

UP-3917-3.02

INITIAL DRUM SETUP ROUTINES

UP-3917-3.02

This manual is published by the UNIVAC Division of Sperry Rand Corporation in loose leaf format as a rapid and complete means of keeping recipients apprised of UNIVAC[®] Systems developments. The UNIVAC Division will issue updating packages, utilizing primarily a page-for-page or unit replacement technique. Such issuance will provide notification of hardware and/or software changes and refinements. The UNIVAC Division reserves the right to make such additions, corrections, and/or deletions as in the judgment of the UNIVAC Division, are required by the development of its respective Systems.

CONTENTS

CONTENTS	1-1 to 1-1
1. INTRODUCTION	1-1 to 1-2
1.1. SCOPE	1-1
1.2. PURPOSE	1-1
1.3. HARDWARE FACILITIES	1-1
1.3.1. FASTRAND Drum Units	1-1
1.3.2. Card Readers	1-2
1.3.3. Printers	1-2
2. DESCRIPTION	2-1 to 2-8
2.1. GENERAL	2-1
2.2. SYSTEMS DIRECTORY	2-1
2.2.1. Systems Directory Descriptor	2-2
2.2.2. Systems Directory Format	2-2
2.3. DATA FILES CLASS	2-2
2.3.1. File Directory Descriptor	2-4
2.3.2. File Directory Format	2-4
2.4. SYSTEMS AREA CLASS	2-5
2.4.1. Program Directory Descriptor	2-8
2.4.2. Program Directory Format	2-8
3. PROGRAMMING PROCEDURES	3-1 to 3-3
3.1. CLEAR CARD	3-1
3.2. CONFIGURATION CARD	3-1
3.3. SDD CARD	3-2
3.4. CLASS CARD	3-2
3.5. SENTINAL CARD	3-3
4. OPERATING PROCEDURES	4-1 to 4-4
4.1. INSTRUCTIONS FOR LOADING	4-1
4.2. IDMS LIST	4-3
4.2.1. Clear Card Line	4-3
4.2.2. Configuration Card Line	4-3
4.2.3. SDD Card Lines	4-3
4.2.4. Class Card Lines	4-3
4.2.5. Sentinal Card Line	4-3
4.3. ERROR RECOVERY	4-3

1. INTRODUCTION

1.1. SCOPE

This document describes the Initial Drum Setup (IDMS) routine for the UNIVAC 1050 Mass Storage System. Requisite to operating a UNIVAC 1050 Mass Storage System under control of UNIVAC-supplied software is a thorough understanding of the information, programming procedures, and operating procedures herein contained.

1.2. PURPOSE

The Initial Drum Setup routine defines areas on the drum wherein specific information is to be stored. It also establishes the hardware configuration of the particular system and writes a drum locator and loader routine (DLL) into the lower locations of the first drum unit. IDMS prints a list of the areas that it defines. It is assumed that nothing of value is on the drum when IDMS is run.

1.3. HARDWARE FACILITIES

The following hardware facilities are required when running IDMS.

1.3.1. FASTRAND Drum Units

Any of the following FASTRAND drum units may be utilized with a maximum of eight of the same or different types controlled by one Control Unit.

- FASTRAND I Drum Unit
- FASTRAND II Drum Unit

The Fastbands option may be added to any of the above.

1.3.2. Card Readers

Any one of the following may be used:

- 0701 Row Reader
- 0704 Column Reader
- 0706 Column Reader
- 1004 Reader (80 or 90) on line.*

1.3.3. Printers

Any one of the following may be used:

- 0755 Printer (buffered or unbuffered)
- 1004 Printer (with standard UNIVAC 1050 code wheel) on line.*

* If used, SENSE SWITCH 1 on the central processor must be turned on; otherwise, this switch must be off.

2. DESCRIPTION

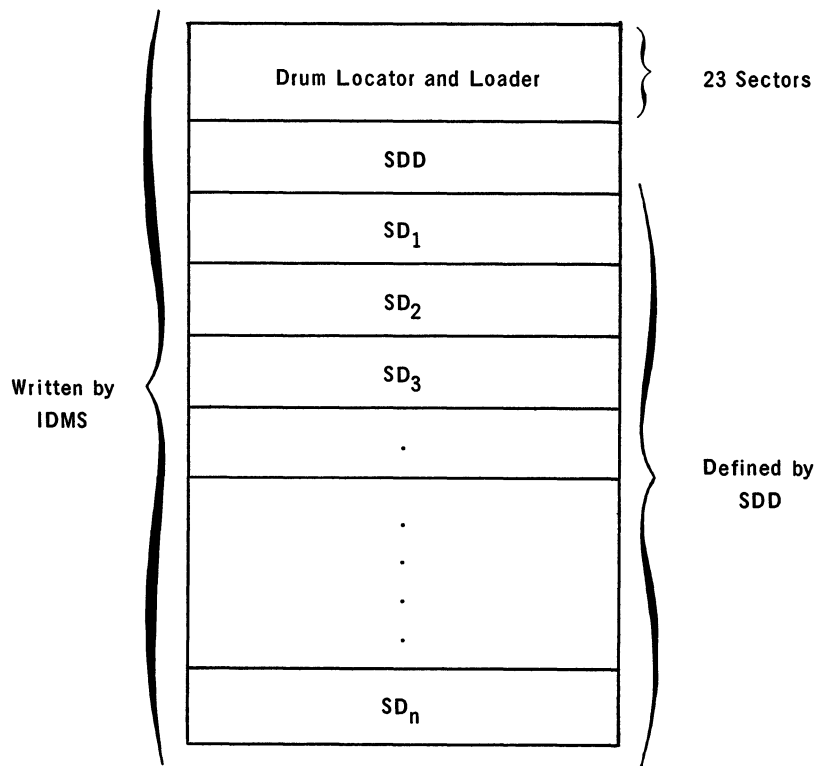
2.1. GENERAL

When setting up the drum, the user must first map out areas in which specific information will reside. Areas must be mapped out for data files and instructions (both user and systems programs). To do this, a network of "Directories" is employed. These directories are written onto the drum by IDMS in accordance with various parameter cards.

2.2. SYSTEMS DIRECTORY

A Systems Directory (SD) uniquely names and defines the boundaries of an area on the drum. Such an area is called a class. For each class so named and defined there must be *one* directory entry. This SD entry will establish the class as either a Data File Area or a Systems Area (in which both library and user routines are to reside). The user has the option of assigning classes in both normal and Fastband areas, and designating a class to have write lockout.

Each Systems Directory is defined by an IDMS parameter card. The area on the drum that is to contain the SD's themselves is defined by a Systems Directory Descriptor (SDD). This descriptor (also defined by an IDMS parameter card) will precede the SD's on the drum in the following manner:



2.2.1. Systems Directory Descriptor

The format of the Systems Directory Descriptor as IDMS writes it on the drum is as follows:

<u>CHARACTERS</u>	<u>CONTENT</u>
0-7	SYSDIRECT
8-11	Drum address of first SD entry
12-15	Drum address of first sector beyond area reserved for SD entries
16-19	Drum address of the first free sector within the SD area that is available for additional SD entries

2.2.2. Systems Directory Format

The format of a Systems Directory itself as IDMS writes it onto the drum is as follows:

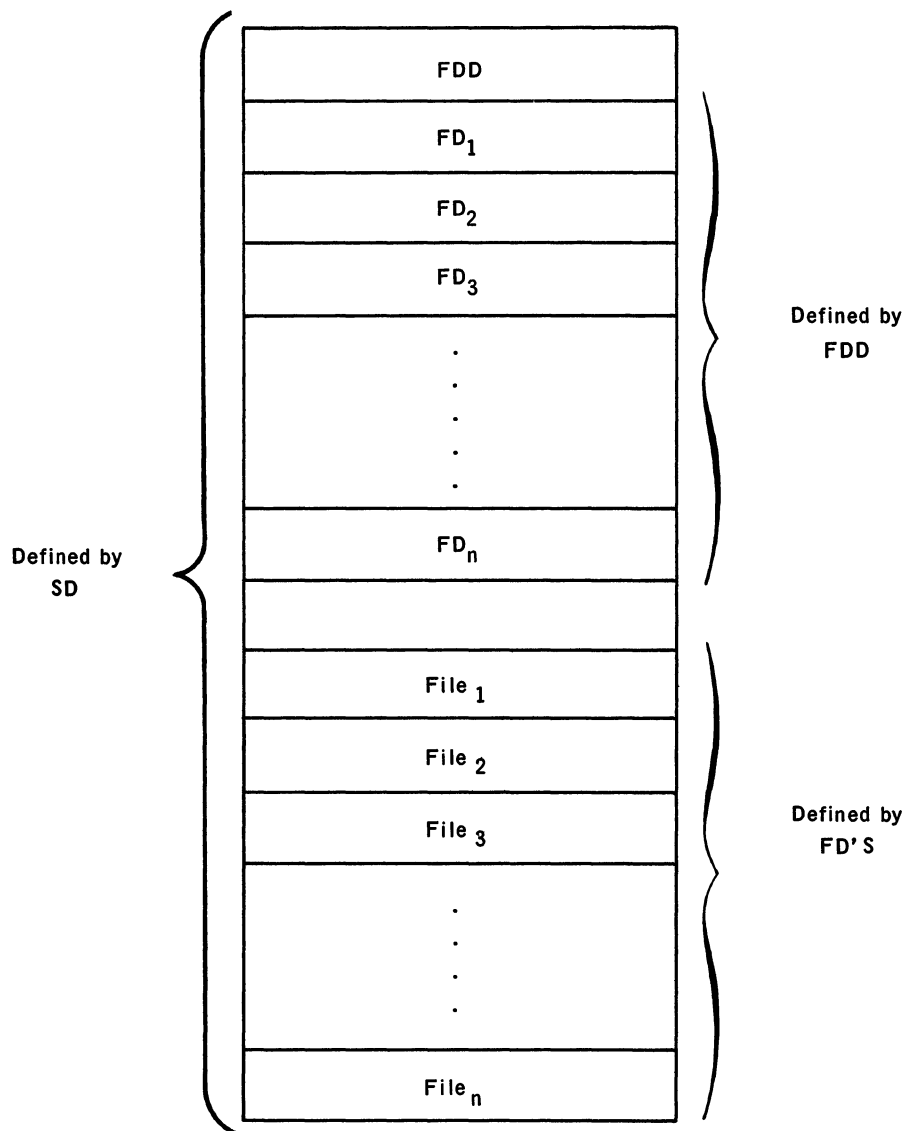
<u>CHARACTERS</u>	<u>CONTENT</u>
0-7	Name of the class to be reserved by this SD
8-11	Drum address of the first sector of this class
12-15	Drum address of the first sector beyond this class
16	Drum type: 0 for both FASTRAND I and II

2.3. DATA FILES CLASS

Each class established by the SD's to hold data files is divided into two areas:

- (1) File Directory area
- (2) File area

As a Systems Directory is written to define an entire drum class, a File Directory (FD) must be written for each file within the class. Preceding the File Directories is a File Directory Description (FDD) which defines the boundaries of the area in which the FD's will reside. A class established for data files will be formatted in the following manner:



It should be noted that the FDD is written onto the drum by IDMS. The FD's, however, are written by the *File Control Routine*.

2.3.1. File Directory Descriptor

The format of the File Directory Descriptor as it is written onto the drum by IDMS is as follows:

<u>CHARACTERS</u>	<u>CONTENT</u>
0-7	FLEDIRCT
8-11	Drum address of first FD entry, i.e., the FDD
12-15	Drum address of first sector beyond the area reserved for FD entries
16-19	Drum address of first free sector within the FD area that is available for additional FD entries

2.3.2. File Directory Format

The format of the File Directory as it is written onto the drum by a data file control routine is as follows:

<u>CHARACTERS</u>	<u>CONTENT</u>
0-7	Name of this file
8-11	Drum address of the first data sector in the file
12-15	Drum address of first sector beyond the file area
16-19	Drum address of the next free sector available within the file area
20-23	Drum address of the next free sector available within a file area for secondary (chained) elements
24	Head position indicator: 1 = File must start at the beginning head position 0 = File can start at any head position

2.4. SYSTEMS AREA CLASS

A class that is established by the SD's to facilitate system operation, rather than data file storage, is called a Systems Area (SA). For each SA, IDMS writes both the FD and the FDD on the drum.

Systems Areas are used as:

- a storage area for the UNIVAC software Systems Programming File (requires one SA)
- a scratch area by UNIVAC software (requires one SA)
- storage areas (optional) for both source and object code worker program files (User Library Files).

The Systems Programming File will reside in one Systems Area. This SA must be large enough to permit maximum expansion of the Systems Programming File.

The Systems Programming File is identified by asterisks in the first two positions of its SA (class) name, as follows:

**nnnnnn
where

nnnnnn = any six-character alphanumeric label,
assigned by the programmer.

UNIVAC software requires one Systems Area to be established for use as a scratch area. The Scratch SA should be at least as large as the largest SA that will be software maintained.

The Scratch SA is identified by an asterisk in the first position and a dollar sign in the second position of its SA (class) name, as follows:

*\$nnnnnn
where

nnnnnn = any six-character alphanumeric label,
assigned by the programmer.

Up to 26 Systems Areas may be established in which to store User Library Files. A User Library File is identified by an asterisk in the first position and a unique alphabetic character in the second position of its SA (class) name, as follows:

A
.
*.nnnnnn
.
Z
where

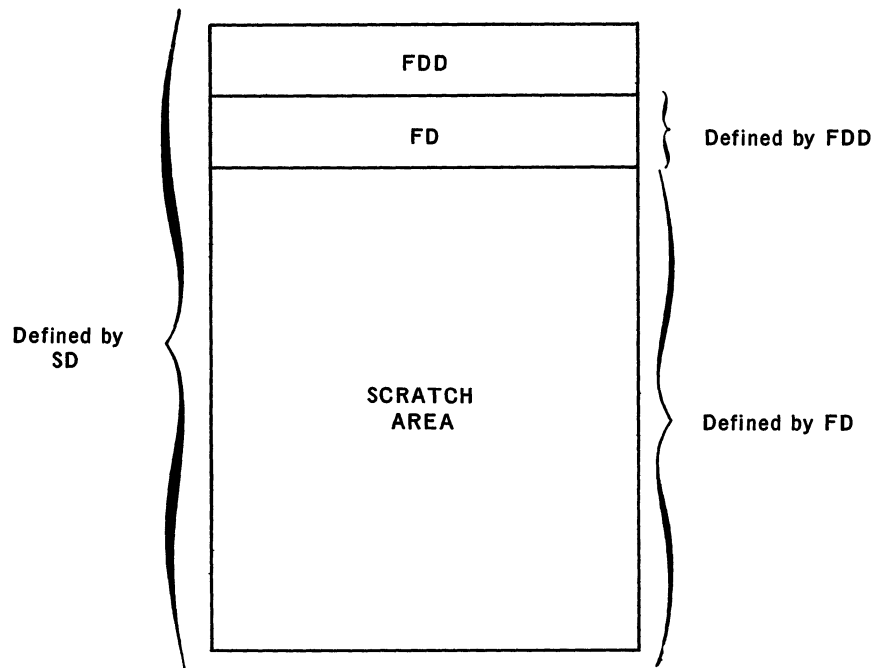
nnnnnn = any six-character alphanumeric label,
assigned by the programmer.

Note: No two User Library File Systems Areas can have the same alphabetic character in the second position of their names.

The Scratch SA is divided into two areas:

- (1) File Directory area*
- (2) Scratch area

The SA established for use as a scratch area will be formatted as follows:



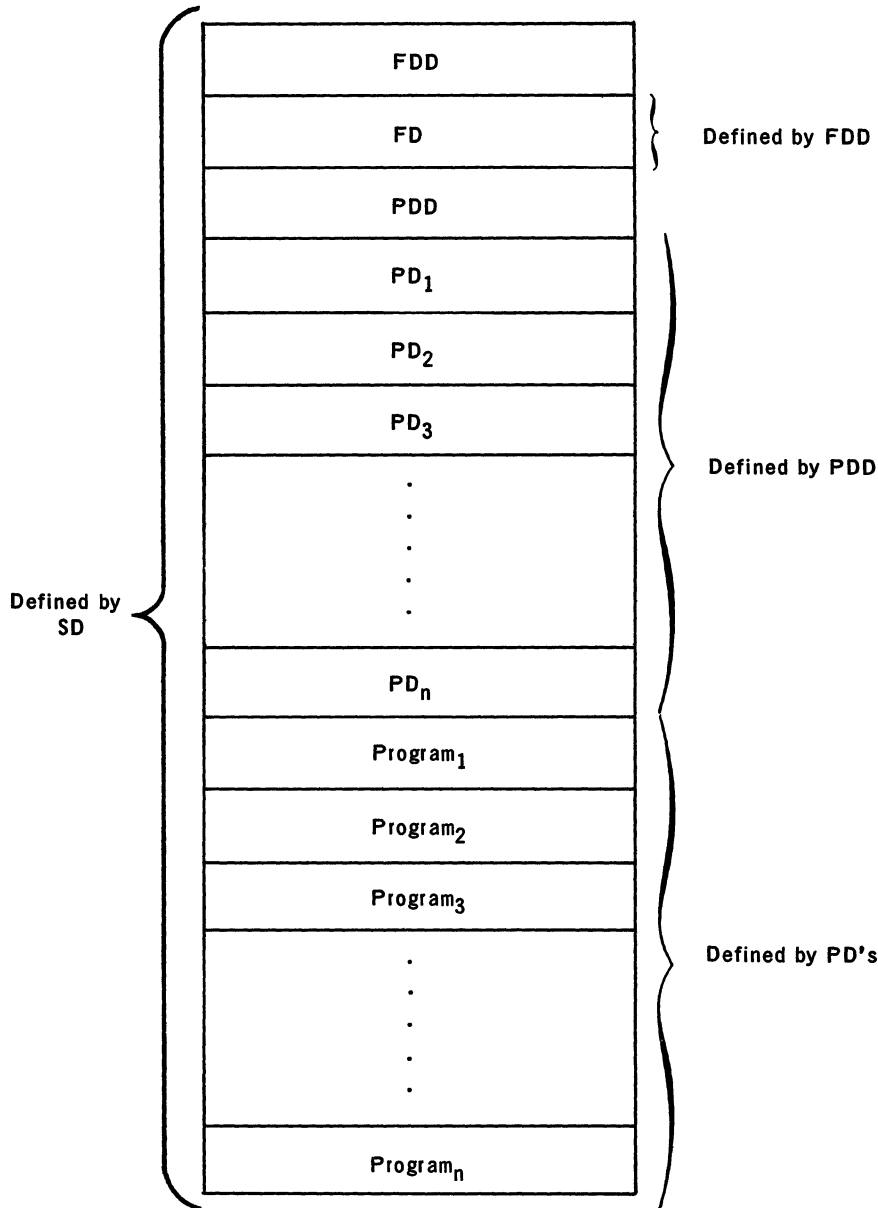
An SA that is established to hold a program file is divided into three areas:

- (1) File Directory area*
- (2) Program Directory area
- (3) Program File area

As a File Directory (FD) is written to define each file within a class, a Program Directory (PD) must be written for each program within a class. Preceding the Program Directories is a Program Directory Descriptor (PDD), which defines the boundaries of the area in which the PD's will reside. The PD's and PDD's will be established and maintained by the AJAXF and OPUSF program file maintenance routines.

* As explained in Sections 2.3.1. and 2.3.2., except that the FD for an SA is written by IDMS.

An SA established to hold either the Systems Programming File or a User Library File will be formatted as follows:



2.4.1. Program Directory Descriptor

The format of the Program Directory Descriptor as it is written on the drum by AJAXF or OPUSF is as follows:

<u>CHARACTERS</u>	<u>CONTENT</u>
0-7	PGMDIRCT
8-11	Relative drum address of the first PD
12-15	Relative drum address of the first sector beyond the area reserved for PD entries
16-19	Relative drum address of the next free sector within the PD area that is available for a PD entry

2.4.2. Program Directory Format

A PD can describe either a source or object code program file, or an object code program segment file.

The format of a Program Directory that describes a source or object program file, as written on the drum by AJAXF or OPUSF, is as follows:

<u>CHARACTERS</u>	<u>CONTENT</u>
0-7	Name
8-15	Type of program: OBJ $\Delta\Delta\Delta$ for object code SORS $\Delta\Delta$ for source code
16-19	Relative drum address of the first sector of this program
20-23	Relative drum address of the first sector of the next program.

The format of a Program Directory that describes an object code program segment file, as written on the drum by AJAXF or OPUSF, is as follows:

<u>CHARACTERS</u>	<u>CONTENT</u>
0-7	Segment Name
8-15	SEGMENT Δ
16-19	Relative drum address of the first sector of this segment

3. PROGRAMMING PROCEDURES

The operation of IDMS is controlled by information punched in parameter cards. Parameter cards are required to describe the system configuration, the Systems Directory Descriptor, and the classes. One or more parameter cards that describe drum areas to be cleared or character filled by IDMS are optional.

3.1. CLEAR CARD

IDMS will clear, or set to a specified character, each area on the drum that is described by a clear card. The format of the clear card is as follows:

```
Column 1      13      19
A   CLEAR   P1, P2, P3, P4, P5, P6, P7
```

where

P₁, P₂, P₃ = Beginning drum address (unit, track, head).

P₄, P₅, P₆ = Ending drum address (unit, track, head); the drum will be cleared up to but not including this address.

P₇ = Fill character (octal, decimal, or alphanumeric), if the area is to be set to other than binary zeroes.

Sector addresses cannot be specified in the clear card; therefore, a track is the smallest drum area that can be cleared by IDMS.

Note: The use of clear cards is optional.

3.2. CONFIGURATION CARD

One configuration card, required to describe the system configuration to IDMS, is punched in the following format:

Column 1 13 19
 B CONFIG P₁, P₂, P₃, P₄

where

P₁ = 1050 or 1004, describing the type of online card reader, printer, and card punch.

P₂ = 80 or 90, describing the type of card used on the card reader and punch. Omit this parameter if P₁ = 1004.

P₃ = ROW or SERIAL, describing the type of card reader. Omit this parameter if P₁ = 1004.

P₄ = 3A or 3C or 6C, describing the type of UNISERVO online to the central processor. Omit this parameter if no tape units are available.

3.3. SDD CARD

One SDD card, required to describe the Systems Directory Descriptor to IDMS, is punched in the following format:

Column 1 13 19
 C SDD nn

where

nn = the number of classes, including Systems Areas, to be established on the drum.

3.4. CLASS CARD

A class card must be punched to describe each class to be established on the drum. The number of class cards must equal nn on the SDD card. Class cards are punched in the following format:

Column 1 13 19
 D n-n m-m,l-1,u,ttt,hh,ss

where

n-n = the number of files allowed for in this class. This number +1 is the number of sectors reserved for the FDD and FD's at the beginning of the class. If this class card describes a Systems area, n-n must be 1.

m-m = an eight-character alphanumeric name, bounded by apostrophes, by which this class will later be referenced. If this class is a Systems Area, the first character will be an asterisk.

l-1 = the length of this class in sectors

u = the starting unit number (0-7) of this class

ttt = the starting track position (0-95 for Fastrand I, 0-191 for Fastrand II) of this class

hh = the starting head position (0-63) of this class

ss = the starting sector (0-63) of this class

utthss represent the beginning drum address of this class. Add this address to 1-1 to obtain the address of the first sector beyond this class.

If utthss are omitted, this class will immediately follow the preceding class on the drum. If these parameters are omitted on the first class card, the first class will immediately follow the Systems Directory.

CLASS cards must be in ascending sequence according to beginning drum addresses.

3.5. SENTINAL CARD

The current run is terminated by a sentinal card, punched in the following format:

Column 13 19
 STOP DLL

where

DLL indicates that the Drum Locator/Loader is to be loaded, but no SDD or classes are to be established. Omit this parameter if the SDD and classes are to be established.

4. OPERATING PROCEDURES

4.1. INSTRUCTIONS FOR LOADING

Prior to loading IDMS and DLL, an object deck for each routine must be punched from a systems tape, using OPUS. This requires the following OPUS control cards:

SEQUENCE				LABEL		OPERATION			OPERANDS		
PAGE	LINE	INS		7	11	13	18	19	30		
1	3	4	5	6			PUNCH		IDMS		
							PUNCH		DLL		
							STOP				

The object decks produced by OPUS must be readied for loading by performing the following steps:

- (1) Remove the first card (R card) from the DLL object deck.
- (2) Place the DLL object deck behind the IDMS object deck.
- (3) Place a card loader in front of the IDMS object deck. The card loader should be assembled relative to 027400.
- (4) Place the IDMS parameter cards after the last card (T card) of the DLL object deck. If IDMS is used only to load DLL, only the CONFIG and STOP parameter cards are required.

To load, use the normal card loading procedures. IDMS will write the DLL on the drum, and immediately begin processing the parameter cards.

		IDMS													
TYPE		UU	SU	TTTT	HH	SS		UU	SU	TTTT	HH	SS	F	N	C
A	CLEAR FROM:	00	00	0000	01	00	TO:	00	00	0000	02	77			
A	CLEAR FROM:	00	00	0002	00	00	TO:	00	00	0002	01	77			
A	CLEAR FROM:	00	00	0003	00	00	TO:	00	00	0004	01	77			
A	CLEAR FROM:	00	00	0004	00	00	TO:	00	00	0004	01	77			
A	CLEAR FROM:	00	00	0005	00	00	TO:	00	00	0006	01	77			
B	CONFIG 1050,80,ROW,3C														20
C	SYSDIRCT AT:	00	00	0000	00	30									
	SD S FROM:	00	00	0000	00	31	TO:	00	00	0000	00	35			
D	FLEDIRCT AT:	00	00	0000	01	00								CLASS 0	
	FD S FROM:	00	00	0000	01	01	TO:	00	00	0000	01	05			
	RANGE FROM:	00	00	0000	01	00	TO:	00	00	0000	01	77			
D	FLEDIRCT AT:	00	00	0002	00	00								*AXXXXXX	
	FD S FROM:	00	00	0002	00	01	TO:	00	00	0002	00	01			
	RANGE FROM:	00	00	0002	00	00	TO:	00	00	0002	00	77			
D	FLEDIRCT AT:	00	00	0003	00	00								*ZXXXXXX	
	FD S FROM:	00	00	0003	00	01	TO:	00	00	0003	00	01			
	RANGE FROM:	00	00	0003	00	00	TO:	00	00	0003	77	77			
D	FLEDIRCT AT:	00	00	0004	00	00								*XXXXXX	
	FD S FROM:	00	00	0004	00	01	TO:	00	00	0004	00	01			
	RANGE FROM:	00	00	0004	00	00	TO:	00	00	0004	77	77			
D	FLEDIRCT AT:	00	00	0005	00	00								*\$XXXXXX	
	FD S FROM:	00	00	0005	00	01	TO:	00	00	0005	00	01			
	RANGE FROM:	00	00	0005	00	00	TO:	00	00	0005	77	77			

END IDMS

Table 4-1. IDMS List

4.2. IDMS LIST

IDMS provides a permanent record of the Drum Setup by printing a descriptive line as each card is successfully processed. Table 4-1 illustrates the list generated by IDMS. The following table describes the key to be used with table 4-1.

KEY	EXPLANATION
TYPE: A	Clear card.
B	Configuration card.
C	SDD card.
D	Class card.
Column F	Fill character designated on clear card.
Column N	Name of the class.

4.2.1. Clear Card Line

The clear card line shows the starting and ending drum address of the area to be cleared, and the designated fill character, if any, for each clear card.

4.2.2. Configuration Card Line

The configuration card line shows the parameters as they are punched on the configuration card.

4.2.3. SDD Card Lines

The first line for each SDD card shows the drum address of the SDD. The second line shows the starting and ending drum address of the SD's.

4.2.4. Class Card Lines

The first line for each class card shows the drum address of the FDD, and the class name. The second line shows the starting and ending drum address of the FD's. The third line shows the starting and ending drum addresses of the area reserved for the class, including the FDD and FD's.

4.2.5. Sentinel Card Line

The sentinel card line will contain:

END IDMS

indicating successful completion.

4.3. ERROR RECOVERY

The following table describes the display stops that may occur during the operation of IDMS, their causes, and action to be taken.

DISPLAY STOP	CAUSE	ACTION
060010	Number of class cards is less than nn on SDD card.	Restart.
060011	Unable to write in first or last sector of class because of invalid address or locked out area.	Restart.
060012	First and last addresses reference different units.	Restart.
060013	Non-unique names among classes.	Restart.
060014	The number of class cards is greater than nn on SDD card.	Restart.
060015	Card sequence error.	Restart.
060016	Drum address sequence error or overlay of a defined area. If card is initial class card, then referenced address is less than address of SDD entry plus nn on SDD card.	Restart.
060017	No configuration card.	Restart.
060020	No SDD card.	Restart.
060021	No class card(s).	Restart.
060023	Sequence error on DLL object deck.	Restart.
060024	Check-sum error on DLL object deck.	Restart.

UNIVAC

UNIVERSITY MICROFILMS INTERNATIONAL