _ _ _ _ _ _ 1 5 0 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
1 2 2 0 0 4 0 6 6 5 1 1 _ 3 0 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
1 2 2 0 0 4 0 7 2 4 5 1 _ _ _ _ _ _ _ 1 5 0 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
```

The format of the BLOCK NO is TEB, where

T = the file number
E = = the segment number in the file
B = the block number in the file

The flexible disk sector numbering is standard logical, i.e.

Track 00, Sector 01 is standard logical sector 0
Track 76, Sector 26 is standard logical sector 2001

The following error messages may be output by the utility, and the table shows the causes and the actions taken by the utility in each case:

Message	Cause	Action
PARITY ERROR	Read error on input	Utility continues
UNKNOWN STATUS WORD = status word		Faulty record ignored, processing continues
NO DATA		
REWINDING		Utility continues

9.6 JESPER

JESPER is used to copy a program load module from disk onto a cassette in a form suitable for loading into memory during a TOSS system start.

If the load module is the TOSS Monitor, it must be preceded on the cassette by an IPL, and this is written using the CPLGEN utility (see 9.2).

The creation of the Monitor and/or Application cassette is normally carried out by using the catalogued procedure $PCAS, but JESPER may be executed separately if required.

The following steps must be taken to execute the JESPER utility:

Insert the cassette to be written in the left-hand cassette drive.

Key in the control command RUN JESPER xxxxxx, where xxxxxx is the name of the application load module to be copied to the cassette.

In the case of errors, the folowing messages may be output by the utility:

Message	Cause
ERROR CT <status-code>	Write failure on cassette
ERROR DK <status-code>	Read failure on disk
ERROR FM <status-code>	Format error in load module

9.7 OBX

OBX is used to obtain a cross-reference listing, on the Line printer, of the reference between a number of specified object modules on disk. It can be used for both CREDIT and ASSEMBLER modules.

The listing produced is in three sections, the module section, the symbol section and the run statistics section.

Module section – the first heading on this section is ENTRY REFERRED FROM MODULE(S) and under this heading appear, for each module, the symbols declared as ENTRY in the module, and for each ENTRY symbol, the names of the modules which refer to that symbol as EXTERNAL. If the symbol is not referred to by any other module, the word UNREFERENCED is shown by the symbol name. For each module that has no symbols declared as ENTRY, the words NO ENTRIES IN THIS MODULE are printed.

– the second heading in this section is EXTRN DEFINED IN MODULE and under this heading appear, for each module, the symbols that are declared as EXTERNAL in the module, and the name of the module where that symbols is referred to as ENTRY. If the symbol is not referred to as ENTRY in any other module, the word UNDEFINED is shown on the listing. If the symbol is referred to by more than one other module as ENTRY, the word MULTIDEFINED is printed, and below this the names of all the modules that refer to that symbol as ENTRY. For each module that has no EXTERNAL symbols, the words NO EXTERNAL REFERENCES appear.

Symbol section – the heading for this section is SYMBOL DEFINED REFERRED, and under this heading are printed, for all modules, all the symbols declared as ENTRY or EXTERNAL, and for each symbol the name of the module in which it is defined as ENTRY, and the names of all the modules that refer to it as EXTERNAL. If a symbol is undefined or multidefined, the listing shows *UD* or *MD* respectively, under the heading DEFINED. If multidefined, this is followed by the names of all the modules in which the symbol is declared as ENTRY.

Run Statistics section – this section shows the number of symbols and the number of modules in the listing, the available and unused number of list items, and the available and unused number of name items in the OBX work area in memory.

The following steps must be taken to execute the OBX utility:

Key in the control command /0

Key in the control command INC XXXXXX, where XXXXXX is the first object module to be included in the run.

Key in further INC commands, for the other modules to be included in the run.

Key in the control command ASG 3, DK

Key in the control command POB

Key in the control command RUN OBX

The following error messages may be output by the utility, in which case only the run statistics part of the report is produced.

DOUBLE DEFINED MODULE <module-name>
NAMES AREA OVERFLOW <name>
CHAIN ITEM AREA OVERFLOW <name>
INPUT I/O ERROR <name>

9.8 PM6800

This utility is used to format a disk for the PTS6875 or PTS6876 disk drive. The utility performs two disk passes, as follows:

On the first pass the disk is divided into sectors and a cylinder number is written in the first word of each sector. At the same time the volume label, granule allocation table (empty), IPL, userid catalogue (containing only SAG) and file directory for SAG (empty) are written to granules 0 and 1. A string of test characters is written to the remaining granules of the disk.

On the second pass the entire disk is re-read to check for faulty granules. The granule allocation table is updated so that any faulty granules will not be used for data. If either granule 0 or granule 1 is faulty an error report will be output. If granule 0 is faulty the disk is unuseable. If granule 1 is faulty the disk may not be used as a system disk.

The following steps must be taken to process the PM6800 utility:

Type in the command RUN PM6800

The following questions are then output by the utility on the console typewriter;

DISC FILE CODE =	type in a two digit code to specify on which drive the disk to be formatted is mounted, e.g. F0, F1, F2 or F3 (with no/preceding)
VOLUME LABEL nnnn OK?	Type in YES if the name represented by nnnn above is the volume label of the disk to be formatted, otherwise NO. If NO is entered, the question DISC FILE CODE is repeated.
DISC TYPE =	Type in 6875 or 6876
LABEL =	Type in a volume name of up to eight characters, with no embedded blanks, to be given to the disk
DATE =	Type in the date in the format DDMMYY
PACKNBR =	Type in a pack number up to three digits long
SYSTEM USERID =	Type in SAG

When the formatting of the disk is completed, the following question is output on the console typewriter:

RUN AGAIN =	Type in YES if another disk is to be formatted, otherwise NO. If NO is entered, the following message is output to the console typewriter; END OF PM6800, and processing is terminated.

9.9 PRDUMP

PRDUMP is used to print the memory dump from cassette or flexible disk resulting from the execution of the DMPGEN or DMPGEF utility. The utility reads data from the device assigned file code /E1, so this must be assigned with the ASG command prior to the execution of the utility.

Up to nine areas of memory may be selected for printing, or the entire dump cassette or disk may be printed. If areas are selected, the user must key in the hexadecimal limits of these areas on the console typewriter. The areas need not be supplied in any particular sequence.

The following steps must be taken to execute the PRDUMP utility:

Assign file code /E1 to the device containing the dump data set

Key in the control command RUN PRDUMP

The following questions are output on the console typewriter by the utility, and must be answered as follows:

SELECTION OF MEMORY DUMP AREAS?	Key in NO if the entire memory dump is required, otherwise YES followed by (CR)

If YES is answered to the previous question, the following is output:

MEMORY DUMP AREA 1	Key in the hexadecimal limits of the area required, as two sets of four digits separated by a hyphen, e.g. 4360-79BE, followed by (CR). The utility will then output a prompt for the next area definition. When all areas required have been entered, reply to the next prompt with a (CR) only.

After the dump has been printed, the utility will then output the following question:

MORE AREAS WANTED?	Key in NO if no further areas are required on the dump, or YES: if YES, the MEMORY DUMP AREA prompt will be repeated for the entry of further area definitions.

At the end of processing the utility outputs the message END OF PROMPT on the console typewriter.

An example of the output from PRDUMP is shown on the following page.

Example listing from PRDUMP

CASSETTE DUMP PRINTOUT PAGE: 1

P R D U M P REL 2.1

PRINTOUT OF MEMORY DUMP FROM CASSETTE.

DATE OF PRINTING:77:03:18
TIME :00:01:20

CASSETTE DUMP PRINTOUT PAGE: 2

0000	3153	4030	3240	4F54	494E	5020	2030	3130	"1SL02L0TINP 010"
0010	3030	3030	3030	3030	3030	3030	3030	3030	"0000000000000000"
0020	993B	2344	0086	0781	E898	1966	4821	DF33	".;#D.....H1.3"
0030	8864	3FDA	7F80	D0DA	79DC	4042	68D6	300E	".d?.....y.@Bh.0."
0040	3D3E	1810	832D	C1BD	7BAF	0094	3460	7F71	"=>...-..{...4`.q"
0050	00B2	7504	F89E	6DA6	505A	C934	8250	3080	".2u...m.PZ.4.P0."
0060	0278	C274	38DE	220C	4404	909E	F063	B8C4	".x.t8.".D....c.."
0070	25AF	0E86	909E	7106	B866	9447	9A40	8840	"%.....q..f.G.@.."

FINIS.
THERE WAS 1 BLOCK IN DUMP.
PROG ELAPSED TIME: 00H-00M-15S-040MR

In the case of an error, a message will be output by the utility, in the following format:
<TK/FL> ERROR STATUS WORD: <status-word>
In this case the utility is terminated abnormally, as a read error has occurred on the input device. The utility must then be restarted.

9.10 RUM

RUM is used to restore a complete disk or the disk library of a single userid from magnetic tape, on which the data has been saved by the utility SUM.

During execution, an optional listing of file names, etc. may be printed on the line printer. This listing includes, for each file, the file name, file type, number of occupied granules, and addresses of occupied granules, together with the letters A or R indicating whether a file has been added to the output disk or replaced on the disk.

If RUM is executed during a system session, the disk file code must be specified (see below). RUM will then restore each library from the tape until two consecutive tape marks are encountered. Any libraries that are present on the disk before execution are **deleted.**

If RUM is executed during a user session, files will be restored from the tape up to the first tape mark. Files which exist on the tape but not in the disk library are added to the library as new files. Files that exist on both input and output are replaced in the output library.

The user may execute RUM a number of times under selected userids, and thus restore a selection of libraries from the tape. Before running RUM the user must position the tape at the start of the appropriate library, if necessary by using the control commands FBS, FFS, REF and/or REW.

The following steps must be taken to process the RUM utility:

Position the tape if necessary as described above

Type in the command RUN RUM

The following question is then output to the console typewriter:

LISTING OF FILE NAMES ETC?	Type in Y if the listing is required, or N if it is not required.

If executing in a system session, the following question is then output to the console typewriter;

DISK FILE CODE?	Type in the disk number (e.g. F2) for the disk on which the files are to be restored. Note that the number must not be preceded by a slash.

The utility will then output details of the label, date and pack number of the disk specified, followed by the following:

OK?	Type in Y if this is the correct disk, or N if it is not correct. In the latter case, the utility outputs the disk file code question again, for the operator to specify a different disk.

9.11 SUM

SUM is used to save a complete disk or the disk library of a single userid on magnetic tape. During execution, an optional list of file names, etc. may be printed on the line printer. This listing includes, for each file, the file name, file type, number of occupied granules and addresses of occupied granules.

If SUM is executed during a system session a complete disk will be saved. The disk number (F0, F1, F2 or F3) must be specified in response to the question "DISK FILE CODE? " (a back slash must **not** precede the disk number). The disk will be saved one library at a time. Each library will be separated by a tape mark. An extra tape mark will be written after the last library. The magnetic tape will remain positioned after the last tape mark.

If SUM is executed during a user session only the library of the current userid will be saved. This will be followed by a tape mark and the magnetic tape will remain positioned after this mark. The user can execute SUM a number of times under selected userids and thus save a selection of libraries on the magnetic tape. An extra tape mark should be written (WEF) after the last library if the tape is to be restored during a system session(see below).

If SUM is executed under userid SAG only the library belonging to SAG will be saved. That is, SUM will **not** behave as if this was a system session.

The following steps must be taken to process the SUM utility:

Type in the command RUN SUM

The following question is then output by the utility on the console typewriter;

LIST OF FILE NAMES ETC?	Type in Y if the listing described above is required, or N if it is not.

9.12 SYSGEN

9.12.1 *Introduction*

It is the users responsibility to generate a version of the TOSS Monitor suited to the needs of his particular PTS6000 configuration. This process may be made simpler by using the utility SYSGEN.

The user must supply this utility with certain parameters describing the PTS6000 configuration. SYSGEN will then generate a file of control commands. If the user then assigns file code /E0 (standard for control command input) to this file the required version of the Monitor will be automatically generated.

The entire TOSS Monitor is stored on the system disk as a set of Assembler language source and object modules. The Monitor is not supplied to the user in load module form. The control commands generated by SYSGEN must, therefore, achieve the following:

Update selected source modules by inserting the parameters specified by the user.

Re-assemble any updated modules.

Selected the subset of TOSS Monitor object modules required for the specified configuration (including those which have been re-assembled).

Link edit the selected object modules to produce the Monitor load module.

The Monitor is generated as a load file on disk. It is the users responsibility to copy the Monitor onto a cassette or disk in a form suitable for loading directly into memory. This is done via the DOS6800 utility $PCAS which is described in section 10.6 or by the TOSS utility CPP (see TOSS Utilities Manual M08).

9.12.2 *Deferred binding*

The parameters to be supplied to SYSGEN are described in section 9.12.5 below. Occasionally it may be necessary to change certain parameters in an existing TOSS Monitor (for example a new type of device may be added to the hardware configuration). In such circumstances it is normally necessary to re-generate the TOSS Monitor and supply all the parameters again.

However, in the case of "task definition" and "common device definition" this is not so. The user can "defer the binding" of these parameters into the Monitor until TOSS System start.

During task definition the user supplies the task identification and priority of each task together with the device classes used by the tasks. During common device definition the user specifies the device classes included in the common device class. These parameters are described in sections 9.12.5.26 and 9.12.5.27.

If the user wishes to defer binding of these parameters he must answer "Y" to the SYSGEN question "MONITOR CONFIGURATION PROGRAM:" (described in section 9.12.5.4). If this is done the TOSS Monitor will execute the Monitor Configuration Program during the system start process. This program will read in the required parameters from a Monitor configuration cassette supplied by the user. The format and content of this cassette are described in the CREDIT PRM (M04) and in the Assembler PRM (M06).

The objective of deferred binding is to provide the user with a quick and simple method of supplying certain parameters which are likely to change frequently. However, if in a particular installation these parameters are not likely to change frequently it may be simpler for the user to answer "N" to the question "MONITOR CONFIGURATION PROGRAM:" and to supply the parameters directly to SYSGEN.

It should be noted that deferred binding is not available if a Monitor containing data management is being generated. That is if "Y" is answered to the question "MONITOR CONFIGURATION PROGRAM:" then "N" must be answered to the question "DATA MANAGEMENT:" (see 9.12.5.12).

9.12.3 *Running SYSGEN*

The following sequence of actions is required to run SYSGEN:

Ensure that the current userid has an empty file directory i.e. declare a new userid or delete all files in an existing userid.

Assign file code /CA to the device being used for the output control command file. This device may be disk (DK), magnetic tape (MT0C) or cassette (TK0E).

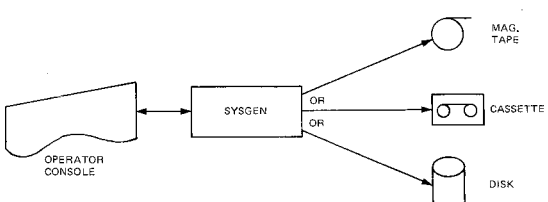
Call SYSGEN into execution in the normal way using the RUN control command.

Key-in parameters in response to questions typed out by SYSGEN. The permitted key-ins are described in section 9.12.5. At the end of this dialogue SYSGEN will close the output file and terminate.

If file code /CA is assigned to disk the file must be made permanent (control command KPF).

At the end of the above sequence the control command file is ready to be used to generate a TOSS Monitor.

The above procedure is summarised in the following diagram:



9.12.4 *Generating the TOSS Monitor*

The following sequence of actions is required to generate the TOSS Monitor:

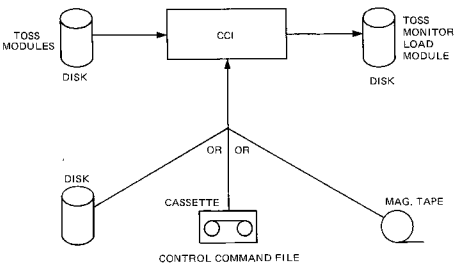
Ensure that the current userid has an empty file directory (apart from the control command file if this is on disk).

- Assign file code /E0 (standard for control command input) to the control command file produced by SYSGEN. This file may be on disk, magnetic tape or cassette.

The control commands in the input file will update and reassemble selected TOSS Monitor source modules, will select the object modules required for the specified configuration and will link edit the Monitor load module. The file containing the load module will be made permanent (control command KPF) and will be given the name specified by the user during the SYSGEN dialogue. Assembly listings will be produced if they have been requested during the SYSGEN dialogue. A map listing will also be produced by the Linkage Editor.

When all commands in the control command file have been obeyed control is handed back to the typewriter and the prompt S: is typed out.

The above process is summarised in the following diagram:



9.12.5 *SYSGEN dialogue*

9.12.5.1 *Special key-ins*

In addition to the special console typewriter keys mentioned in section 1.10 the following keys are recognised by SYSGEN, and should not be followed by (CR) or (LF) :

Normal PTS6862 Symbol	Hexadecimal Code	Meaning
^	/5E	Erase current answer: the next line is printed with an asterisk (*), and a new answer may be entered.
_ (Shift zero)	/5F	Erase previous character, i.e. backspace one character.

Normal PTS6862 Symbol	Hexadecimal Code	Meaning
#	/23	Abort. Terminate SYSGEN immediately.
$	/24	Erase current section of dialogue. This key-in may be used when entering conditional assembly parameters, LKM processor parameters, task definition parameters or common device parameters. The current section of dialogue is discarded and parameters may be input again. The sections of dialogue discarded are defined below.

Note that the above symbols may be different if a non standard keyboard is used on the PTS6862 console typewriter. However, the hexadecimal codes for those symbols must be the ones shown above. Note also that the hexadecimal code /A (normally (LF)) is ignored when given in reply to a SYSGEN question.

The following sections describe the key-ins permitted during the SYSGEN/ user dialogue. Each question/response in this dialogue has the following format:

SYSGEN-question: [user-response] (CR)

In certain circumstances the user may choose a default response by keying-in a (CR) immediately after the question. The values of these defaults are given in the following sections.

The syntax of the parameters used in user responses is defined in appendix A. The notation conventions are described in section 1.9.

9.12.5.2 *Assembly listing*

ASSEMBLY LIST: {Y / N}

Specify if an assembly listing of the Monitor modules is required. If N, no device is assigned to the print unit file code /02. Operations on this file will therefore have no effect.

9.12.5.3 *Monitor name*

MONITOR NAME: file-name

Specify the name which will be given to the new monitor load module, to a maximum of six characters. It is the file name which is specified in the keep file (KPF) command at the end of TOSS Monitor generation.

9.12.5.4 *Monitor Configuration Program*

MONITOR CONFIGURATION PROGRAM: {Y / N}

If "Y" is keyed-in, the questions in sections 9.12.5.26 and 9.12.5.27 will not be typed out by SYSGEN. The parameters in these sections will be supplied by the user on a monitor configuration cassette during TOSS system start. The format and content of this cassette are described in the CREDIT PRM (M04) and the Assembly PRM (M06).

If "Y" is keyed-in to the above question "Y" must also be keyed-in to the question "MONITOR AND APPL. ON SAME CAS:" (see section 9.12.5.7).

9.12.5.5 *Memory Management*

MEMORY MANAGEMENT: {Y / N}

Specify whether memory management (overlaying) is to be included in the Monitor. At present only assembler programs are able to use the memory management facilities. It is usual, therefore, to key-in "N" to the above question if the Monitor is to be used with a CREDIT application (see 9.12.5.11).

If "Y" is keyed-in two further questions will be asked by SYSGEN:

PARTITION LENGTH: hexadecimal-number
NUMBER OF PARTITIONS: decimal-number

where:

hexadecimal-number is four digits specifying the partition length in bytes.
decimal-number is three digits specifying the number of segments.

If memory management is to be used both the Monitor and application program must be held on a TOSS formatted disk (see CREDIT Reference Manual M04 and section 9.12.5.6 below).

9.12.5.6 *Program Loading from Disk*

PROGRAM LOADING FROM DISK: {Y / N}

Specify whether the Monitor and application program should be loaded from disk. If so the Monitor and application program must be held on a TOSS formatted disk (see CREDIT Reference Manual M04).

9.12.5.7 *Monitor and Application on Same Cassette*

MONITOR AND APPL. ON SAME CAS: {Y / N}

Specify whether the monitor and application programs will be output on the same or separate cassettes (i.e. by $PCAS). "Y" must be keyed-in if "Y" was answered to the question "MONITOR CONFIGURATION PROGRAM" (see section 9.12.5.4).

If "Y" is keyed-in SYSGEN will type out a further question:

CONFIGURATION DATA ON PROGRAM CASSETTE: {Y / N}

Specify whether the CREDIT configuration data will be held on the same cassette as the Monitor and application program.

9.12.5.8 *Monitor Initialization Program*

MONITOR INITIALIZATION PROGRAM: {Y / N}

This program development facility which allows the operator to restart the System at address /90 without having to reload the TOSS Monitor. It is not normally included in production systems.

5.12.5.9 *Assembler Debugger*

DEBUGGER: {Y / N}

Specify whether or not the Assembler Debugging Program is to be included. This is not usually included in production systems.

9.12.5.10 *Subroutine Call Interpreter*

SUBROUTINE CALL INTERPRETER: {Y / N}

This routine which reduces subroutine calls (CF A14, LABEL) to one word by using the illegal instructions /C001—/C0FF. The right byte of the instruction is an index (1, 2, 3 .....), used to fetch the subroutine address from an address table, pointed to by the contents of address /8E in the communication vector table.

Its inclusion is dependent upon the number of subroutines in the application.

9.12.5.11 *CREDIT Application*

CREDIT APPLICATION: {Y / N}

Specify whether the application is written in CREDIT. If YES, a CREDIT-adapted teller terminal printer driver, DRTP02, is included in the Monitor. All other drivers included are updated to accept CREDIT order codes only.

9.12.5.12 *Data Management Functions*

DATE MANAGEMENT: {Y / N}

Specify whether data management routines should be included.
Note: "N" must be keyed-in if "Y" was keyed-in to the question "MONITOR CONFIGURATION PROGRAM:" (see 9.12.5.4).

If data management is included disk units one and two or flexible disk drives one and two are set as a common device. File codes /F0 to /F3 are assigned to fixed/cartridge units 1 and 2 respectively, or file codes /F8 to /FB to left/right flexible units. The choice is dependent on the answers to the following section, Device Definition. One task table is generated for each disk unit with task identifiers D0 and D1. For flexible disk, a table with task identifier D2 is generated. These task identifiers must not be used by the application program.

9.12.5.13 *Device List:* {Y / N}

Specify if a list of device abbreviations is required, for use in the device specification (see below) (N).

9.12.5.14 *Define Devices*

DEFINE DEVICES: letter ‖ letter [,letter ‖ letter]....

Specify the types of device to be used by keying-in a selection of the following mnemonics, separated by a comma between each one.

KB keyboard
TP teller terminal printer
GP general printer
DN numeric display
DI indicator display/keyboard lamps
DY video/plasma display

LT	local terminals
RT	local and remote terminals
FD	flexible disk
MS	magnetic stripe unit
OD	optical document reader
II	intertask communication input
IO	intertask communication output
CT	customer operated teller
TC	tape cassette
SI	SOP input
SO	SOP output
MT	magnetic tape unit
TW	typewriter
DU	disk unit
LP	line printer
CR	card reader
CC	cassette changer
D1	data communication driver DRDC07
D2	data communication driver DRDC15
D3	data communication driver DRDC17
D4	data communication driver DRDC81
D5	data communication driver DRDC82
D6	users data communication driver 1
D7	users data communication driver 2

The following note explains the use of device types LT and RT.

Devices KB, TP, GP, DN, DI, DY and CT must be connected to the Terminal Computer via a Channel Unit for Local Terminals (CHLT) and/or a Channel Unit for Remote Terminals (CHRT). Devices which are used locally (i.e. not via modems) must be connected to the CHLT. Devices which are used remotely (i.e. via modems) must be connected to the CHRT.

One of two drivers may be used to control devices attached to the CHLT and CHRT. Driver DRLT01 is used to control devices attached to the CHLT. Driver DRRT01 is used to control devices attached either to the CHLT or CHRT. That is driver DRLT01 controls locally connected devices only, and driver DRRT01 controls both locally and remotely connected devices. Only one of these drivers may be included in the Monitor.

If DRLT01 is required then LT must be included in the device list. If DRRT01 is required then RT must be included in the device list. LT and RT cannot both be included in a device list.

If neither are specified, DRLT01 is automatically included by SYSGEN.

9.12.5 [illegible]

[illegible]

[illegible] TOSS Monitor is generated. The driver name must be in the format DRDCnn and the work table name in the format DWnn01.

LT local terminals
RT local and remote terminals
FD flexible disk
MS magnetic stripe unit
OD optical document reader
II intertask communication input
IO intertask communication output
CT customer operated teller
TC tape cassette
SI SOP input
SO SOP output
MT magnetic tape unit
TW typewriter
DU disk unit
LP line printer
CR card reader
CC cassette changer
D1 data communication driver DRDC07
D2 data communication driver DRDC15
D3 data communication driver DRDC17
D4 data communication driver DRDC81
D5 data communication driver DRDC82
D6 users data communication driver 1
D7 users data communication driver 2

The following note explains the use of device types LT and RT.

Devices KB, TP, GP, DN, DI, DY and CT must be connected to the Terminal Computer via a Channel Unit for Local Terminals (CHLT) and/or a Channel Unit for Remote Terminals (CHRT). Devices which are used locally (i.e. not via modems) must be connected to the CHLT. Devices which are used remotely (i.e. via modems) must be connected to the CHRT.

One of two drivers may be used to control devices attached to the CHLT and CHRT. Driver DRLT01 is used to control devices attached to the CHLT. Driver DRRT01 is used to control devices attached either to the CHLT or CHRT. That is driver DRLT01 controls locally connected devices only, and driver DRRT01 controls both locally and remotely connected devices. Only one of these drivers may be included in the Monitor.

If DRLT01 is required then LT must be included in the device list. If DRRT01 is required then RT must be included in the device list. LT and RT cannot both be included in a device list.

If neither are specified, DRLT01 is automatically included by SYSGEN.

9.12.5.15 *Data Communication Driver Name*

DC-DRIVER NAME, USERID: file-name, userid

SYSGEN asks the above question only if D6 or D7 has appeared in the device list described in 9.7.5.14.

File-name and userid identify a library file containing the driver, in source language form. The same library must also contain a data communication device work table (DWT) adapted to the specified driver. The specified library should be on-line when the TOSS Monitor is generated. The driver name must be in the format DRDCnn and the work table name in the format DWnn01.

9.12.5.16 *PTS6236 Keyboard in use*

KEYBOARD 6236 IN SYSTEM: {Y / N}

Specify whether or not a PTS6236 keyboard is in use in the system (N).

9.12.5.17 *Only PTS6236 (only if Y was answered to the preceding question).*

ONLY PTS6236: {Y / N}

Specify if only PTS6236 is in use within the system (N).

9.12.5.18 *PTS6805 Monitor required*

MONITOR FOR PHILIPS 6805: {Y / N}

Specify if the monitor is required for a PTS6805 Terminal Computer (N).

9.12.5.19 *Conditional Assembly Parameters for Drivers*

Conditional assembly parameters for each driver in the device list (9.7.5.13) must be specified. Each driver is listed in turn. If "S" is keyed-in as a response, standard values are assumed, except for data communication drivers, where the values **must** be supplied by the operator. If only a (CR) is entered, the operator must specify the parameters for that driver as requested. (For more detailed driver information see Assembler PRM M06).

Each driver is detailed below; the standard parameter values (which will be assumed if S is specified) are shown in parenthesis. In the case of data communication drivers, the values shown in parenthesis are not standard, but suggested values: values must be supplied by the operator.

If the operator makes an incorrect key-in, he may type $, whereupon all questions concerning current driver are repeated.

DRKB01 Keyboard (or DRKB03 if PTS6236 Keyboard)

- KEYBOARD TIME OUT: {Y / N}

  Specify if time-out function should be included (Y).

- ECHO FUNCTION: {Y / N}

  Specify if echo function should be included (Y).

- STANDARD READ: {Y / N}

  Specify if order /02, standard read, should be included (Y).

- NUMBER OF ZEROES FOR MULTIPLE ZERO KEY: decimal-digit
  Specify the number of zeroes for multiple zero key by typing one digit (2).

- COMPLETION OF READ REQUEST AT POWER ON: {Y / N}

  Specify if completion of read request is wanted at power on (N).

- CIRCULAR INPUT BUFFER SIZE: decimal-digit||decimal digit
  Specify the circular input buffer size, in characters, by typing two decimal digits (08).

- CODE CONVERSION/8-BIT SETTING: {Y / N}

  Specify if code conversion or 8 bit setting should be included (N).

DRTP01 (Assembler) Teller Terminal Printer

- WRITE TALLY ROLL: {Y / N}

  Specify if order /07, write tally roll, should be included (Y).

- CUT/PERFORATE JOURNAL TAPE: {Y / N}

  Specify if the functions cut/perforate journal tape should be included (N).

- SPECIAL CHARACTER/13: {Y / N}

  Specify if special character /13 should be included (N).

- SPECIAL CHARACTER/14: {Y / N}

  Specify if special character /14 should be included (N).

- COMPLETION OF REQUEST IF PRINT OBJECT IS REMOVED: {Y / N}

  Specify if order /08, print voucher/passbook, should be completed if print object is removed (N).

- INDICATION OF RECOVERY IN RETURN CODE (/80): {Y / N}

  Specify if indication of recovery is wanted in return code (/80) (N).

- ROOMLESS POINT CODES: { S / hexadecimal-number-1, hexadecimal-number-2, hexadecimal-number-3, hexadecimal-number-4, hexadecimal-number-5 }

If " S " is keyed-in standard values will be assumed for roomless point codes. These values are 2122, 2426, 3B3C, 3E40, 5E5F. Alternately a list of roomless point codes may be specified. In this case, hexadecimal-number-N comprises four hexadecimal digits specifying the ISO-7 codes for the number pairs 0 and 1, 2 and 3, 4 and 5, 6 and 7, 8 and 9. For example, if hexadecimal-number-1 has the value 2122 the ISO-7 code for 0. would be 21 and the ISO-7 code for 1. would be 22. The actual codes used depend upon the type of keyboard to be used.

- END OF REQUEST WHEN SELECTOR UNIT OR PRINTER INACTIVE: {Y / N}

Specify if I/O request should be ended if selector unit or printer is inactive (N).

DRTP02 (CREDIT) Teller Terminal Printer

- RETURN CODE UPON RECOVERY: {Y / N}

Specify if indication of recovery is wanted in return code (/80) (N).

- CUT/PERFORATE JOURNAL TAPE: {Y / N}

Specify if the functions cut/perforate journal tape should be included (N).

- SPECIAL CHARACTER /13: {Y / N}

Specify if special character /13 should be included (N).

- SPECIAL CHARACTER /14: {Y / N}

Specify if special character /14 should be included (N).

- COMPLETION OF REQUEST IF PRINT OBJECT IS REMOVED: {Y / N}

Specify if printing of voucher/passbook should be completed if print object is removed (N).

- END OF REQUEST WHEN SELECTOR UNIT OR PRINTER IS INACTIVE: {Y / N}

Specify if I/O request should be ended if selector unit or printer is inactive (N).

- ROOMLESS POINT CODES: { S / hexadecimal-number-1, hexadecimal-number-2, hexadecimal-number-3, hexadecimal-number-4, hexadecimal-number-5 }

Specify roomless point codes as described for DRTP01 above.

DRGP01 General Terminal Printer

- ROOMLESS POINT CODES: { N / S / hexadecimal-number-1, hexadecimal-number-2, hexadecimal-number-3, hexadecimal-number-4, hexadecimal-number-5 }

Specify roomless point codes as described for DRTP01 with the following exceptions:
"N" specifies that no roomless point is to be included.
Standard codes are 2122, 2324, 3B3C, 3E40, 5E5F.

- SPECIAL CHARACTER /13: {Y / N}

  Specify if special character /13 should be included (N).

- SPECIAL CHARACTER /14: {Y / N}

  Special if special character /14 should be included (N).

- ECHOING OF EXTRA SPACE BETWEEN CHARACTERS: {Y / N}

  Specify if echoing of an extra space is wanted between characters (N).

- ECHO OF EOR KEY: {Y / N}

  Specify if echo of EOR key is wanted (N).

- SUPPRESSION OF TRAILING BLANKS: {Y / N}

  Specify if trailing blanks should be suppressed (N).

- END OF REQUEST WHEN SELECTOR UNIT OR PRINTER INACTIVE: {Y / N}

  Specify if I/O request should be ended if selector or printer is inactive (N).

DRDN01 Numeric Display

- ORDER, WRITE NUMERIC DISPLAY: {Y / N}

  Specify if the order write numeric display, should be included (Y).

DRD101 Signal Display and Lamps on Keyboards

- ORDER, WRITE PROGRAM DISPLAY: {Y / N}

  Specify if the order write program display, should be included (N).

- FLASH FUNCTION: {Y / N}

  Specify if the flash function should be included (N).

DRDY01 Displays Videc and Plasma

- GRAPHIC MODE: {Y / N}

  Specify if graphic mode must be included in system (N).

- ONLY PTS6344: {Y / N}

  Specify if Video Display PTS6344 only is to be used (Y).

- PTS6351 INCLUDED: {Y / N}

  Specify if Plasma Display PTS6351 is to be used (N). This question is not asked if "Y" was keyed-in to the previous question.

- LOWER CASE CHARACTERS: {Y / N}

  Specify if lower case characters are to be used (N).

- ECHO OF EOR-KEY: {Y / N}

  Specify if echo of EOR key is wanted (N).

- SUPPRESSION OF TRAILING BLANKS: {Y / N}

  Specify if trailing blanks should be suppressed (Y).

- CURSOR STEADY ON 6386 (ELSE TWINKLING): {Y / N}

  Specify if cursor on PTS6386 is to be steady light (N).

DRLT01 Local Terminals

- LOGG FUNCTION FOR INPUT/OUTPUT CHRS: {Y / N}

  Specify if a logg function for input and output characters should be included (Y).

- NUMBER OF LOCAL CHANNEL UNITS: decimal-digit

  Specify the number of local channel units by keying-in one digit (1)

- ACCUMULATORS FOR NAK, RETRANSMISSION FAULT AND UNDEFINED CONTROL CHRS: {Y / N}

  Specify if accumulators for NAK, retransmission fault and undefined control characters should be included (Y).

- SOFTWARE TIME OUT HANDLING MISSING DATA REQUESTS FROM PRINTERS: {Y / N}

  Specify if a software time-out, handling missing data requests from printers, should be included (Y).

- ECHO FUNCTION: {Y / N}

  Specify if the echo function should be included (Y).

DRRT01 Local and Remote Terminals

- LOGG FUNCTION FOR INPUT/OUTPUT CHRS: {Y / N}

  Specify if a logg-function for input and output characters should be included (Y).

- NUMBER OF LOCAL CHANNEL UNITS: decimal-digit

  Specify the number of local channel units (1).

- ACCUMULATORS FOR NAK, RETRANSMISSION FAULT AND UNDEFINED CONTROL CHRS: {Y / N}

  Specify if accumulators for NAK, retransmission fault and undefined control characters should be included (Y).

- NUMBER OF REMOTE CHANNEL UNITS: decimal-digit

  Specify the number of remote channel units (2).

- REMOTE TEST FILE CODE: decimal-digital decimal-digit
  Specify remote test file code by keying-in two digits (15).

- ECHO FUNCTION: {Y / N}

  Specify if the echo function should be included (N).

DROD01 Optical Document Reader

- COMPLETION OF READ REQUEST AT POWER ON: {Y / N}

  Specify if completion of read request is wanted at power on (N)
- CIRCULAR INPUT BUFFER SIZE: decimal-digit decimal-digit
  Specify the circular input buffer size, in characters, by keying in two decimal digits (80)

DRTC01 Digital Cassette Recorder

- NUMBER OF CASSETTE DRIVES: decimal-digit
  Specify the number of cassette drives on the Terminal Computer (1).

- ORDERS LOCK, ERASE AND REVERSE ONE BLOCK: {Y / N}

  Specify if these functions must be included (N).

DRSOP1 System Operators Panel

- EXTRA SOP INPUT DWT: {Y / N}

  Specify if the facility to handle two simultaneous read requests, each with its own file code should be included in the driver (N). This is used to stop CREDIT Debugging Program dumps.

- FLASH FUNCTION: {Y / N}

  Specify if the flash function should be included (N).

DRTW01 Console Typewriter

- KEYBOARD TIME OUT: {Y / N}

  Specify if a keyboard time-out function should be included (N).
- COMPLETION OF READ REQUEST AT POWER ON: {Y / N}

  Specify if completion of read request is wanted at power on (N).

DRDC07 Data Communication Driver

- RECEIVE BUFFER LENGTH: decimal-digit‖decimal-digit.‖decimal-digit

  Specify the number of words in the receive buffer by keying in three decimal digits (130).

- TRANSMIT BUFFER LENGTH: decimal-digit‖decimal-digit‖decimal-digit.

  Specify the number of words in the transmit buffer by keying in three decimal digits (0).

- DC–TASK DATCOM FILE CODE: file-code

  Specify the file code to be used for data communication. The file code must not be preceded by a slash (60).

- INTERRUPT LOGGING: {Y / N}

  Specify if interrupt logging is required to be included (Y).

- POLL TIMEOUT VALUE: decimal-digit‖decimal-digit‖decimal-digit.

  Specify poll timeout value by keying in three decimal-digits (300).

- NUMBER OF RECEIVE BUFFERS (3–14): decimal-digit‖decimal-digit.

  Specify the number of receive buffers required by keying in two decimal-digits.

- FULL DUPLEX: {Y / N}

  Specify if full duplex mode is to be included (N).

- PHYSICAL LINE NUMBER: decimal-digit‖decimal-digit.

  Specify physical line number to be used by keying in two decimal-digits.

DRDC15 Data Communication Driver

- RECEIVE BUFFER LENGTH: decimal-digit‖decimal-digit‖decimal-digit

  Specify the number of words in the receive buffer by keying-in three decimal-digits.

- TRANSMIT BUFFER LENGTH: decimal-digit‖decimal-digit‖decimal-digit.

  Specify the number of words in the transmit buffer by keying-in three decimal-digits.

- DC–TASK DATCOM FILE CODE: file-code.

  Specify the file-code to be used for data communication. The file-code must not be preceded by a slash (60).

- INTERRUPT LOGGING: {Y / N}

  Specify if interrupt logging is required to be included (Y).

- POLL TIMEOUT VALUE: decimal-digit‖decimal-digit‖decimal-digit

  Specify poll timeout value by keying-in three decimal-digits (600).

- STATUS AND RVI HANDLING: {Y / N}

  Specify if status and RVI handling should be included (Y).

DRDU01 Disk Drive

- READ AFTER WRITE: {Y / N}

  Specify if the Read after Write check is to be included (N).
- NUMBER OF DISK UNITS: decimal digit

  Specify the number of disk drives connected (1).
- UNIT 1 PTS6876: {Y / N}

  Specify if the first disk unit is a PTS6876 type (N).

  If more than one unit has been specified, the above question is repeated for each unit connected.

DRLP01 Line Printer

- LP ON PROGRAMMED CHANNEL (ELSE MUX): {Y / N}

  Specify if the line printer is connected to the programmed channel (N).

DRFD01 Flexible Disk Drive

- FD ON PROGRAMMED CHANNEL (ELSE MUX): {Y / N}

  Specify if the flexible disk units are connected to the programmed channel or the multiplexer (Y).
- FD ADAPTED FOR DATA MANAGEMENT: {Y / N}

  Specify if the flexible disk units are adapted for data management (N).
- IBM LABELLED DISK HANDLING: {Y / N}

  Specify if IBM labelled flexible disks are to be handled by the system (N).
- FD ADAPTED FOR INIMON: {Y / N}

  Specify if the flexible disk driver is to be adapted to include the monitor initialisation module, see 9.12.5.8 (N).

DRCR01 Card Reader

- CR ON PROGRAMMED CHANNEL (ELSE MUX): {Y / N}

  Specify if the card reader is connected to the programmed channel or the multiplexer (Y).

- READ COMMAND HANDLING: {Y / N}

  Specify if read command handling should be included (Y).
- TRANSMIT BLOCK LENGTH: decimal-digit‖decimal-digit‖decimal-digit
  Specify transmit block length by keying-in three decimal-digits (254).
- EBCDIC CODE (IF NO ASCII): {Y / N}

  Specify if EBCDIC code is to be used (Y).
- SPECIFIC POLL HANDLING: {Y / N}

  Specify if specific poll handling should be included (N).
- LINE SPEED HIGH: {Y / N}

  Specify if line speed is high (Y).
- SIEMENS MSV1 PROCEDURE: {Y / N}

  Specify if Siemens MSV1 procedure should be included (N).
- NUMBER OF RECEIVE BUFFERS: decimal-digit
  Specify the number of receive buffers by keying-in one decimal-digit (2)
- PHYSICAL LINE NUMBER: decimal-digit‖decimal-digit
  Specify the physical line number to be used by keying-in two decimal-digits.

DCDC17 Data Communication Driver

- NUMBER OF LINES: decimal-digit.
  Specify the number of lines by keying in one decimal-digit (1) .
- BSC PROCEDURE (ELSE SIEMENS MSV2): {Y / N}

  Specify if BSC procedure should be included (Y).
- EBCDIC CODE (IF NO ASCII): {Y / N}

  Specify if EBCDIC code is to be used (Y).
- EBCDIC TRANSPARENCY: {Y / N}

  Specify if EBCDIC transparency should be included (N).

- BID TIMEOUT VALUE: decimal-digit‖decimal-digit
  Specify the BID timeout value by keying-in two decimal-digits (10).
- LOGG AREA LENGTH PER LINE: decimal-digit‖decimal-digit‖decimal-digit.
  Specify the logg area length per line by keying-in three decimal-digits (500).
- PHYSICAL LINE NUMBER: decimal-digit‖decimal-digit.
  Specify physical line number to be used by keying-in two decimal-digits.

DRDC81 Data Communication Driver

- NUMBER OF LINES: decimal-digit‖decimal-digit.
  Specify the number of lines by keying-in the two decimal-digits.
- POLL LIST LENGTH: decimal-digit‖decimal-digit
  Specify the length of the poll list by keying-in two decimal-digits.
- P852 INSTRUCTION SET: {Y / N}

  Specify if the P852 Instruction set is to be included.
- MAX BLOCK LENGTH: decimal-digit‖decimal-digit‖decimal-digit.
  Specify the length of the maximum block by keying in three decimal-digits.
- INTERRUPT LOGGING: {Y / N}

  Specify if interrupt logging should be included (Y).
- RETRY LIMIT: decimal-digit‖decimal-digit
  Specify the retry limit by keying-in two decimal-digits.
- PHYSICAL LINE NUMBER: decimal-digit‖decimal-digit.
  Specify the physical line number to be used by keying-in two decimal-digits.

DRDC82 Data Communication Driver

- NUMBER OF LINES: decimal-digit‖decimal-digit.
  Specify the number of lines by keying-in two decimal-digits.
- POLL LIST LENGTH: decimal-digit‖decimal-digit.
  Specify the length of the poll list by keying in two decimal-digits.
- P852 INSTRUCTION SET: {Y / N}

  Specify if the P852 instruction set is to be included.
- MAX BLOCK LENGTH: decimal-digit‖decimal-digit‖decimal-digit
  Specify the length of the maximum block by keying-in three decimal-digits.
- INTERRUPT LOGGING: {Y / N}

  Specify if interrupt logging should be included (Y).

- RETRY LIMIT: decimal-digit‖decimal-digit

  Specify the retry limit by keying-in two decimal digits.
- PHYSICAL LINE NUMBER: decimal-digit‖decimal-digit.

  Specify the physical line number to be used by keying-in two decimal-digits.

DRDCXX Data Communication Driver

- TRANSMIT BUFFER LENGTH: decimal-digit‖decimal-digit‖decimal-digit

  Specify the number of words in the transmit buffer by keying-in three decimal-digits.
- RECEIVE BUFFER LENGTH: decimal-digit‖decimal-digit‖decimal-digit.

  Specify the number of words in the receive buffer by keying-in three decimal-digits.
- DC-TASK DATCOM FILE CODE: file-code

  Specify the file code to be used for data communication. The file code must not be preceded by a slash.
- PHYSICAL LINE NUMBER: decimal-digit‖decimal-digit .

  Specify the physical line number to be used by keying-in two decimal-digits.

DRDCYY Data Communication Driver

- TRANSMIT BUFFER LENGTH: decimal-digit‖decimal-digit‖decimal-digit

  Specify the number of words in the transmit buffer by keying-in three decimal-digits.
- RECEIVE BUFFER LENGTH: decimal-digit‖decimal-digit‖decimal-digit

  Specify the number of words in the receive buffer by keying-in three decimal-digits.
- DC–TASK DATCOM FILE-CODE: file-code

  Specify the file code to be used for data communication. The file code must not be preceded by a slash.
- PHYSICAL LINE NUMBER: decimal-digit‖decimal-digit

  Specify the physical line number to be used by keying-in two decimal-digits.

DRCG01 Customer Operated Teller

The questions in this section are not typed-out by SYSGEN if "Y" was answered to the question "CREDIT APPLICATION" (see section 9.12.5.11).

- PIN CALCULATION FIELDS: S1E1, S2E2, S3E3, S4E4.

  Specify the personal identification number calculation fields by keying-in the start positions of the Nth field (SN) and the end positions of the Nth field (EN).

  A COT transaction card contains 104 positions. A PIN calculation field is a contiguous subset of these positions. If the start position is given a higher value than the end position, for a PIN calculation field, this field is omitted.

  The standard values are: 3035, 0310, 0100, 0100.
- SWALLOW CARD TIMEOUT (MULTIPLES OF 2.56 MS): decimal-digit .....

  Specify the time after which the transaction card will be swallowed (2).
- TRANSMISSION TIMEOUT (MULTIPLES OF 2.56 MS): decimal-digit .....

  Specify the transmission timeout (32).

- PIN LOADING TIMEOUT (MULTIPLES OF 2.56 MS): decimal-digit .....
  Specify the PIN loading timeout (20).
- BUZZER TIMEOUT (MULTIPLES OF 10 MS): decimal-digit .....
  Specify the buzzer timeout (10).
- NOTE LENGTH TIMEOUT (MULTIPLES OF 10 MS): decimal-digit .....
  Specify the note length timeout (16).
- STANDARD KEYBOARD LAYOUT: {Y / N}

  Specify if the standard keyboard layout will be used (Y).

Standard layout:

7	8	9
4	5	6
1	2	3
F0	0	F1

Non standard layout

1	2	3
4	5	6
7	8	9
F0	0	F1

9.12.5.20 *LKM Processors*

Various LKM processors may be included in the system: optional LKM processors are as follows:

- DELAY/DELAY AND ACTIVATION: {Y / N}

  Specify if the timing functions above are required.
- DYNAMIC BUFFER ALLOCATION: {Y / N}

  Specify if dynamic buffer allocation is required.
- ABORT FUNCTION: {Y / N}

  Specify if the abort function is required.
- GET/SET TIME: {Y / N}

  Specify if the get/set time requests are required.
- ATTACH/DETACH: {Y / N}

  Specify if the attach/detach function is required.

If the operator wishes to change the answers after replying, he may key-in **$**, and the above questions are then repeated.

There are three types of device class:

- Terminal device class
- Echo device class
- Special device class

One or more of each of these device classes may be defined.

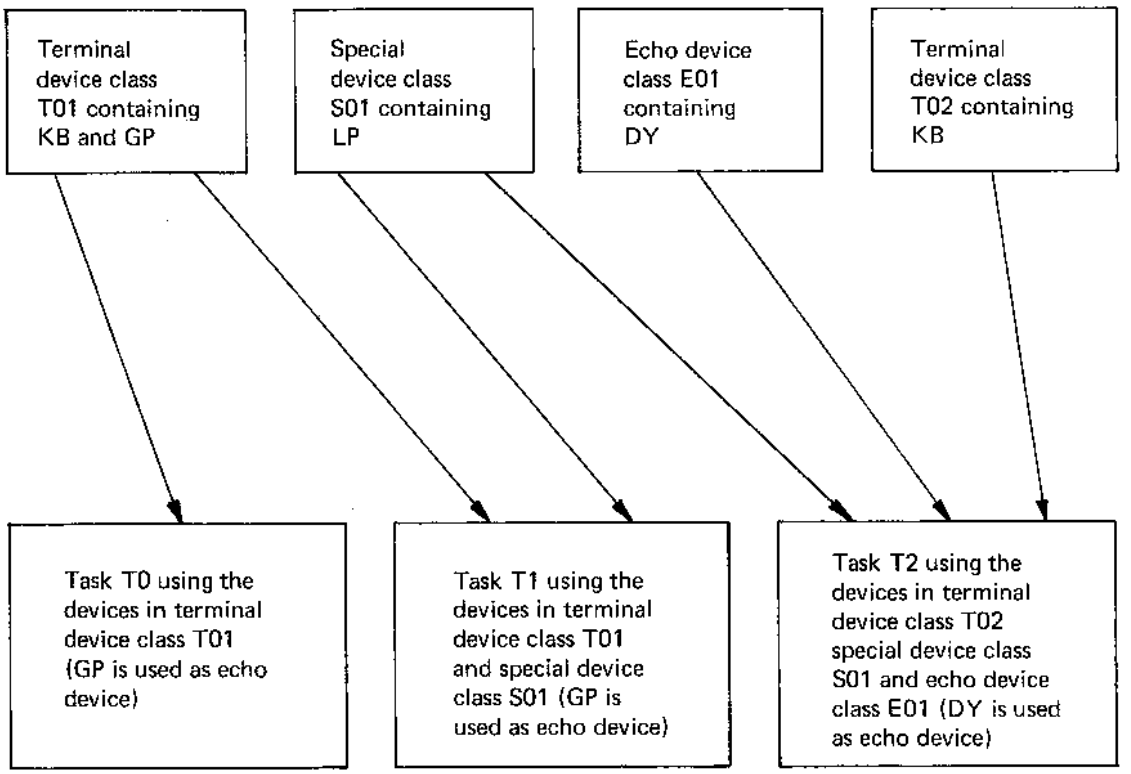
A terminal device class must contain one or more devices that are connected to a single CHLT or CHRT line (i.e. device types KB, TP, GP, DN, DI, DY and/or CT).

An echo device class must contain **one** device that is connected to a CHLT or CHRT line and is to be used as an echo device to a keyboard (i.e. device types GP or DY). Echo devices may be specified in a terminal device class. However, if the echo device is not connected to the same CHLT/CHRT line as the keyboard, it should be defined in a separate echo device class.

A special device class must contain one or more devices that are **not** connected to a CHLT or CHRT (i.e. device types SO, SI, TC, DU, LP, TW, MT and/or CC).

The present section describes the definition of terminal device classes. Sections 9.7.5.23 and 9.7.5.24 respectively describe the definition of echo device classes and special device classes.

The following diagram illustrates the use of device classes.



9.12.5.21 *Data Management*

If Data Management has been requested (see 9.12.5.12), the following questions are asked:

- MAX NUMBER OF DISK FILES: decimal-digit‖decimal-digit

  Key-in two digits specifying the maximum number of data files assigned at the same time.

- MAX NUMBER OF COMMON DISK-FILE FILE CODES: decimal-digit‖decimal-digit.

  Key-in two digits specifying the maximum number of file codes used for data files.

- MAX NUMBER OF DISK BUFFERS: decimal-digit‖decimal-digit.

  Key-in two digits specifying the number of sector buffers (2).

- EXCLUSIVE ACCESS HANDLING: {Y / N}

  The exclusive access check is included if Y is keyed-in.

- INDEXED ACCESS: {Y / N}

  Specify if indexed access is required. If Y is answered to this question, the following two questions are asked, in addition to those following, which are always asked.

- MAX KEY LENGTH IN BYTES: decimal-digit‖decimal-digit .....

  Key in up to three digits to specify the maximum key length.

- NUMBER OF WORDS IN MASTER INDEX AREA: decimal-digit‖decimal-digit‖decimal-digit‖decimal-digit

  Key in the number of words in the master index area in four digits.

- MAX NUMBER OF CURRENCY BUFFERS: decimal-digit‖decimal-digit

  Specify the number of currency buffers by keying-in two digits.

- MAX NUMBER OF RECORDS WITH EXCLUSIVE ACCESS: decimal-digit

  Specify the number of records with exclusive access (maximum) by keying-in one decimal-digit.

- LOCAL DISC-FILE FILE CODES: {Y / N}

  Specify if local disk file codes are required for this task.

- MAX NUMBER OF LOCAL DISC-FILE FILE CODES: decimal-digit

  Specify the maximum number of local file codes by keying in one decimal-digit.

9.12.5.22 *Terminal Device Classes*

For each device used by a task certain parameters (e.g. file code, device address) must be keyed-in. Several different tasks may use the same types of device. To avoid keying-in the same parameters for several tasks during task definition (9.7.5.22) these parameters may be keyed-in once in a "device class". This device class may then be included in the "task configuration list" during task definition.

- T01: {Y / N}

  A "T" followed by terminal device class number is output. By answering Y/N the operator declares if he wants to define the terminal device class or not. If N, terminal device class definition is ended.

- DEVICE–ABBREVIATIONS: {Y / N}

  For each CHLT/CHRT device-type chosen (see 9.12.5.14), the two character device abbreviations is output. The device-type is included in the terminal device class if the operator replies Y, else not (N).

  If the device is included in the terminal device class, the operator has to specify some of the device parameters described below.

- FILE CODE: file-code [,file-code].....

  Note there is no slash (/) before the file code.
  Normally only one file code is specified for a device.

  If n file codes are specified, the first one is assigned to the device. n-1 **extra terminal device classes** is then generated, containing this device only, but with the file code replaced with the 2nd, 3rd, ... , nth file code in the file code list. It is possible to declare several file codes for only one of the devices in a terminal device class.

  If the device is a "CREDIT printer", each print station on the printer should have its own file code. The 1st file code, in the file code list, is then assigned to journal tape, the 2nd to voucher/passbook and the 3rd to tally roll.

  Since one terminal device class is generated for each file code in the file code list, a "CREDIT printer", with the three print stations, is built up by three terminal device classes. At task definition time (see below) these three terminal device classes must be defined in the task configuration list.

- DEVICE ADDRESS: hexadecimal-digit [,hexadecimal-digit]

  Hexadecimal-digit is the device address. This must not be zero. Normally one device address is assigned to a device.

  It is however possible to define two device addresses together with the same file code for keyboards. Thus the different keyboards are assigned to the same device work table (DWT).

  Recommended device addresses:

  KB : 1
  GP, TP : 2
  DI : 3
  DN, DY : 4

- PTS NUMBER: decimal-number

  Key-in the four digit PTS product number. This question is only asked if there is more than one PTS product available within the current device type. For example, if the device type is MT the model of magnetic tape unit must be specified by keying in 6164, 6168 or 6872. A list of these device types and product numbers is given below:

Keyboard	(KB)	:	PTS6231/6232/6233/6234/6236/6331
Teller terminal printer	(TP)	:	PTS6221/6222/6224
Signal display	(DI)	:	PTS6232/6233/6234/6241/6242/6331
Video/plasma display	(DY)	:	PTS6342/6344/6351
Magnetic tape	(MT)	:	PTS6164/6168/6872

Note: The product numbers for magnetic tape may only be used in special device classes.

- PIN CHARACTER: hexadecimal-digit‖hexadecimal-digit

This question is only output for driver DRCG01 (Customer Operated Teller). Specify the hexadecimal code for the ISO-7 character that should be output on the COT display when entering personal identification number (PIN).

The following questions are only asked for displays:

- NUMBER OF LINES ON DISPLAY: decimal-digit‖decimal-digit
- NUMBER OF CHARACTERS/LINE: decimal-digit‖decimal-digit

One of the following should be keyed-in:

Video Display Unit: 20 lines of 64 characters
Plasma Display Unit: 08 lines of 36 characters

The following questions are asked for keyboards only:

- BACKSPACE KEY: hexadecimal-digit‖hexadecimal-digit
- CANCEL KEY: hexadecimal-digit‖hexadecimal-digit
- TRIPLE ZERO KEY: hexadecimal-digit‖hexadecimal-digit
- MULTIPLE ZERO KEY: hexadecimal-digit‖hexadecimal-digit

Specify the hexadecimal codes for the ISO-7 characters to be used for the above functions. If a function is not required FF should be keyed-in.

- ECHO DEVICE: GP|DN|DY|N|E

Specify the device on which the input characters will be echoed. GP, DN and DY are explained in section 9.12.5.14. N means that no echo device will be used. E means that the echo device is defined later in the echo device class.

If the operator answers GP, DN or DY the echo device must be defined in the same terminal device class as the keyboard and hence connected to the same CHLT/CHRT line.

Type E if the echo device is connected to another line on the CHLT/CHRT than the keyboard. If so, the terminal device class must be followed by an echo device class (containing echo device) in the task configuration list (see below).

If the operator detects that a terminal device class (as well as an echo device class or special device class) has been incorrectly defined, he defines a new class. The incorrect class is thereafter not used by the operator at task/common device definition time.

- CODE CONVERSION: {Y / N}

Specify if code conversion is required.

If code conversion is included, ISO-7 code columns that should be changed, must be specified in the following way:

COLX (CR) [COLY (CR) ..... COLZ (CR)] (CR)

where:

COLX: : = X : CODES
CODES: : = CODE1 || CODE2, ..., CODE15|| CODE16

X	digit between 0—7 specifying ISO-7 code column number.
CODE1 CODE2	four hexadecimal digits specifying the first and second code word in the ISO-7 code column X.
(CR)	carriage return.

Note that the above question is not output if only PTS6236 keyboards are included in the system: in this case, it is the responsibility of the programmer to supply a code conversion table, if required (see Assembler Programmers Manual M06 Part 2).

- 8-BIT SETTING: {Y / N}

Specify if eight bit setting is to be included.
If 8-bit setting is included, bit 0 is set to 1 for all characters, except digits. If two device addresses are specified, 8-bit setting is implemented for the first keyboard (first device address). 8-bit setting is automatically included for keyboard PTS6234.

- KEYBOARD TIMEOUT: {Y / N}

Specify if a 30 sec. timeout should be used for read requests on keyboards.

9.12.5.23 *Echo Device Classes*

An echo device class contains one CHLT/CHRT device that could be used as echo device to a keyboard. The echo device should be defined in an echo device class if it is connected to another CHLT/CHRT line than the keyboard (see 9.12.5.22). Note that echo device classes are not supported by the Monitor Configuration Program.

Echo device classes are declared in the same way as terminal device classes.

9.12.5.24 *Special Device Classes*

A special device class is a group of devices other than CHLT/CHRT devices (see 9.12.5.22).

For CREDIT applications both tape cassette units must be included (used by the Configurator). Each unit must be included in a different special device class.

If MONCON is to be used, a special device class must be specified to read the cassette containing the Monitor configuration data.

For each device included, some of the parameters below should be defined.

Devices are included in a special device class in the same way as for terminal device classes.

- FILE CODE: file-code[,file-code]. . . .

  Note that there is no slash (/) before the file code.

  It is only allowed to specify one file code, in the file code list, for disk unit and typewriter. For these devices two questions are output, to allow the operator to specify the file codes for cartridge disk, fixed disk and keyboard printer respectively.

If several file codes are defined, in the file code list, the meaning is the same as described in 9.12.5.22.

As an option the System Operators Panel may handle two simultaneous read requests, each with its own file code. If this facility is included in the driver the first file code in the above file code list refers to the first device work table; the second file code refers to the second device work table. This facility is used to stop CREDIT debugging program dumps. The second file code should be 14.

If the application program is written in CREDIT and the trace facility of the CREDIT Debugging Program is to be used, file code 16 should be devoted to the trace device.

- UNIT NUMBER: decimal-digit

Unit number is only used for cassette tape drives and magnetic tape units.

Cassette tape:

1 = left cassette drive
2 = right cassette drive

Magnetic tape:

1 1 st mag. tape unit
2 2 nd mag. tape unit
. .
. .
. .
. .
8 8 th mag. tape unit

9.12.5.25 *Extra Monitor Modules*

- EXTRA MONITOR MODULES: {Y / N}

Specify if extra monitor modules should be included in the Monitor.

If Y, extra modules are declared in a module list. SYSGEN types out a series of asterisks, each on a new line. The format of each module declaration is as follows:

* file-name, userid

The above response may be repeated on successive lines following the * prompt typed out by SYSGEN. The last * in the list should be followed by a response of (CR) only.

It should be noted that if a monitor module should be replaced by another module, the name of the replaced and replacing modules should be the same.

9.12.5.26 *Task Definition*

The questions in this section are not typed out by SYSGEN if "Y" was answered to the question "MONITOR CONFIGURATION PROGRAM:" (see 9.12.5.4).

TASK 01: {Y / N}

By replying Y or N to the question TASK XX : (XX = task number), the operator declares if he wants to define task XX.
If N, task definition is ended.

For each task, the operator has to specify task identification and task configuration list.

- TASK IDENTIFICATION: alphanumeric character‖alphanumeric character [,alphanumeric-character‖alphanumeric-character] . . . .

The alphanumeric character pairs are task identifiers.
The task identifier is considered as a decimal number and converted to binary, if the characters input are decimal digits, else as a two character ISO-7 string.

If n task-identifiers are specified n-1 extra tasks are generated, with the 2 nd, 3 rd, ... ,n th task identifiers respectively. All n tasks will get the configuration defined in the configuration list below, but with the CHLT/CHRT devices connected to different lines.

If a terminal class is connected to CHLT/CHRT line x, in the configuration list below, this class will be connected to CHLT/CHRT line, x, x + 1, . . . , x + n for the 1 st, 2 nd . . . . , n th task defined in the task identification list respectively.

- TASK LEVEL: 51|52|53|54|55|56|57|58|59|60|61|62

Priority level is defined by typing one of the decimal level numbers between 51–62. Normal level is 60.

If the CREDIT Debugging Program is to be used a special task must be specified. The task identification is TB and it runs on priority level 55. It uses a task with a special device class in which the typewriter device is included.

The input file code is then	:	21
output file code	:	31
backspace key	:	5F
cancel key	:	5E
end of record key	:	0D

- TASK CONFIGURATION LIST: $\left\{\begin{matrix}T\\E\\S\end{matrix}\right\}$ CLANBR [LINBR $\left\{\begin{matrix}L\\R\end{matrix}\right\}$]

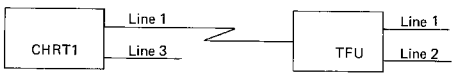
The above response may be repeated on successive lines following the * prompt typed out by SYSGEN. The last item in the list should be followed by a response of (CR) only.

A task configuration is built up by connecting a number of terminal device classes, echo device classes and special device classes to the task. This is done by defining these classes in the task configuration list.

T	terminal device class
E	echo device class
S	special device class
CLANBR	decimal class number, one/two decimal-digits
LINBR	decimal line number on CHLT/CHRT, one/two decimal-digits
	1 – 8 for CHLT1
	9 – 16 for CHLT2
	1 – 4 for CHRT1
	5 – 8 for CHRT2
L	devices connected to local channel unit
R	devices connected to remote channel unit

If the operator answers just (CR) , no devices are attached to the task.

Note: CHRT1 may have a Transfer Unit connected to each line, giving effectively four lines (see diagram below):



If an odd line number is specified, the line operates in block mode. If an even number is specified, the pair of lines operate in character mode.
Example: Specification of lines 2 and 3 gives lines 1 and 2 operating in character mode and line 3 operating in block mode.
If no Transfer Unit is connected, then the two lines available are considered to be numbered 1 and 3.

The first task defined, is always started by the Monitor after program loading.

The task configuration list for the first task in a CREDIT program must contain a special device class in which the System Operators Panel input (and optionally output) is defined. It should be noted that it it has been declared in 9.12.5.22) that the echo device is later defined in an echo device class, the concerned terminal device class must be immediately followed by that echo device class, in the task configuration list.

If the operator detects an error in task definition, he may type $, whereupon task definition is restarted.

Example:

A task containing

KB	on CHLT-line 1
DN	numeric display as echo device on CHLT-line 2
TP	on CHLT-line 1
MT	one magnetic tape unit

This task may be defined in the following way.

T1	declare terminal device class T1 containing KB and TP. Echo device is defined by answering "E" (see 9.12.5.22)
E1	declare echo device class E1 containing DN
S1	declare special device class S1 containing MT

Task configuration list should then be specified in the following way:

T1, 1L (CR) terminal device class 1 on CHLT line 1
E1, 2L (CR) echo device class 1 on CHLT line 2
S1 (CR) special device class 1
(CR)
(Line feed is made by SYSGEN).

9.12.5.27 *Common Device Definition*

The questions in this section are not typed out by SYSGEN if "Y" was answered to the question "MONITOR CONFIGURATION PROGRAM;" (see section 9.12.5.4).

- COMMON DEVICES: {Y / N}

By replying Y or N the operator declares if he wants to define common devices, i.e. devices that are common to all tasks in the System.

Common device configuration is built up by a number of terminal device classes, echo device classes and special device classes defined in 9.12.5.22 to 9.12.5.24.

These classes are defined in a configuration list, with the same syntax as described in 9.12.5.26.

If the application is written in CREDIT and the trace facility of the CREDIT Debugging Program is to be used, the special device class containing the trace device should be included.

If the $ key is depressed, common device definition is restarted.

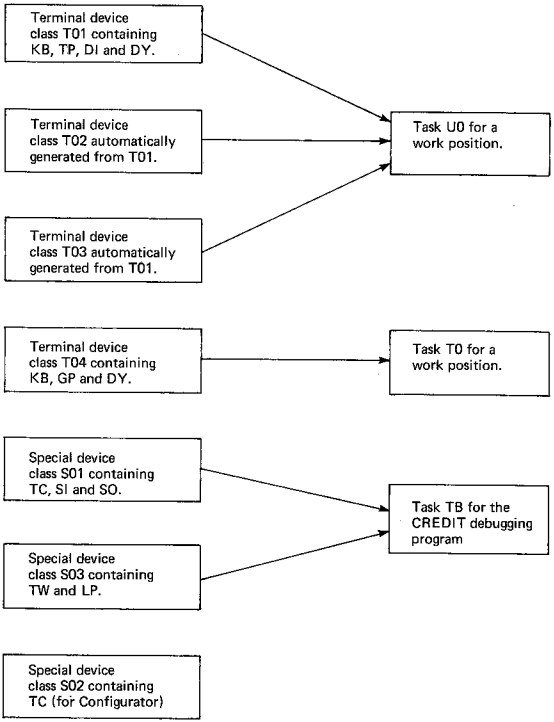
9.12.5.28 *SYSGEN End*

SYSGEN is ended after common device definition and control given back to the console typewriter.

9.12.6 *Example SYSGEN*

This section comprises an example of the console typewriter listing produced during the execution of SYSGEN.

The following diagram illustrates the composition of the device classes and tasks in this example and shows the relationships between them. Note that the TC included in special device classes S01 and S02 are used by the Configurator.


Terminal device class T01 containing KB, TP, DI and DY.
Terminal device class T02 automatically generated from T01.
Terminal device class T03 automatically generated from T01.
Task U0 for a work position.
Terminal device class T04 containing KB, GP and DY.
Task T0 for a work position.
Special device class S01 containing TC, SI and SO.
Special device class S03 containing TW and LP.
Task TB for the CREDIT debugging program
Special device class S02 containing TC (for Configurator)

```
S:ASG   /CA, DK
S:RUN   SYSGEN

S Y S G E N
SYSTEM GENERATION PROGRAM 6800
RELEASE # 3.1
SUPPORTING TOSS RELEASE # 8.1
**************************************

ASSEMBLY LIST : N
MONITOR NAME : BTMON2
MONITOR CONFIGURATION PROGRAM : N
MEMORY MANAGEMENT : N
PROGRAM LOADING FROM DISK : Y
MONITOR INITIALIZATION PROGRAM : Y
DEBUGGER : N
SUBROUTINE CALL INTERPRETER : N
CREDIT APPLICATION : Y
DATA MANAGEMENT : Y
******************** DEVICE ABBREVIATIONS ********************
DEVICE LIST (Y/N) : Y
KB      KEYBOARD
TP      TELLER TERMINAL PRINTER
GP      GENERAL PRINTER
DN      NUMERIC DISPLAY
DI      INDICATOR DISPLAY/KEYBOARD LAMPS
DY      VIDEO/PLASMA DISPLAY
LT      LOCAL TERMINALS
RT      LOCAL AND REMOTE TERMINALS
CT      CUSTOMER OPERATED TELLER
MS      MAGNETIC STRIPE UNIT
OD      OPTICAL DOCUMENT READER
TC      TAPE CASSETTE
SI      SOP INPUT
SO      SOP OUTPUT
MT      MAGNETIC TAPE UNIT
TW      TYPEWRITER
DU      DISK UNIT
```

```
LP    LINE PRINTER
FD    FLEXIBLE DISC
CR    CARD READER
II    INTERTASK COMMUNICATION INPUT
IO    INTERTASK COMMUNICATION OUTPUT
D1    DATA COM DRIVER DRDC07
D2    DATA COM DRIVER DRDC15
D3    DATA COM DRIVER DRDC17
D4    DATA COM DRIVER DRDC81
D5    DATA COM DRIVER DRDC82
D6    USER OWN DATA COM DRIVER 1
D7    USER OWN DATA COM DRIVER 2
CC    CASSETTE CHANGER

DEFINE DEVICES :  KB, TP, GP, DI, DY, RT, TC, SI, SO, TW, DU, LP
KEYBOARD 6236 IN SYSTEM : N
**********D R I V E R S**********

********** DRKB01 :

KEYBOARD TIMEOUT :  N
ECHO FUNCTION :  Y
STANDARD READ:  Y
NUMBER OF ZEROES FOR MULTIPLE ZERO KEY  :  2
COMPLETION OF READ REQUEST AT POWER ON  :  N
CIRCULAR INPUT BUFFER SIZE:  24
CODE CONVERSION/8-BIT SETTING :  Y

**********DRTP02 :  S
**********DRGP01 :  S
**********DRDI01 :  S
**********DRDY01 :

GRAPHIC MODE  :  N
ONLY PTS 6344 :  N
LOWER CASE CHARACTERS  :  N
ECHO OF EOR-KEY  :  N
SUPPRESSION OF TRAILING BLANKS  :  N
CURSOR STEADY ON 6386 (ELSE TWINKLING)  :  Y
**********DRRT01 :  S
**********DRTC01 :
```

```
NUMBER OF CASSETTE DRIVES: 2
ORDERS LOCK, ERASE AND REVERSE ONE BLOCK : Y

**********DRSOP1 :

EXTRA SOP INPUT DWT : Y
FLASH FUNCTION : N
**********DRTW01 : S
**********DRDU01 :

READ AFTER WRITE : Y
NUMBER OF DISK UNITS: 1
UNIT 1 PTS 6876 DISC : N
**********DRLP01:

LP ON PROGRAMMED CHANNEL (ELSE MUX) : N

********** L K M – P R O C E S S O R S **********

DELAY/DELAY AND ACTIVATION: Y
DYNAMIC BUFFER ALLOCATION : Y
ABORT FUNCTION : Y
GET/SET TIME : Y
ATTACH/DETACH: N

********** DATA MANAGEMENT PARAMETERS**********

MAX NUMBER OF DISK FILES : 20
MAX NUMBER OF COMMON DISK-FILE FILE CODES : 20
MAX NUMBER OF DISK BUFFERS : 5
EXCLUSIVE ACCESS HANDLING : Y
INDEXED ACCESS: N
MAX NUMBER OF CURRENCY BUFFERS : 2
MAX NUMBER OF RECORDS WITH EXCLUSIVE ACCESS : 9
LOCAL DISC FILE CODES : Y
MAX NUMBER OF LOCAL DISC FILE CODES : 2
********** DEFINE TERMINAL DEVICE CLASSES**********

**********T01 : Y
********* KB : Y
```

```
FILE CODE : 20
DEVICE ADDRESS: 1,5
PTS NUMBER : 6234
BACKSPACE KEY : FF
CANCEL KEY: FF
TRIPLE ZERO KEY : FF
MULTIPLE ZERO KEY : 3A
ECHO DEVICE : DY
CODE CONVERSION : N

******** TP: Y

FILE CODE : 30, 31, 32
DEVICE ADDRESS : 2
******** GP: N
******** DI : Y

FILE CODE : 20
DEVICE ADDRESS: 1
PTS NUMBER : 6234
******** DY : Y

FILE CODE : 50
DEVICE ADDRESS: 4
PTS NUMBER : 6351
NUMBER OF LINES : 08
NUMBER OF CHARACTERS/LINE : 36
**********T04 : Y
******** KB : Y

FILE CODE : 20
DEVICE ADDRESS : 1,5
PTS NUMBER : 6234
BACKSPACE KEY : FF
CANCEL KEY: FF
```

```
TRIPLE ZERO KEY : FF
MULTIPLE ZERO KEY: 3A
ECHO DEVICE  : DY
CODE CONVERSION : N
******** TP  : N
******** GP  : Y

FILE CODE : 30
DEVICE ADDRESS: 2
******** DI  : N
******** DY  : Y

FILE CODE : 50
DEVICE ADDRESS: 4
PTS NUMBER : 6344
NUMBER OF LINES  : 24
NUMBER OF CHARACTERS/LINE  : 80
********** T05 : N

********** DEFINE ECHO DEVICE CLASSES **********
********** E01 : N

********** DEFINE SPECIAL DEVICE CLASSES **********

********** S01 : Y
******** TC  : Y

FILE CODE : 12
UNIT NUMBER  : 1
******** SI : Y

FILE CODE : 10, 14
******** SO  : Y
```

```
FILE CODE : 11
********* TW  : N
********* LP  : N
********************  S02 : Y
********* TC  : Y

FILE CODE : 13
UNIT NUMBER  : 2
********* SI : N
********* SO  : N
********* TW  : N
********* LP  : N
********************  S03 : Y
********* TC  : N
********* SI : N
********* SO  : N
********* TW  : Y

INPUT FILE CODE : 21
OUTPUT FILE CODE : 31
BACKSPACE KEY : 5F
CANCEL KEY : 5E
END OF RECORD KEY : 0D
********* LP  : Y

FILE CODE : 16
********************  S04: N

EXTRA MONITOR MODULES  : N
```

```
**********TASK DEFINITION**********

**********TASK 01 : Y

TASK IDENTIFICATION : TB
TASK LEVEL : 55
TASK CONFIGURATION LIST : S1
*S3
*

**********TASK 02 : Y

TASK IDENTIFICATION : T0
TASK LEVEL : 60
TASK CONFIGURATION LIST : T4, 1L
*

**********TASK 03 : Y

TASK IDENTIFICATION : U0
TASK LEVEL : 60
TASK CONFIGURATION LIST : T1, 1R
* T2, 1R
* T3, 1R
*

********** TASK 04: N

COMMON DEVICES : Y
CONFIGURATION LIST : S1
* S2
*

SYSTEN ENDED
S:KPF /CA,BTMON2
S:ASG /D0,DK,BTMON2
```

9.13 XRF

XRF is used to obtain a listing, on the line printer, of cross-references within a source module on disk, i.e. a listing of symbols used in the module, including line numbers of both their definition and references. The utility can be used for both CREDIT and ASSEMBLER modules.

The listing produced is headed SYMBOL DEF REFERENCES, and under this heading are listed the symbol names, with line numbers as described above. Where a symbol is referred to more than once, the line numbers are printed in ascending sequence. If a symbol is undefined (i.e. referred to but not defined), three asterisks appear on the listing under the DEF column heading. If a symbol is multidefined, only the lowest line number of the definition appears under DEF. The remaining line numbers for the definition statements appear under the REFERENCES, together with the line numbers of the references, but with a plus sign (+) printed after the line number.

The following steps must be taken to execute the XRF utility:

Key in the control command XRF xxxxxx, where xxxxxx is the name of the source module for which the cross-reference list is required.

10. CATALOGUED PROCEDURES

10.1 Introduction

A catalogued procedure is a string of control commands held in a special format on disk. Such a procedure may be invoked by simply keying-in the procedure name on the console typewriter or any other device to which file code /E0 is assigned. At the end of the procedure control is automatically handed back to the typewriter.

Predefined control commands may of course be invoked simply by assigning file code /E0 to a library file containing control commands. However, a catalogued procedure can be used more flexibly because the basic control commands in the procedure can be automatically modified according to parameters keyed-in when the procedure is invoked.

10.2 Invoking Catalogued Procedures

The general form of a catalogued procedure invokation is:

procedure-name ␣[formal-parameter = actual-parameter[,formal-parameter = actual-parameter] . . . .]

The format of the procedure name is defined in appendix A. It has a maximum length of eight characters. A formal parameter is a character string of one to four characters. An actual parameter is a character string which will be built into a control command before it is obeyed by the CCI. The length and value of an actual parameter depends upon the type of control command to which it applies.

When a catalogued procedure is invoked, the CCI will search for a file named M:PROC in the library of the current userid. If this file does not exist the CCI will locate the standard M:PROC file in the system library. The CCI will then search the M:PROC file for the named catalogued procedure.

Having found the catalogued procedure the CCI will search the procedure for the formal parameters specified in the invokation. It will replace these parameters with the corresponding actual parameters and will create a disk file containing the modified procedure. (File code /EE will be used for this file. The user should avoid using this file code for any other purpose).

This file of control commands is then obeyed by the CCI and control is then handed back to the device from which the procedure was invoked.

10.3 The Catalogued Procedure File

Catalogued procedures must be held in a special file called M:PROC. There is an M:PROC file in the system library which contains any standard procedures supplied as part of DOS6800 System Software. An M:PROC file may also exist in each user library to hold any catalogued procedure created by the user.

Each M:PROC file may contain several procedures. Each procedure must start with a procedure name of upto five characters (including the first character which must be $) and end with the word END followed by :EOF. The following example illustrates these points:

```
$PROC1
   commands of procedure 1
END
$PROC2
   commands of procedure 2
END
:EOF
```

Procedures may be updated or new procedures may be inserted by using the Line Editor (see section 6.11.3 for an example).

10.4 The Use of Parameters

Formal parameters may be used in either the command mnemonics or parameter fields of control commands in catalogued procedures. Neither formal parameters nor comments may be used with the procedure name or END command.
Formal parameters may be used in three formats:

```
@Formal-parameter
@Formal-parameter = character-string
@Formal-parameter =
```

The user may, if it is appropriate, omit the specification of certain parameters in a procedure invokation. In this case the action taken by the CCI on the corresponding formal parameters will depend upon the format used. The rules for each format are described below. However, if the actual parameter is specified in the invokation the action taken by the CCI is the same for each format: the actual parameter simply replaces the formal parameter (including @ and =).

If the format @formal-parameter is used the CCI will simply omit the parameter if an actual parameter is not specified. For example:

```
$PR1
RDS
ASM  /S, @NLIST
LKE
END
:EOF
```

If the procedure is invoked by keying-in $PR1 the ASM control command will become simply ASM /S, and the comma will be ignored as it is not followed by a parameter. If the procedure is invoked by keying-in $PR1 NLIST=NL the ASM control command will become ASM /S,NL.

If the format @formal-parameter=character-string is used, the character-string will be used as a default value if an actual parameter is not specified in the invokation.
For example:

```
$PR2
LED   PROG,@FC=/S
ASM  @FC=/S
LKE
END
:EOF
```

If this procedure is invoked by keying-in $PR2 the LED and ASM control commands will become LED PROG,/S and ASM /S respectively. If the procedure is invoked by keying-in $PR2 FC=/A the LED and ASM control commands will become LED PROG,/A and ASM /A respectively.

If the format @formal-parameter= is used the complete line will be ignored if an actual parameter is not specified.
For example:

```
$PR3
@COM=
LED   @PNAM=
ASM   /S
INC   @MOD=
LKE
END
```

If this procedure is invoked by keying-in $PR3 COM=RDS the following sequence of control commands will be obeyed:

```
RDS
ASM   /S
LKE
```

If the procedure is invoked by keying-in $PR3 PNAM=PRGRM,MOD=MAIN the following sequence of control commands will be obeyed:

```
LED   PRGRM
ASM   /S
INC   MAIN
LKE
```

In the last example the Linkage Editor would read input data from the file with file code /E0 (standard for console typewriter). It is not possible to include input data in catalogued procedures. Processors or utilities (e.g. LED, SYSGEN) which read input data will always access the device with file code /E0.

Catalogued procedures may be invoked from within other catalogued procedures. However, the procedures will not be "nested". That is, control is not returned to an invoking procedure at the end of an invoked procedure.

If an invoked catalogued procedure cannot be found by the CCI the invokation is ignored and the CCI continues processing the original procedure.

10.5 Error Messages

The following error messages may be output by the CCI when processing a catalogued procedure:

```
ERROR IN PROCEDURE DEFINITION
ERROR IN PROCEDURE GENERATION (procedure is not correct)
PROCEDURE NOT CATALOGUED (procedure unknown in M:PROC file)
M:PROC NOT CATALOGUED
I/O ERROR
```

10.6 $PCAS

$PCAS is a catalogued procedure which is supplied with DOS6800 System Software and is held in the system library. The procedure runs the utilities CPLGEN and JESPER to create a cassette containing a Monitor load module and/or an application load module.

JESPER is used to copy a load module from disk onto cassette in a form suitable for loading into memory during TOSS system start. If the load module is the TOSS Monitor then it must be preceded on the cassette by an IPL. This is written by CPLGEN.

The $PCAS invoking command has the following format:

$PCAS followed by one space and then a combination of the following parameters, separated by a comma between each one:

M = mname
MU = userid
A = aname
AU = userid
C = cname
CU = userid
D = dname
DU = userid

where mname is the name of the Monitor load module to be copied
aname is the name of the application load module to be copied
cname and
dname are the names of the data files (i.e. configuration data) to be copied, in the sequence C then D.
userid is the user identity under which the file to be copied resides; this need not be specified if it is the current user.

The following combinations of parameters are valid (see also below):

A M M+A M+A+C M+A+C+D C C+D

Using $PCAS the necessary load modules and data can be copied onto one, two or three cassettes. The way in which these files can be distributed among the cassettes and the sequence in which the files must occur is explained below.

For CREDIT applications:

One cassette	Two cassettes	Three cassettes
Monitor Application [Monitor Configuration data] CREDIT configuration data	Monitor Application	Monitor
	Monitor Configuration data CREDIT configuration data	Application CREDIT configuration data [Application data]
or	or	
Monitor Application CREDIT configuration data [Application data]	CREDIT configuration data [Application data]	

For Assembler applications:

One cassette	Two cassettes
Monitor Application [Monitor configuration data] [Application data]	Monitor Application
	Monitor configuration data [Application data]

Note that as many runs of $PCAS are necessary as there are cassettes to be created. In addition, it must be noted that $PCAS assumes all the files to reside on the same disk volume: it is the users responsibility to ensure that this is the situation prior to running the utility.

APPENDIX A : PARAMETER SYNTAX

This appendix defines the syntax of the various parameters used in the control commands, control messages, Line Editor commands and SYSGEN responses described in this Manual. The notation conventions are described in section 1.9.

alphanumeric-character : : = { letter / decimal-digit }

decimal-digit : : = 0|1|2|3|4|5|6|7|8|9

decimal-number : : = decimal-digit . . . .

device-name : : = CR0D|LP0F|MT0C|TK0E|TK1E|TY10

disk-number : : = /F0|/F1|/F2|/F3/F8/F9/FA/FB

file-code : : = /1|/2|/3 . . . . . /ED|/EE|/EF

file-name : : = ISO-7-character . . . . . .

hexadecimal-digit : : = 0|1|2|3|4|5|6|7|8|9|A|B|C|D|E|F

hexadecimal-number : : = hexadecimal-digit . . . . .

label : : = letter [alphanumeric-character] . . . . . .

letter : : = A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|O|R|S|T|U|V|W|X|Y|Z

line-number : : = decimal-number

module-name : : = letter[alphanumeric-character] . . . . .

procedure-name : : = $||letter[alphanumeric-character] . . . . .

unit-address : : = 0C|0D|0E|0F|10|1E

userid : : = letter[alphanumeric-character] . . . . .

utility-name : : = CPLGEN| DUMPA| DMPGEF| DMPGEN| JESPER| OBX | PM6800
PRDUMP | RUM | SUM | SYSGEN | XRF

APPENDIX B : ERROR REPORTS

Introduction

During the execution of control commands, processors and utilities (i.e. programs) errors may occur which result in the output of error reports on the console typewriter or printer. Reports which are unique to particular control commands, processors or utilities are described in sections 6.12, 8.2.4, 8.3.4 and 9.1 of this Manual. Reports which may be output during the execution of **any** type of program are described in this appendix.

Abort Reports

When a program is aborted a report may be output on the device with file code /1 (standard for console typewriter) or file code /2 (standard for printer) specifying the following:

- The location at which the program abort occurred:
  PROG ABORTED AT XXXX
- The reason for the abort:
  NOT WIRED INSTRUCTION
  OVERFLOW IN SIMULATION ROUTINE SAVE AREA
  BUFFER AREA DESTROYED
  TOO MANY SCHEDULED LABELS
  OPERATOR ABORT
  BUFFER ALLOCATION OVERFLOW
  DISK OVERFLOW
  DISK QUEUE OVERFLOW
  MEMORY OVERFLOW DURING LOADING PHASE
- The contents of the PSW and the registers at the moment of the abort.

Peripheral Unit Error Report

When an error occurs during an I/O operation on a peripheral unit the following report is output on the device with file code /1 (standard for console typewriter):

PU ␣ device-name,status-code,RY

where:

device-name	identifies the peripheral device (see section 4.1)
status-code	indicates the type of error (see status code section below)
RY	indicates that the user may retry the operation.

The user may respond to such a report with an RY control message (retry an I/O operation) or an RD control message (release device). See section 7.4.

Disk Error Report

When an error occurs during an I/O operation on a disk unit the following report is output on the device with file code /1 (standard for console typewriter):

DKER ␣ unit-address ␣ location ␣ status-code

where:

unit-address identifies the disk unit (second two digits of device name -- see section 4.2).
location indicates in a single number the cylinder, track and sector number at which the error occurred.
status-code indicates the type of error (see status code section below).

Status Codes

The status code is printed as a hexadecimal representation of a 16 bit status word. The code must be translated into binary form before it is analysed by the user. The analysis of the resulting binary word is described below.

The following tables show the significance of each bit in the word. The bits are numbered from 0 to 15. The leftmost bit is counted as 0. If bits 0 and 1 are both set to 1 the Software Status Table should be referred to. Otherwise the Control Unit Status Table should be referred to.

Software Status Table

Bit set to 1	Significance
0	See remaining bits
1	See remaining bits
2–10	Not significant
11	Function unknown or not compatible with device
12	Buffer size is illegal
13	Buffer address is illegal
14	Device attached to other program
15	Illegal file code or non-existing

Control Unit Status Table

Bit set to 1	Significance	TY	CR	DK	LP	TK	MT
0	–						
1	ready			X		X	X
2	rewind						X
3	tape mark has been read					X	X
4	no data					X	
5	on cylinder load point			X		X	X
6	seek error			X			
	write unable					X	X
7	A or B side					X	
8	Device address			X		X	X
9	Device address			X		X	X
10	EOT					X	
11	program error			X		X	X
12	Incorrect length		X	X		X	X
13	Parity error					X	
	Data fault		X	X			X
14	throughput error	X	X	X		X	X
15	not operable	X	X	X	X	X	X

where:

TY is console typewriter
CR is card reader
DK is disk
LP is line printer
TK is cassette unit
MT is mag. tape unit

APPENDIX C : ISO CODE CHARACTER SET

Recommendation R646

CHARACTER/FUNCTION DECLARATION

HEX.	ISO MNEMONIC	DECLARATION
0.0	NUL	NULL
0.1	SOH	START OF HEADING
0.2	STX	START OF TEXT
0.3	ETX	END OF TEXT
0.4	EOT	END OF TRANSMISSION
0.5	ENQ	ENQUIRY
0.6	ACK	ACKNOWLEDGE
0.7	BEL	**BELL**
0.8	BS	BACKSPACE
0.9	HT	HORIZONTAL TAB
0.A	LF	**LINE FEED**
0.B	VT	VERTICAL TAB
0.C	FF	**FORM FEED**
0.D	CR	**CARRIAGE RETURN**
0.E	SO	SHIFT OUT
0.F	SI	SHIFT IN
1.0	DLE	DATA LINK ESCAPE
1.1	DC1	**X-ON READER**
1.2	DC2	**X-ON PUNCH**
1.3	DC3	**X-OFF READER**
1.4	DC4	**X-OFF PUNCH**
1.5	NAK	NEGATIVE ACKNOWLEDGE
1.6	SYN	SYNCHRONOUS IDLE
1.7	ETB	END OF TRANSMISSION BLOCK
1.8	CAN	CANCEL
1.9	EM	END OF MEDIUM
1.A	SUB	SUBSTITUTE
1.B	ESC	ESCAPE
1.C	FS	FILE SEPARATOR
1.D	GS	GROUP SEPARATOR
1.E	RS	RECORD SEPARATOR
1.F	US	UNIT SEPARATOR
2.0	SP	SPACE
7.F	DEL	DELETE/RUB OUT

There are 12 positions variable for national usage marked with 

Per I/O device the character-set may vary.

$b_7 b_6 b_5$ →	000	001	010	011	100	101	110	111
$b_4 b_3 b_2 b_1$ ↓ col / row	0	1	2	3	4	5	6	7
0000	0	NUL	DLE	SP	0	@	P	`	p
0001	1	SOH	DC1	!	1	A	Q	a	q
0010	2	STX	DC2	"	2	B	R	b	r
0011	3	ETX	DC3	#	3	C	S	c	s
0100	4	EOT	DC4	$	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v
0111	7	BEL	ETB	'	7	G	W	g	w
1000	8	BS	CAN	(	8	H	X	h	x
1001	9	HT	EM	)	9	I	Y	i	y
1010	A	LF	SUB	*	:	J	Z	j	z
1011	B	VT	ESC	+	;	K	[	k	{
1100	C	FF	FS	,	<	L	\	l	|
1101	D	CR	GS	−	=	M	]	m	}
1110	E	SO	RS	.	>	N	^	n	~
1111	F	SI	US	/	?	O	−	o	DEL

APPENDIX D : SYSTEM CONSTRAINTS

Introduction

This appendix describes some constraints imposed on the user by the PTS6000 hardware and DOS6800 System Software. Hints on how to stay within these constraints are also given.

Constraints

Briefly the constraints are as follows:

(I) The number of the file codes in the Monitor's file code table is less than the maximum number of valid file codes (/1—/EF).

(II) The number of userids on a single disk (including system userids) may not be greater than 149.

(III) The number of files in the library of a single userid may not be greater than 199.

(IV) The number of words in a single file may not be greater than 320K (K=1000 decimal).

(V) The number of words available to the user on a single disk (excluding granules 0 and 1) is less than 1296K. If the disk is a system disk the available space will be reduced by about 13% for the SAG library and by a further 60% for the libraries TOSSUTIL, INT:PROD and TOSSWORK if they are present. This figure is further reduced by 1.6K (one granule) for each userid declared on that disk.

Hints

The following numbered hints refer to the corresponding constraints in the preceding section:

(I) Scratch temporary files when they are no longer required (control command SCR). BYE scratches all temporary files automatically.

(II) Delete userids that are no longer required (DLU). If necessary save any files in the libraries of these userids before deleting the userids (INC, MOV, SVD, SVU, PCH, PLD or POB).

(III) Delete files in the relevant library that are no longer required (DEL). If necessary save these files before deleting them (INC, MOV, SVD, SVU, PCH, PLD or POB).

(IV) The only instance in which an individual file can be reduced in size is in the case of "checkerboarding" in an /ØBJCT file — see section 6.7.

(V) Delete files on the relevant disk that are no longer required (DEL). If necessary save these files onto another disk or onto tape before deleting them (INC, MOV, SVD, SVU, PCH, PLD or POB).

Checkerboarding in /ØBJCT files should also be eliminated — see section 6.7.

Scratch temporary files when they are no longer required (SCR). BYE scratches all temporary files automatically.

APPENDIX E : INDEX