

7 MAR 1979



Computer Automation

NAKED MINI. Division

18651 Von Karman, Irvine, California 92713 Tel 714 833 8830 TWX 910 595 1767

THIS RELEASE

PRELIMINARY

UNSUPPORTED

COMPUTER AUTOMATION
NM4 STANDARD DISK FILE FORMAT

93440-00B0

February 1979

TABLE OF CONTENTS

Section

1. INTRODUCTION
 - 1.1 FILE SYSTEM
 - 1.2 FILE NAMING SYSTEM
 - 1.3 DISK SPACE ALLOCATION AND ADDRESSING
 - 1.4 DISK TABLE STRUCTURE
2. SYSTEM BOOTSTRAP
3. VOLUME CONTROL BLOCK
4. F-LIST
 - FREE AU FILE
6. BAD FILE
7. ROOT DIRECTORY

THIS RELEASE

PRELIMINARY

UNSUPPORTED

COMPUTER AUTOMATION STANDARD DISK FORMAT

1. INTRODUCTION

This format is the standard for all NM4 software supported disks. It is modular enough to handle:

- Disks of different capacities, manufacturing and data formats
- Files of different organization, access ability, and data format

Though the Standard Format allows for a great deal of variance, some conventions have been chosen within this framework to allow easier implementation. The standard consists of:

- The definition of named file capability
- The definition of addressing capability on the disk
- Definitions of tables required to control the usage of the disk and files within it
- The definition of conventions whose adherence is required by various Computer Automation software packages

1.1 File System Overview

The NM4 disk file system provides access to named information (files) residing on magnetic disks. The file manager provides uniform access to all types of information. Thus to a user, a file may appear to have any arbitrary structure by interposing an intervening process. The file system, therefore, has only one type of file on mass storage, the randomly accessible array of binary information.

The objective of the file system is to have no bottlenecks in the I/O procedure so that a user may implement any type of service which he desires and the operating system does not provide. Toward that end, there are only two intrinsic types of files associated with a file name: data and directory. All data files are simply blocks of binary information. They are accessible randomly as an array of bytes and have no intrinsic type of data structure of their own other than that assumed by the program accessing them. Any structure is supplied by the process handling the file and not by the file system. A directory is simply a data file containing file names and pointers which is accessed only indirectly by the user.

The fundamental operations on a file are: create, delete, open, seek, (to new read/write position), read (quantity of bytes), write (quantity of bytes), close. There is no restriction on the formatting of information in a file.



1.2 File Naming System

A simple file name is a series of up to fourteen ASCII characters. A complex file-name (or path name) consists of a series of file-names. Each name but the last is the name of a directory file which may point to other files and directories. A name identifies one file in a given directory. A complex file name describes a path from a given directory through other directories to the final file name desired. A data file name need only be unique in its own directory to insure that any complex file name (or path name) is also unique.

Since a directory is a file like any other, the file system will in general have a tree-like hierarchical directory structure.

1.3 Disk Space Allocation and Addressing

Each data or directory file is allocated space on the disk according to its own requirements. To enable system control over files, several control blocks are maintained on the disk. These control blocks contain the information necessary to find all the files on the disk according to each file's name.

The basic units of space on the disk are sector, track and cylinder. To some extent, these are hardware defined by the device and or controller in use. This standard is not dependent on any sizes for these units.

Because of the range of disk capacities available, no single allocation size can be chosen. Hence, we define the Allocation Unit (AU) as the fundamental unit of allocation within any disk. The Volume Control Block describes the size of the AU for each disk. The AU is then used for addressing within the disk; i.e., if the disk space were considered a contiguous, ordered set of pieces of AU size, then the AU address is the ordinal of a piece. All disk addressing is done using an AU address. This does not affect how space within a file may be managed.

File space allocation is done in extents. Each extent is some number of AU's. Many extents can be allocated for each file. This allows allocation of a small extent initially with subsequent extent allocations available for large quantities of data. This standard in no way restricts the size of any extent or the allocation scheme used for finding the desired amount of space in the allocation table. The list of allocated extents is contained in the Extent Information Block for each file. The AU size must be chosen such that there are less than 65,537 AU's on a disk.

1.4 Disk Table Structure

The System BOOT and VCB are always located at fixed positions on the disk.



1.4 Disk Table Structure (cont.)

All other disk tables may reside anywhere on the disk since they are accessed by pointers. Typically the F-LIST, ROOT Directory and FREE Table are located near the center of the disk. The position and size of these tables is determined when the disk is labeled.

Figure 1.1 and 1.2 shows the typical overall NM4 Disk Structure

THIS RELEASE INTERNAL

PRELIMINARY

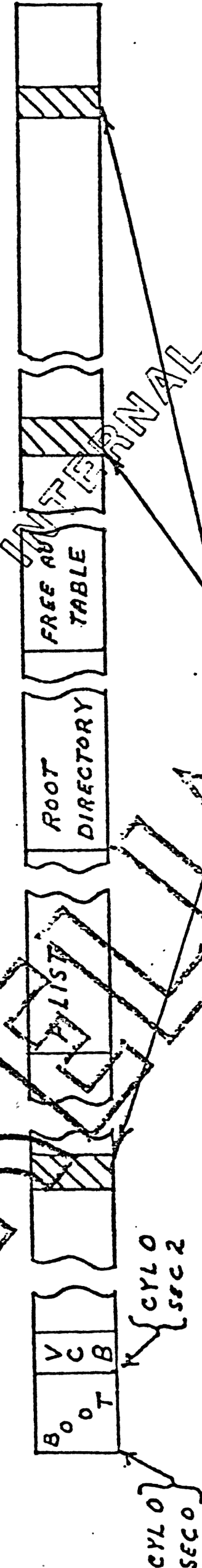
UNSUPPORTED

THIS RELEASE

PRO

TYPICAL NM4 DISK (Table and Control Block Distribution)

USE



INTERNAL TO C A I

USER FILES

FIGURE 1.1

C A I

UNSUPPORTED

MINI-MANARY

TYPICAL DISK TABLE LINKAGES

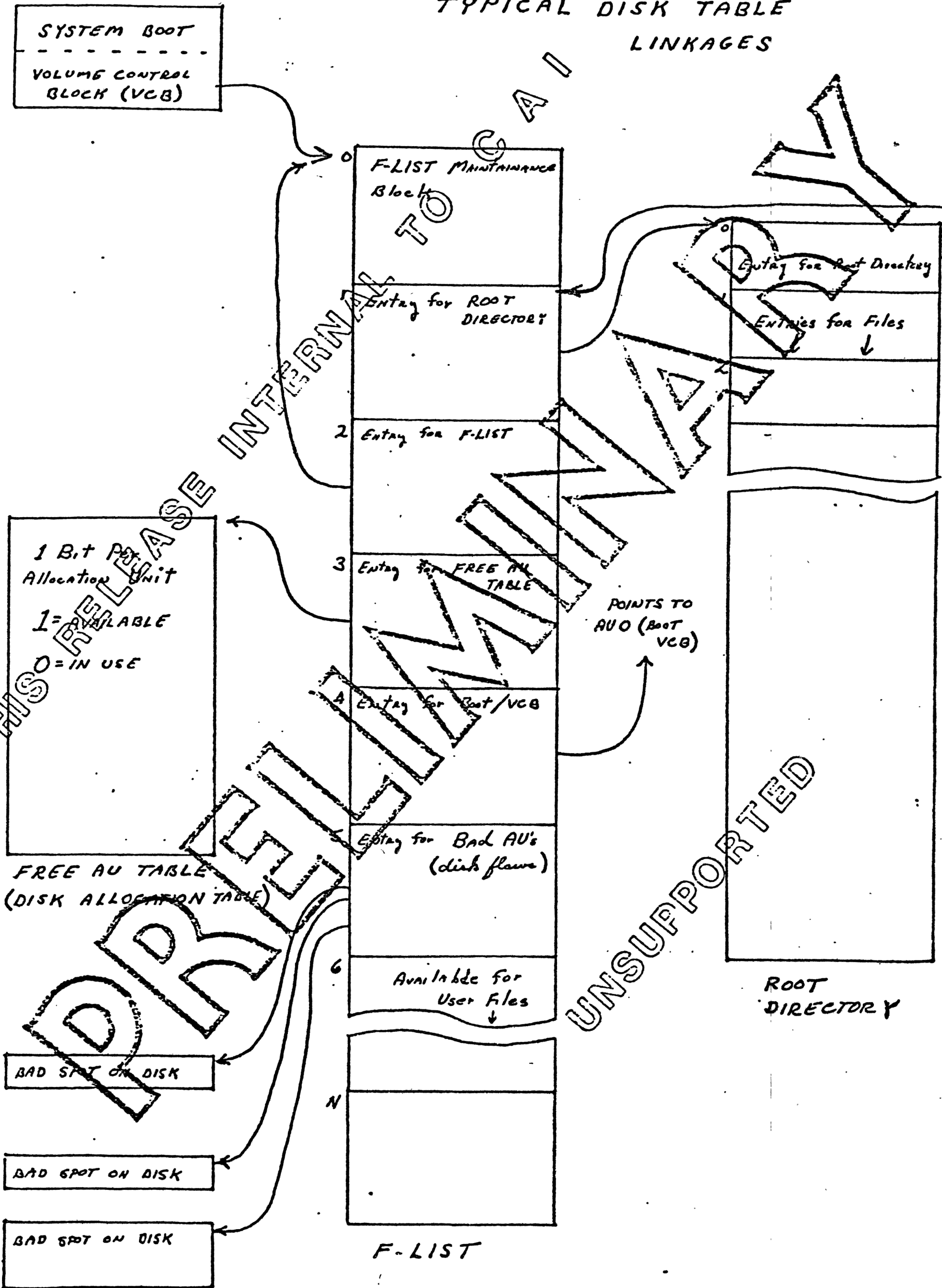


FIGURE 1.2

2. SYSTEM BOOTSTRAP

The system bootstrap resides in a fixed location on the disk. It must begin at cylinder 0, head 0 and sector 0. On medium and high capacity disks with a sector size of 256 words, the boot occupies 512 words (2 sectors). On the miniflexible and floppy disk drives, the sector size is 64 words. On these disks, the bootstrap is divided into two parts; part 1 is 64 words and part 2 is 512 words.

Figure 2.1 shows the position of the system boot on the various disks.

THIS RELEASE INTERNAL TO
PRELIMINARY
UNSUPPORTED



3. VOLUME CONTROL BLOCK

The Volume Control Block, VCB, is always found at cylinder 0, head 0 and sector 2. The VCB contains pertinent parameters for manipulating the disk, including the F-LIST pointer. All pointers to information on the disk are in allocation units (AU's).

The VCB contains integer and ASCII information. The integer information is in 16-bit words. The ASCII information appears as 7-bit right adjusted in an 8-bit byte, with upper bit = 0. All ASCII fields are left justified with trailing blanks.

Also contained in the VCB is information identifying the operating system on which the disk was created and disk format.

Figure 3.1 is a diagram of the VCB. Table 3.1 contains the physical attributes of the standard NM4 disk drives.

3.1 Volume Control Block Definition

WORDS 0 - 2	6 character System ID in ASCII of the operating system which created the disk.
WORD 3	Binary number of bytes per physical sector.
WORD 4	Binary number of sectors per track.
WORD 5	Binary number of tracks per cylinder. (Heads)
WORD 6	Total number of cylinders in binary.
WORD 7	Binary identifier for the disk format. For standard NM4 disks, this word contains the value :00C1.
WORD 8	Binary number of physical sectors per allocation unit.
WORD 9	Pointer to the start of the F-list. (AU number)
WORD 10	Flag word. No currently defined flags. This word is reserved.
WORD 11	Reserved for future expansion.
WORDS 12 - 14	6 character disk creation date in ASCII. The form is YYMMDD.
WORDS 15 - 17	6 character disk creation time in ASCII. The form is HHMMSS.
WORD 18	Reserved for future expansion.



3.1 Volume Control Block Definition (cont.)

WORDS 19 - 21	6 character date in ASCII representing the last date the disk was modified.
WORDS 22 - 24	6 character time in ASCII representing the last time the disk was modified.
WORD 25	Reserved for future expansion.
WORDS 26 - 28	6 character date in ASCII representing the last back-up of the disk.
WORD 29 - 31	6 character time in ASCII representing the last back-up of the disk.
WORDS 32 - 55	Reserved for future expansion.
WORD 56	Disk title length in bytes.
WORDS 57 - 255	Disk title text in ASCII. The first 14 characters are the volume ID.

THIS RELEASE

PRELIMINARY

UNSUPPORTED

VOLUME CONTROL BLOCK (VCB)

WORD

WORD

0	
1	SYSTEM ID
2	
3	BYTES PER SECTOR
4	SECTORS PER TRACK
5	TRACKS PER CYLINDER
6	CYLINDERS PER DISK
7	SPECIAL ID
8	SECTORS PER AU
9	F-LIST POINTER
10	FLAG WORD
11	-RESERVED-
12	
13	CREATION DATE
14	
15	
16	CREATION TIME
17	
18	-RESERVED-
19	
20	MODIFICATION DATE
21	
22	MODIFICATION TIME
23	
24	
25	-RESERVED-
26	
27	BACKUP DATE
28	
29	
30	BACKUP TIME
31	

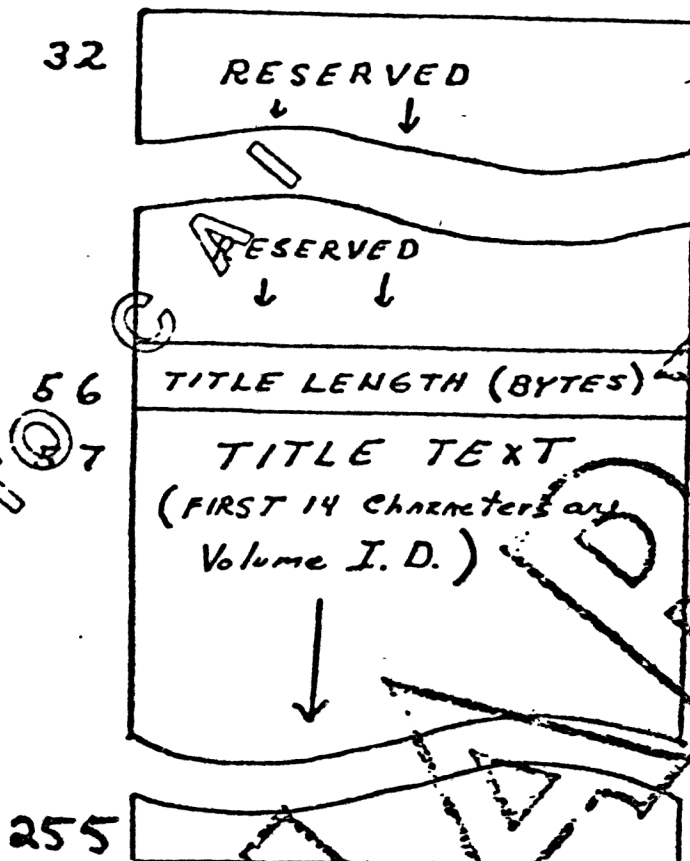
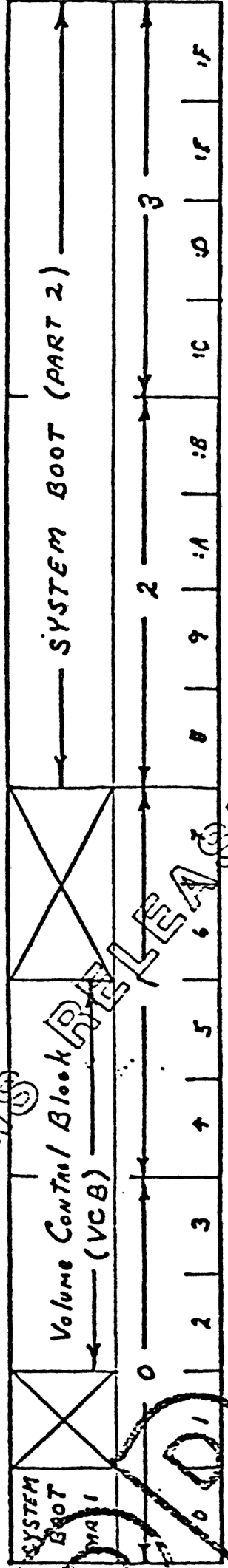


FIGURE 3.1

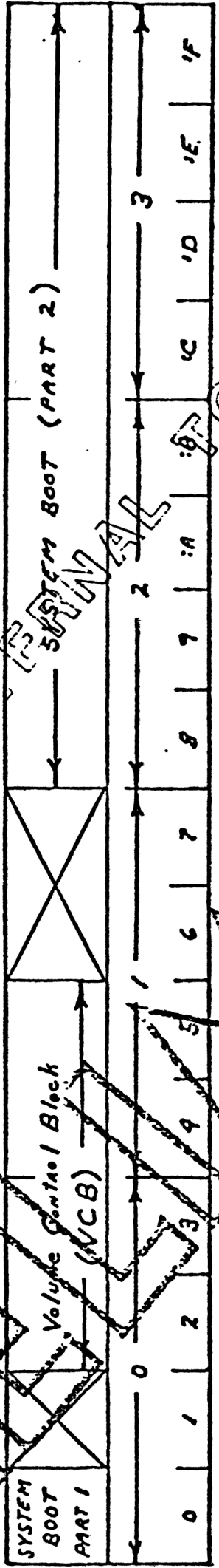
FIGURE 2.1 SYSTEM BOOTSTRAP AND VOLUME BLOCK

Mini-flexible



Allocation Unit #
Physical Sector #

Floppy Diskette



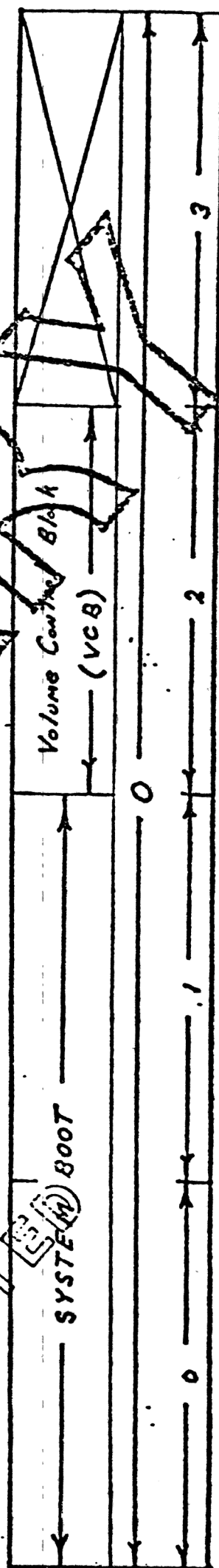
Allocation Unit #
Physical Sector #

Mod. Cap. Disk



Allocation Unit #
Physical Sector #

High Cap. Disk



Allocation Unit #
Physical Sector #

128 Bytes
512 Bytes

TABLE 3.1

NM4 STANDARD DISK PHYSICAL ATTRIBUTES

Disk Type	MF	SF	DF	MC*	HC	HC	HC	HC
Disk Capacity				2.5mb	5mb	40mb	80mb	150mb
Bytes/Sector	128	128	256	512	512	512	512	512
Sectors/Track	26	26	26	12	12	32	32	32
Tracks/Cylinder(heads)	1	1	2	2	2	5	5	19
Cylinders/Disk	35	74	77	203	406	823	823	411
Sectors/AU	4	4	2	1	1	8	8	8
AU's/Disk	280	580	1001	4872	9754	16,460	16,460	31,236
Avialable Bytes/Disk	140kb	250kb	1mb	2.5mb	5mb	33mb	66mb	128mb
								256mb

*NOTE:

1. The 2.5mb medium capacity disk describes one of the two disks contained in a 5mb drive.
2. The 5mb medium capacity disk describes one of the two disks contained in a 10mb drive.

LEGEND

- kb = kilo-bytes
mb = mega-bytes
MF = Miniflexible Disk
SF = Single Sided Floppy
DF = Double Sided Floppy
MC = Medium Capacity Disk
HC = High Capacity Disk

37
462
1540
2002



4. F-LIST

The F-LIST is a file containing information on all files on the disk. The Operating System gains access to the F-LIST through a pointer in the VCB. F-LIST consists of a series of fixed length entries. The length of the F-LIST entries is 256 words (512 bytes).

Each entry in the F-LIST is referred to by a index number called the F-number (FNO). Several of the F-LIST entries at the beginning of the F-LIST are reserved for disk tables (special files). The remaining F-LIST entries are available for file creation. The following F-LIST entries are reserved:

<u>FNO</u>	<u>NAME</u>	<u>USAGE</u>
0	None	F-LIST Maintenance Block
1	ROOT	References Root Directory
2	F-LIST	References F-LIST itself
3	FREE	References FREE AU table
4	BOOT	References BOOT/VCB area
5	BAD	References Bad spots on disk (flaws)

4.1 F-LIST Entries

F-LIST entries are fixed length (256 words) blocks and are composed of two parts; the header and the body. The format of the body of the F-LIST entry depends on the F-LIST entry TYPE. The entry type is contained in the header. The basic format is shown below in Figure 4.1.

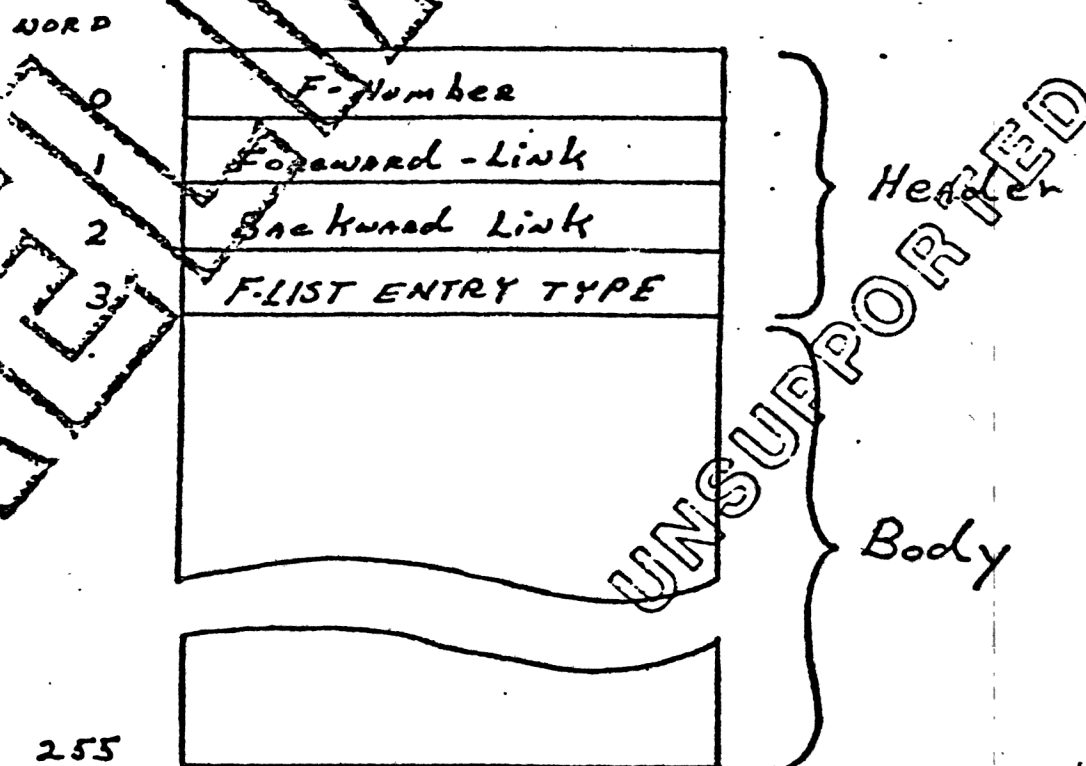


FIGURE 4.1

4.1.1 F-LIST Entry Header Definition

WORD 0 - F-NUMBER (FNO)

The FNO is the index to an F-LIST entry. For a very large file, it might be necessary to use more than F-LIST entry to describe all file extents. When several F-LIST entries are used to describe a file, they all contain the FNO of the head block. The FNO only has meaning on an allocated block.



4.1.1 F-LIST Entry Header Definition

WORD 1 - FORWARD LINK

If a file requires more than one F-LIST entry for its description information this word points to the next F-LIST entry representing this file. That is, F-LIST entries can be linked together to form a linked list (chain).

WORD 2 - BACKWARD LINK

If an F-LIST entry is part of a chain of F-LIST entry blocks, this word points to the previous block on the chain.

WORD 3 - F-LIST ENTRY TYPE

This word denotes the F-LIST entry type and subsequent body format. Valid TYPES are:

VALUE

0
1
2
3
4-10

TYPE

Available F-LIST entry
FILE/EXTENDER
TITLE
MAINTENANCE
Reserved

4.1.2 F-LIST Entry Body Definition

The F-LIST entry body format is different for each F-LIST entry TYPE.

4.1.2.1 F-LIST Entry TYPE 0

This is an available entry, the body contains no defined information.

4.1.2.2 F-LIST Entry TYPE 1

This is either a FILE entry or an EXTENDER entry. These two are identical except for the file information. The FILE block is always located at the FNO of the file. Any EXTENDER blocks are linked to the FILE block and contain the FNO of the FILE block.

The format of the FILE/EXTENDER is described in Figure 4.2.

WORDS 0-3 F-LIST entry header

WORD 4 Total number of extent descriptors to follow.

WORDS 5 thru $2n+5$ contain 2 word extent descriptors (n = number contained in word 4). Each descriptor consists of the base AU number and the length (in AU's) of the extent.

COMPUTER AUTOMATION STANDARD DISK FORMAT

4.1.2.2 F-LIST Entry TYPE 1 (cont.)

WORDS $2n+5$ thru $2n+53$ contain the rest of the F-LIST body. The definition follows (relative 0):

WORD 0	User identifier
WORD 1	Group identifier
WORD 2	Execution identifier
WORD 3	Title number
WORD 4	Number of directory links (NAME)
WORDS 5-6	File length in bytes (2 words)
WORD 7	File creation identifier
WORDS 8-10	6 character ASCII file creation date (YYMMDD)
WORDS 11-13	6 character ASCII file creation time (HHMMSS)
WORD 14	Modification identifier
WORDS 15-17	6 character ASCII date of last modification (YYMMDD)
WORDS 18-20	6 character ASCII time of last modification (HHMMSS)
WORD 21	Last access identifier
WORDS 22-24	6 character ASCII date of last access (YYMMDD)
WORDS 25-27	6 character ASCII time of last access (HHMMSS)
WORD 28	Secondary extent size in AU's
WORD 29	Record length in bytes
WORDS 30-48	Reserved for future expansion



4.1.2.3 F-LIST Entry TYPE 2

This is a TITLE entry. The TITLE F-LIST entry allows text information to be attached directly to a file independently of file name. The format of the TITLE entry is described in Figure 4.3.

WORD 0-3 F-LIST entry header
WORD 4 Text length in bytes
WORDS 5-255 TEXT in ASCII (7 bit, parity bit off)

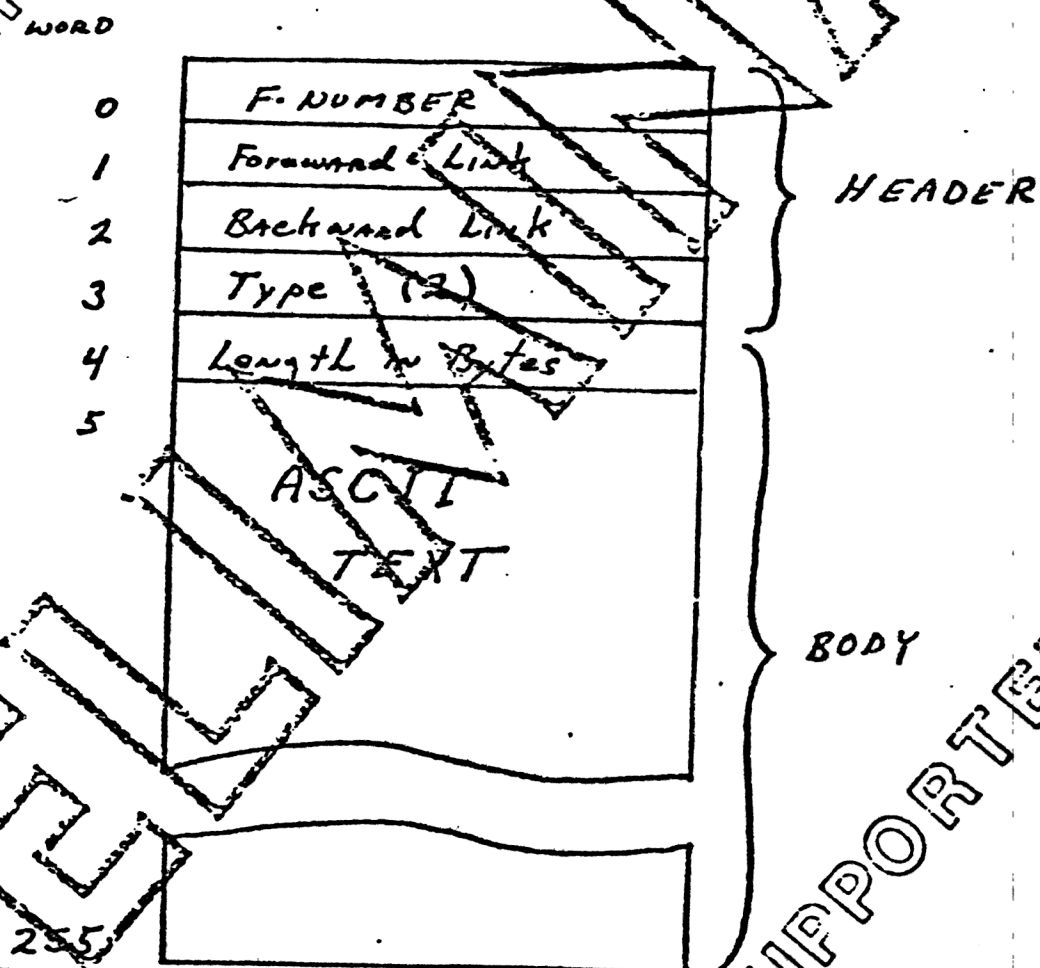


Figure 4.3

14

UNSUPPORTED

FIGURE 4.2



4.1.2.4 F-LIST Entry TYPE 3.

This is an F-LIST maintenance entry. The only F-LIST entry which can be TYPE 3 is the F-LIST maintenance block. This is always located at FNO=0. This block contains a bit map of free F-LIST entries in the F-LIST.

This block is intended for fast allocation of F-LIST entries and the structure is described in Figure 4.4.

WORDS 0-3 F-LIST entry header

WORDS 4-255 F-LIST entry allocation table bit map

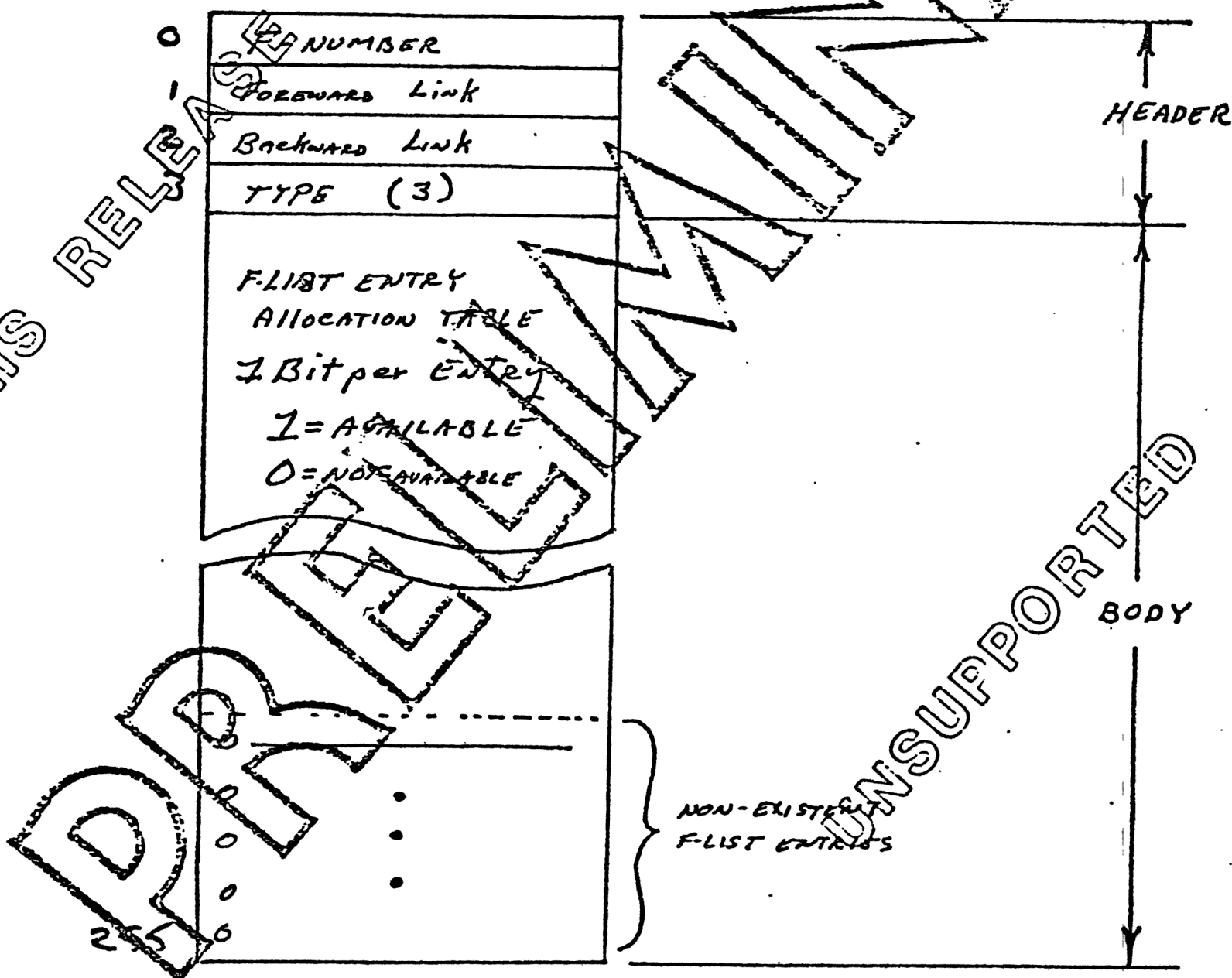


FIGURE 4.4



5. FREE AU FILE

The FREE AU file contains information which defines usage of disk allocation units (AU's). Whenever a file is created or extended, the correct bits must be cleared in the FREE file to reserve the disk space. When a file is deleted and disk space is released, the appropriate bits must be set to indicate that the space may be reused.

The FREE file is in actuality a bit table. Each AU on the disk is represented as a single bit. A 1 indicates an available AU. A 0 indicates an in-use or unavailable AU. See Figure 5.1.

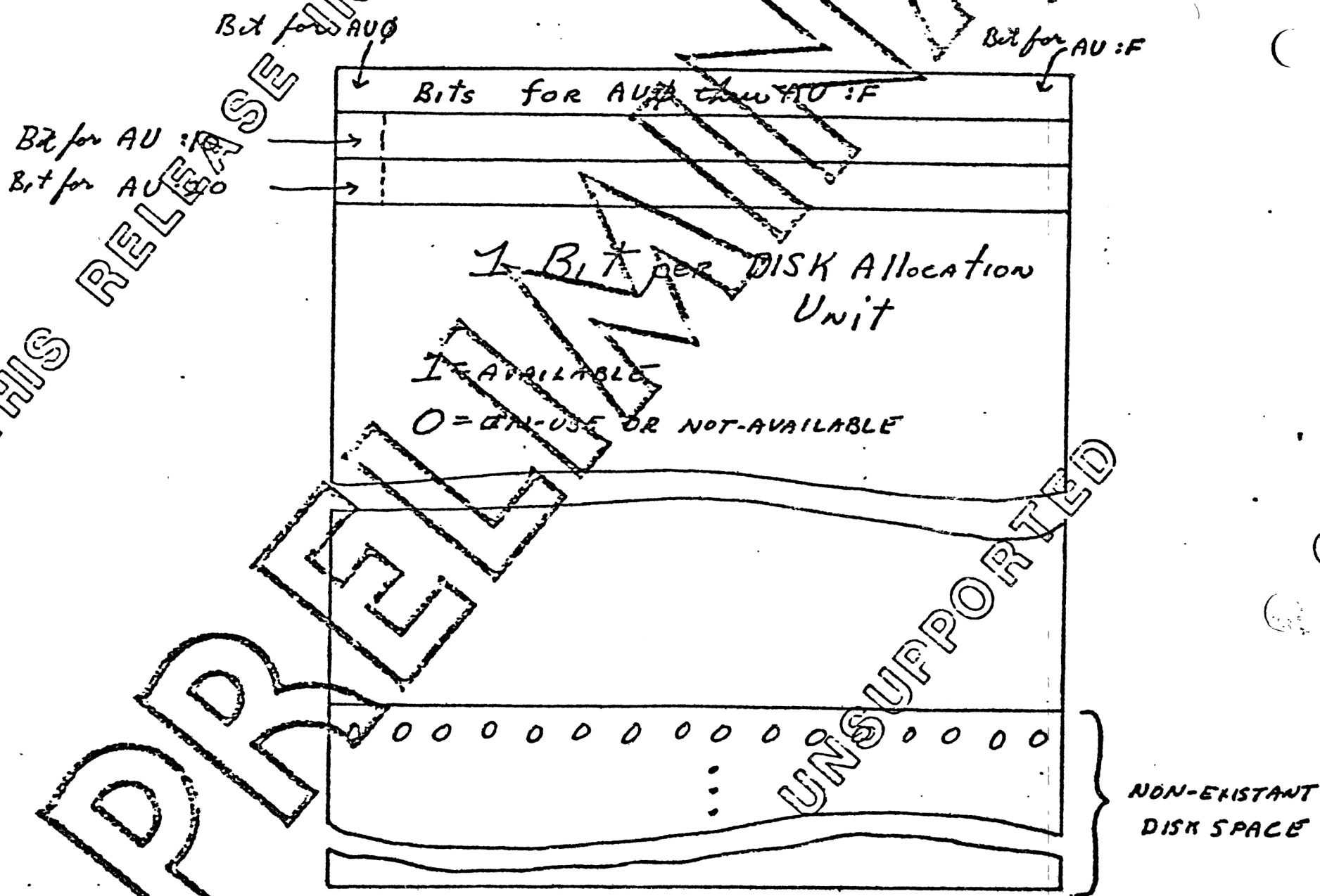


FIGURE 5.1

NOTE: The FREE file is always referenced by FN0=3 of the F-LIST.



6. BAD FILE

The BAD file is generated when a disk is labeled. It is a file which contains all AU's on the disk which have been found unusable. Bad AU's are typically caused by flaws on the disk surface.

The BAD file is referenced by FN0=5 of the F-LIST.

The F-LIST entry for the BAD file is a TYPE 1 F-LIST entry and therefore looks like a normal file. The extent descriptors within the F-LIST entry (FN0=5) point to the unusable AU's.

THIS RELEASE INTERVIEW

PRELIMINARY

UNSUPPORTED



COMPUTER AUTOMATION STANDARD DISK FORMAT

7. ROOT DIRECTORY

The Root Directory contains information necessary to find named files on the disk. The directory is a table of entries which contain FNO-filename correspondences.

The Root Directory is always referenced by FNO=1.

Directory entries are 8 words long. See Figure 7.1.

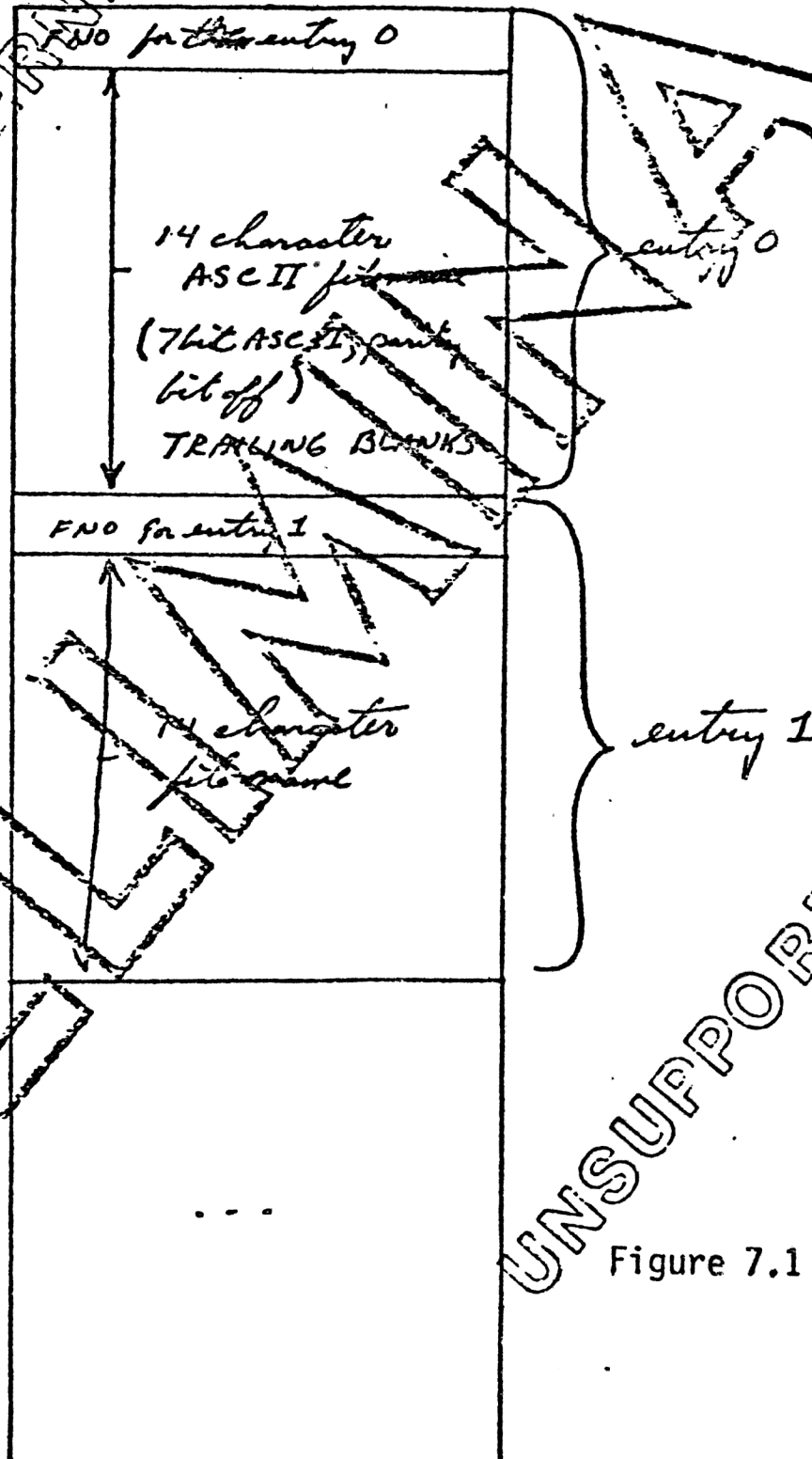


Figure 7.1

NOTE: An empty directory entry is specified if the referenced FNO is 0.