RADIO CORPORATION OF AMERICA | ELECTRONIC DATA PROCESSING | CAMDEN, N.J.

SYSTEMS STANDARDS

RCA
301
EDP

ELECTRONIC DATA PROCESSING SYSTEMS A BROAD RANGE OF SPEEDS AND CAPABILITIES TO MATCH MANY USER REQUIREMENTS

RCA 301

SYSTEM STANDARDS

93-29-000

June 1965

The information contained herein is subject to change without notice. Revisions will be provided to advise of such additions and/or corrections.

This manual has been reprinted to include all previous revisions released through June, 1965.

FOREWORD

This manual specifies the RCA 301 general system, program, and data standards for the following RCA 301 Standard Library Systems:

Magnetic Tape Library*
Card/Paper Tape Libraries
Data Record File Library
Data Disc File Library
Assembly Systems
Narrator Library (COBOL)
FORTRAN Library
Scientific Interpreter Library

Compliance with the standards described herein is an essential requirement for ensuring system compatibility between the RCA 301 Library Systems listed above and related User's programs. Additional standards (where necessary) for the proper operation of a specific library system are included in the appropriate publication(s) for each system.

*Includes Random Access Computer Equipment Standards (Chapter XI).

TABLE OF CONTENTS

I—CODING AND PUNCHING STANDARDS FOR ALL RCA 301 CHARACTERS

Due to the nature of RCA 301 coding, standards have been established to facilitate the coding and card-punching of 301 characters. The standards are defined in the chart below.

CHARACTER DESCRIPTION	CODING STANDARD	KEY-PUNCHING
Zero	0	0
One	1	1
Two	2	2
Three	3	3
Four	4	4
Five	5	5
Six	6	6
Seven	7	7
Eight	8	8
Nine	9	9
Space or Underline	$\underline{S}$p	No Punch
Number	#	3/8
At the rate of	@	4/8
Open Parenthesis	(	5/8
Closed Parenthesis	)	6/8
Ampersand	$\textcircled{\&}$	Y(Zone Punch)
+ 0	⊕	Y(Zone Punch)/0
A	A	A
B	B	B
C	C	C
D	D	D
E	E	E
F	F	F
G	G	G
H	H	H
I	I	I
Plus	+	B/8
Period	$\boxed{.}$	C/8
Semicolon	;	D/8

CHARACTER DESCRIPTION	CODING STANDARD	KEY-PUNCHING
Apostrophe	$\textcircled{'}$	F/8
Minus	⊖	X (Zone Punch)
J	J	J
K	K	K
L	L	L
M	M	M
N	N	N
O	Ø	O
P	P	P
Q	Q	Q
R	R	R
End of Info.	E/I	K/8
Dollar	$	L/8
Asterisk	*	M/8
End Data	E/D	N/8
End File	E/F	Ø/8
Quotes	"	X(Zone Punch)/0
Slash	/	0/1
S	S	S
T	T	T
U	U	U
V	V	V
W	W	W
X	X	X
Y	Y	Y
Z	Z	Z
End Block	E/B	S/8
Comma	,	T/8
Percent	%	U/8
Item Separator	•	V/8
Equal	=	W/8
Colon	$\boxed{:}$	E/8

NOTE: The circles and boxes around the characters indicated in the coding standard column are to facilitate the differentiation between similar characters when hand writing the coding. These boxes and circles are not to be utilized when coding is typed for documentation.

II—STANDARD HSM LAYOUT

Computer Required Standard Areas - The locations from 0000 to 0225 encompass all of the standard areas used by the 301 computer.

Service Routine System Required Standard Areas - Special areas of HSM have been designated for use by the service routines and are also available for production use. An area of 160 locations has been allocated for use as a print area. It can also be used as a message read-in area or a work area.

Production programs may be coded starting at HSM address 2000 to allow sufficient memory space for service routine storage. A summary of all the standard HSM locations is below:

0000 - 0099	Sum Table
0100 - 0199	Difference Table
0200 - 0200	Storage Area for the tape trunk of the Program Library Tape
0202 - 0205	Card Punch - Temporary storage of address
0206 - 0209	Arithmetic - Temporary storage of address
0212 - 0215	STA
0216 - 0219	STP
0222 - 0225	Store P During Repeat
0250 - 0720	Insertion or Load System
0256 - 0257	Reading Device Indicator and Type Insertion Indicator
0258 - 0259	HSM Size Indicator
0600 - 0759	Work Area and Standard Print Area
0760 - 0767	Standard Date
0770 - 0779	Standard Exit
0780 - 0834	Multiply and Divide Parameters
0835 - 0852	Standard Tape Table
** 0853 - 0857	Last DDF Control Panel, Unit and Track selected
0858 - 0879	Standard Overlay Area
0880 - 1999	Storage Area for Debugging Routines
2000 - X899	Coding area
*X900 - X977	Print Table

For additional HSM allocations see specific standards for Cards, Paper Tape, Magnetic Tape, Data Record File, and Data Disc File.

*X = Z, I, or 9 for 40K, 20K, 10K HSM respectively.

** When using 3488 software, HSM locations 0853-0856 must contain the address of the System Control Card.

III — PRINT TABLE

The Standard Print Table is required to be in HSM location 9900-9977 for a 10K HSM configuration, I900-I977 for 20K HSM or Z900-Z977 for a 40K HSM.

For all routines which require printing HSM or data from an I/O medium, the Standard Print Table should be converted into a Print-all Table so that all possible characters will be printed.

The standard print table appears as follows:

HSM LOCATION	SYMBOL ON PRINT WHEEL	RCA 301 CHARACTER	CONTENTS (OCTAL)	HSM LOCATION	SYMBOL ON PRINT WHEEL	RCA 301 CHARACTER	CONTENTS (OCTAL)
*X900	-(minus)	-(minus)	40	X939			*
01	+	+	32	40	A	A	21
02	space		17	41	B	B	22
03	0	0	00	42	C	C	23
04	1	1	01	43	D	D	24
05	2	2	02	44	E	E	25
06	3	3	03	45	F	F	26
07	4	4	04	46	G	G	27
08			*	47	H	H	30
09			*	48			*
X910	5	5	05	49			*
11	6	6	06	X950	I	I	31
12	7	7	07	51	J	J	41
13	8	8	10	52	K	K	42
14	9	9	11	53	L	L	43
15	,(comma)	,(comma)	73	54	M	M	44
16	.(period)	.(period)	33	55	N	N	45
17	@	@	14	56	O	O	46
18			*	57	P	P	47
19			*	58			*
X920	%	%	74	59			*
21	:	:	35	X960	Q	Q	50
22	#	#	13	61	R	R	51
23	$	$	53	62	S	S	62
24	)	)	16	63	T	T	63
25	"(quote)	"(quote)	60	X964	U	U	64
26	subscript$_{10}$		17	65	V	V	65
27	(	(	15	66	W	W	66
28			*	67	X	X	67
29			*	68			*
X930	]		17	69			*
31	;	;	34	X970	Y	Y	70
X932	>		17	71	Z	Z	71
33	÷		17	72	C_R		17
34	↑		17	73	'(apostrophe)	'(apostrophe)	36
35	[		17	74	*	*	54
36	<		17	75	&	&	20
37	=	=	76	76	/	/	61
38			*	77	⌑		17

*X = Z, I or 9 for 40K, 20K or 10K HSM, respectively.

PRINT SYMBOLS FOR RCA 301 CHARACTERS (PRINT-ALL TABLE)

Certain RCA 301 characters will not print as RCA 301 symbols, so a representative code is used to designate these characters. When it is desired to display these codes the appropriate RCA 301 character should be inserted into the print table. The characters and their representative code symbols are as follows:

PRINT SYMBOL	RCA 301 CHARACTER	BINARY REPRESENTATIVE	MEANING
⊐		001010	Underline (Space)
⊏	EI	101010	End of Information
>	ED	101101	End of Data
<	EF	101110	End of File
÷	EB	111010	End of Block
↑	•	111101	Item Separator
Space	$(17)_8$	001111	Not Designated
C_R	$(37)_8$	011111	Not Designated
Subscript$_{10}$	$(57)_8$	101111	Not Designated
⌑	$(77)_8$	111111	Not Designated

All RCA 301 characters not listed above are printed as standard RCA 301 symbols.

IV — 301 AND 501 TRANSLATION TABLES

Alphabetic, numerics, and the following characters:

( ) " : $ % ; ⊖ * / , # E/F E/D • ' &.

will be translated from their RCA 501 Configurations to corresponding RCA 301 Configurations and vice versa.

The following table gives the translation for all remaining characters from RCA 501 Configurations to RCA 301 Configurations and vice versa.

RCA 501		RCA 301	
SYMBOL	CONFIGURATION	SYMBOL	CONFIGURATION
Blank	000000	@	001100
Space	000001	sp	001010
Cross	000010	+	011010
Carriage Shift	001111	$(17)_8$	001111
Page Change	010000	$(37)_8$	011111
Line Shift	010001	$(57)_8$	101111
Carriage Normal	011111	=	111110
End Message	111101	EI	101010
Start Message	111110	EB	111010
Delete	1111111	Unused	111111

The RCA 501 Character $(77)_8$ will be translated to RCA 301 Character $(77)_8$. This RCA 301 Character can be used for sensing or writing to a Record File or tape on a Tape Adaptor. However, it cannot be written to a tape on the High Data group. Octal characters $(17)_8$, $(57)_8$, and $(77)_8$ cannot be introduced into the RCA 301 System via cards or $(17)_8$, $(37)_8$, $(57)_8$, and $(77)_8$ via paper tape; and therefore, must be generated if desired for sensing purposes.

A. 501 TO 301 TRANSLATION TABLE (501 MAGNETIC TAPE INPUT)

The following table shows the construction of a translate table in 301 HSM used to translate RCA 501 characters (introduced via a magnetic tape) to the corresponding RCA 301 Character Configurations. The chart is as the table would appear on a coding sheet where XX is arbitrary.

	00	01	02	03	04	05	06	07	08	09
XX00	@	_	+	(	)	"	:	$	Free	Free
10	%	;	&	'	⊖	*	.	$(17)_8$	Free	Free
20	$(37)_8$	$(57)_8$	/	0	1	2	3	4	Free	Free
30	5	6	7	8	9	,	#	=	Free	Free
40	A	B	C	D	E	F	G	H	Free	Free
50	I	J	K	L	M	N	Ɵ	P	Free	Free
60	Q	R	S	T	U	V	W	X	Free	Free
70	Y	Z	E/F	E/D	•	EI	EB	$(77)_8$	Free	Free

B. 501 TO 301 TRANSLATION TABLE (501 PAPER TAPE INPUT)

The following table shows the construction of a translate table in HSM used to translate RCA 501 Characters (introduced via 501 paper tape) to the corresponding RCA 301 Character Configurations. The chart is as the table would appear on a coding sheet where XX is arbitrary.

	00	01	02	03	04	05	06	07	08	09
XX00	$77_{(8)}$	E/B	E/I	•	E/D	E/F	Z	Y	Free	Free
10	X	W	V	U	T	S	R	Q	Free	Free
20	P	~~0~~	N	M	L	K	J	I	Free	Free
30	H	G	F	E	D	C	B	A	Free	Free
40	=	#	,	9	8	7	6	5	Free	Free
50	4	3	2	1	0	/	$(57)_8$	$(37)_8$	Free	Free
60	$(17)_8$	.	*	–	'	&	;	%	Free	Free
70	$	:	"	)	(	+	_	@	Free	Free

C. 301 TO 501 TRANSLATION TABLE

The following table shows the construction of a translate table in HSM used to translate RCA 301 Characters to the corresponding RCA 501 Character Configurations. The chart is as the table would appear on a coding sheet where XX is arbitrary. The symbols internal to the table contain the proper octal configurations to produce the correct translation to 501 characters.

	00	01	02	03	04	05	06	07	08	09
XX00	C	D	E	F	G	H	I	+	Free	Free
10	.	;	1	'	0	3	4	$(17)_8$	Free	Free
20	_	–	J	K	L	M	N	~~0~~	Free	Free
30	P	Q	2	)	9	6	#	&	Free	Free
40	@	R	E/I	$	*	E/D	E/F	$(57)_8$	Free	Free
50	"	/	•	7	(	,	E/B	A	Free	Free
60	5	B	S	T	U	V	W	X	Free	Free
70	Y	Z	=	:	8	%	$(37)_8$	$(77)_8$	Free	Free

V — GENERAL DATA STANDARDS

A. LEGEND FOR DATA SYMBOLS

E/B represents the RCA 301 End Block Symbol.

E/I represents the RCA 301 End Information Symbol.

E/F represents the RCA 301 End File Symbol.

E/D represents the RCA 301 End Data Symbol.

BL represents the Beginning Label.

EL represents the Ending Label.

BLK represents a Block of Data or One Cell which contains data.

B. CHARACTER RESTRICTIONS

The following control characters are restricted from being an integral part of a data record.

$(17)_8$

End of Information (E/I)

End of Data (E/D)

End of File (E/F)

End of Block (E/B)

$(57)_8$

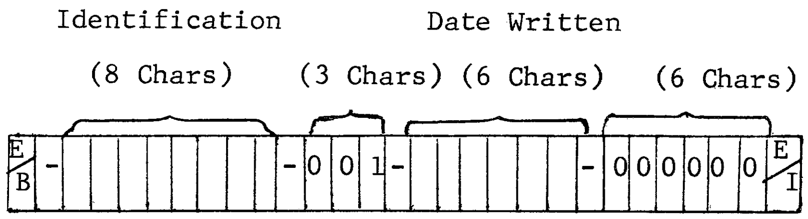
$(77)_8$

VI — CARD STANDARDS

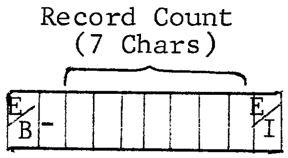
A. ORGANIZATION OF DATA

1. Standard Label Formats

   a. Beginning Label (29 characters)



   b. Ending Label (10 characters)



2. Card Label Conventions

   a. The length of user-defined labels must not exceed 80 characters including the E/B _ (space) and E/I symbols. This means that a user may modify the labels, if it is so desired.

   b. Standard Labels may be indicated.

   c. Beginning and Ending Labels may be omitted.

   d. A Beginning Label may be specified without an Ending Label, however, an Ending Label may not be specified without a Beginning Label.

3. Standards for Data on Cards

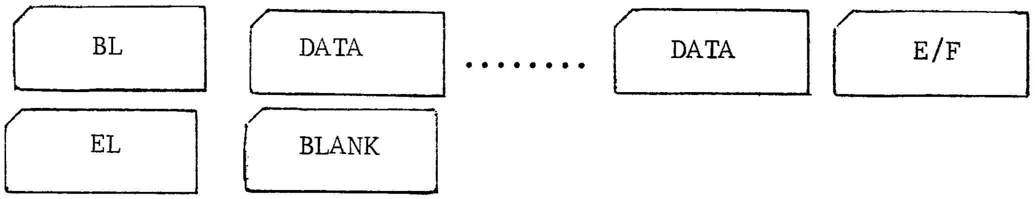
   a. Data records must be a fixed 80 characters in length.

   b. Data records are not terminated by an E/I symbol, nor any other symbol. This rule does not apply to labels.

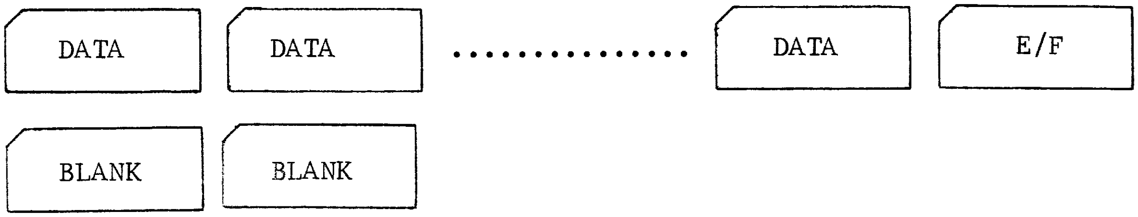
   c. Data records may not be batched.

4. Format of Data on Cards

   a. With Labels



b. Without Labels



NOTES: 1. An E/F card signifies the end of the card file. The E/F symbol must appear in the first column, and the remainder of the card must be blank.

2. A card file with labels must be followed by at least one blank card; a card file without labels must be followed by at least two blank cards.

B. PROGRAM STANDARDS

1. Floatable Programs - must be coded with respect to HSM location 0000.

2. Program Segments - the first instruction of each program segment must contain the letters PS as the first two characters and the MSC of the segment must be specified in the A Address. The B Address will contain information to identify the program segment.

3. Standard Entrance Location - the second instruction of the first program segment is assumed as the standard entrance location of each program.

4. Standard Exit Location - all programs (with the exception of a subroutine) must exit to HSM location 0770 at the end of the program in lieu of stopping within the program. The CARD LOADER will pre-store a transfer to 0260 in this location.

5. Standard Date - the date must always be put in HSM at locations 0760 to 0767. (e.g., 07/27/61)

6. The CARD LOADER is placed in HSM from 0250 to 0599 by the BOOTSTRAP and operates from these locations for all processors. Columns 7 and 8 of the LOADER BOOTSTRAP contain the HSM size indicator. This indicator will be placed in the CARD LOADER at 0258-0259 by the LOADER BOOTSTRAP. This indicator is set as follows:

99	for	10K
I9	for	20K
Z9	for	40K

C. ORGANIZATION OF PROGRAMS STORED ON CARDS

The program instructions are punched in eighty column cards in the following format:

CARD COLUMN	CONTENTS
1	The number of instructions in the message (maximum of 6).
2 to 5	The starting HSM location of the first instruction on the card.
6	The "float indicator" for the A Address of the first instruction on the card.
7	The "float indicator" for the B Address of the first instruction.
8 to 17	The first instruction (OP N A B).
18 etc.	The "float indicator" for the A Address of the second instruction for condensed decks. The first character of the tag field for pseudo/object (uncondensed) decks.

Note: The last three columns of the card (78-80) contain the Program Identification for condensed decks. Columns 72-73 contain the Program Identification for pseudo/object decks.

D. STANDARD HALTS

N Character	Meaning
0	Normal Halts
1	Error Halts
4	Equipment Error Halts

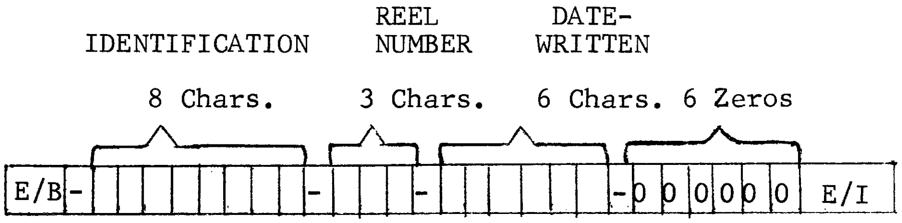
The A Address contains numeric characters to further clarify the halt.

VII — PAPER TAPE STANDARDS

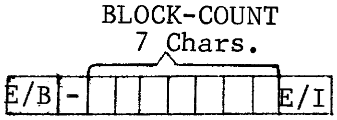
A. ORGANIZATION OF DATA

1. Standard Label Formats

   a. Beginning Label (29 Characters)



   b. Ending Label (10 Characters)



2. Tape Label Conventions

   a. Standard Labels may be indicated.

   b. Beginning and Ending Labels may be omitted.

   c. A Beginning Label may be specified without an Ending Label. However, an Ending Label may not be specified without a Beginning Label.

   d. The User may specify the format of the entire label(s) in which none of the standard items appear (own coding).

   e. Additional information may appear in the label following all standard items indicated above, except the E/I symbol. The E/I symbol must always be the last character of the label.

   Note: The BLOCK-COUNT of a file is the total number of data blocks on a tape, and does not include Beginning or Ending Label Blocks.

3. Standards for Data on Paper Tape

   a. All data records are terminated by an E/I symbol.

   b. A batch may contain:

      1. A variable number of fixed length records,

      2. A fixed number of fixed length records,

      3. A variable number of variable length records,

      4. A fixed number of variable length records.

   c. A batch is always terminated by an E/F symbol. The E/F must be the last character in the batch. No batch on tape, other than the labels may begin with "EB_".

d. When the input to a run is composed of <u>all</u> single records, no E/F symbol appears after the E/I symbol in any of the records. However, when the input to a run is batched, each batch contains an E/F symbol as the last character of the batch even though some batches may consist of single records.

e. When labels are not present:

1. Each reel of a file begins with a block consisting of a single E/F symbol.

2. Initial and intermediate reels of a multi-reel file are terminated by a block consisting of a single E/D symbol.

3. The final reel of a multi-reel file, or a single reel file, is terminated by two successive blocks, the first of which consists of a single E/F symbol, and the second, a single E/D symbol.

f. When labels are present:

1. Each reel of a file begins with a Beginning Label block, followed by a block consisting of a single E/F symbol.

2. Initial and intermediate reels of a multi-reel file are terminated by three successive blocks:

(a) A block consisting of a single E/F symbol.

(b) An Ending Label block.

(c) A block consisting of a single E/D symbol.

3. The final reel of a multi-reel file, or a single reel file, is terminated by four successive blocks:

(a) A block consisting of a single E/F symbol.

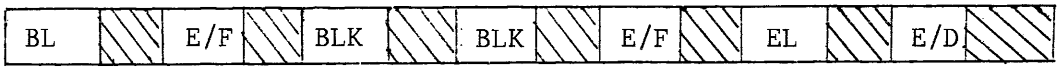
(b) An Ending Label block.

(c) A block consisting of a single E/F symbol.

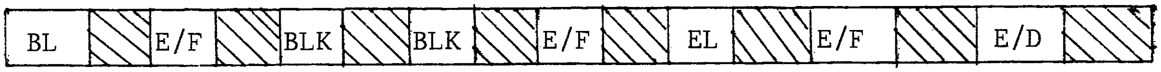
(d) A block consisting of a single E/D symbol.

4. Format of Data on Paper Tape

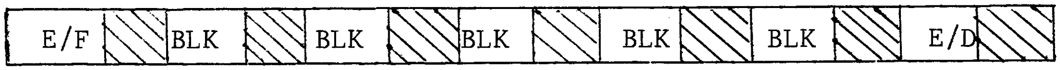
a. Initial and Intermediate Reels of a Multi-Reel File (With Labels)



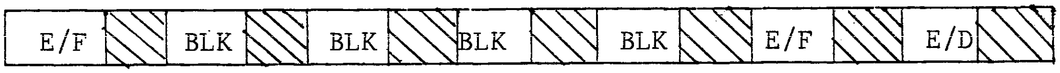
b. Final Reel of a Multi-Reel File (With Labels) or Single Reel File (With Labels)



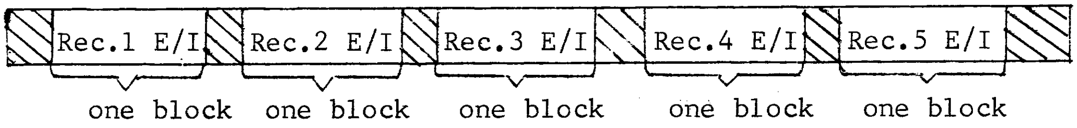
c. Initial and Intermediate Reels of a Multi-Reel File (With No Labels)



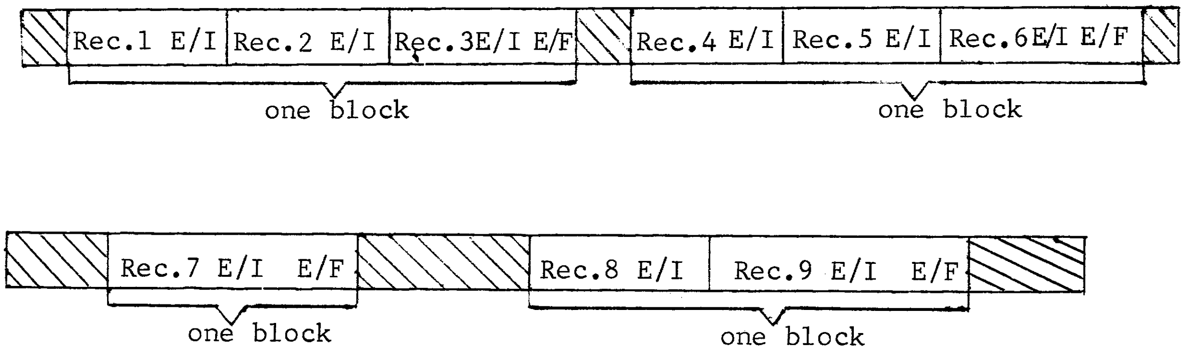
d. Final Reel of a Multi-Reel File (With No Labels) or Single Reel File (With No Labels)



e. Single Records on Tape



f. Batched Records on Tape



▨ represents an interblock gap

B. PROGRAM STANDARDS

1. Floatable Programs - must be coded with respect to HSM location 0000.

2. Program Segments - the first instruction of each program segment (block) must contain the letters PS as the first two characters and the MSC of the segment must be specified in the A Address. The B Address will contain information to identify the program segment.

3. Standard Entrance Location - the second instruction of the first program segment is assumed as the standard entrance location of each program.

4. Standard Exit Location - all programs (with the exception of a subroutine) must exit to HSM location 0770 at the end of the program in lieu of stopping within the program. The PAPER TAPE LOADER will pre-store a transfer to 0260 in this location.

5. Standard Date - the date must always be put in HSM at locations 0760 to 0767. (e.g., 07/27/61)

6. The PAPER TAPE LOADER is placed in HSM from 0250 to 0599 by the BOOTSTRAP and operates from these locations for all processors. Tape positions 7 and 8 of the LOADER BOOTSTRAP contain the HSM size indicator. This indicator will be placed in the PAPER TAPE LOADER at 0258-0259 by the LOADER BOOTSTRAP. This indicator is set as follows:

99	for	10K
I9	for	20K
Z9	for	40K

C. ORGANIZATION OF PROGRAMS STORED ON PAPER TAPE

The program instructions are punched in eighty column messages in the following format:

TAPE POSITION	CONTENTS
1	The number of instructions in the message (maximum of 6).
2 to 5	The starting HSM location of the first instruction in the message.
6	The "float indicator" for the A Address of the first instruction in the message.
7	The "float indicator" for the B Address of the first instruction in the message.

(continued)

C. ORGANIZATION OF PROGRAMS STORED ON PAPER TAPE (Cont'd)

TAPE POSITION	CONTENTS
8 to 17	The first instruction (OP N A B)
18 etc.	The "float" indicator for the A Address of the second instruction.
78 - 80	The program identification number.

D. STANDARD HALTS

N Character	Meaning
0	Normal Halts
1	Error Halts
4	Equipment Error Halts

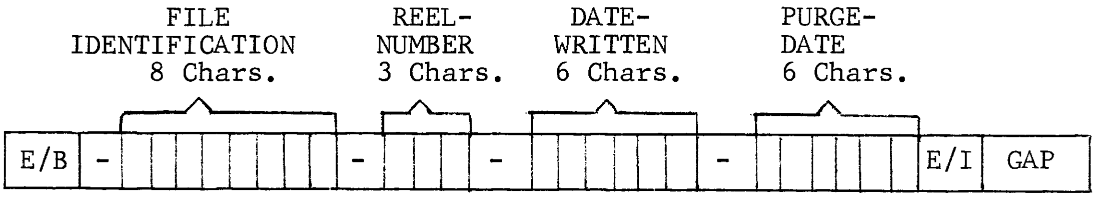
The A Address contains numeric characters to further clarify the halt.

VIII — MAGNETIC TAPE STANDARDS

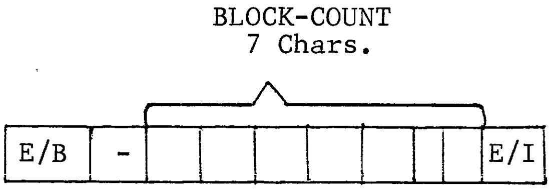
A. ORGANIZATION OF DATA

1. Standard Label Formats

   a. Beginning Label (29 Characters)



   b. Ending Label (10 Characters)



2. Tape Label Conventions

   a. Standard Labels may be indicated.

   b. Beginning and Ending Labels may be omitted.

   c. A Beginning Label may be specified without an Ending Label. However, an Ending Label may not be specified without a Beginning Label.

   d. The User may specify the format of the entire label(s) in which none of the standard items appear. (Own Coding)

   e. Additional information may appear in the label following all standard items indicated above, except the E/I symbol. The E/I symbol must always be the last character of the label.

   NOTES: 1. The PURGE-DATE of a file is the earliest date on which the contents of the file can be destroyed.

   2. The BLOCK-COUNT of a file is the total number of data blocks on a tape, and does not include Beginning or Ending Tape Labels.

3. Standards for Data on Magnetic Tape

   a. All data records are terminated by an E/I symbol.

   b. A batch may contain:

      1. A variable number of fixed length records,

      2. A fixed number of fixed length records,

      3. A variable number of variable length records,

      4. A fixed number of variable length records.

c. A batch is always terminated by an E/F symbol. The E/F must be the last character in the batch. No batch on tape, other than the labels may begin with "E/B_".

d. When the input to a run is composed of <u>all</u> single records, then no E/F symbol appears after the E/I symbol in any of the records. However, when the input to a run is batched, then each batch contains an E/F symbol as the last character of the batch even though some batches may consist of single records.

e. When labels are not present:

   1. Each reel of a file begins with a block consisting of a single E/F symbol.

   2. Initial and intermediate reels of a multi-reel file are terminated by a block consisting of a single E/D symbol.

   3. The final reel of a multi-reel file, or a single reel file, is terminated by two successive blocks, the first of which consists of a single E/F symbol, and the second, a single E/D symbol.

f. When labels are present:

   1. Each reel of a file begins with a Beginning Label, followed by a block consisting of a single E/F symbol.

   2. Initial and intermediate reels of a multi-reel file are terminated by three successive blocks:

      (a) A block consisting of a single E/F symbol.

      (b) An Ending Label.

      (c) A block consisting of a single E/D symbol.

   3. The final reel of a multi-reel file, or a single reel file, is terminated by four successive blocks:

      (a) A block consisting of a single E/F symbol.

      (b) An Ending Label.

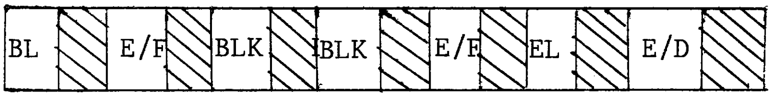
      (c) A block consisting of a single E/F symbol.

      (d) A block consisting of a single E/D symbol.

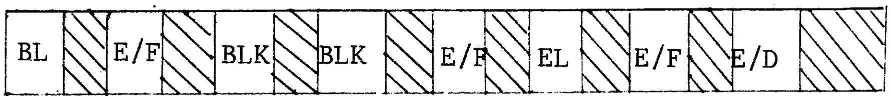
g. Octal 77 - $(77)_8$ cannot be written to a Model 381 Tape Station.

4. Format of Data of Magnetic Tape

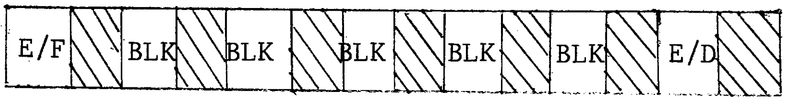
a. Initial and Intermediate Reels of a Multi-Reel File (With Labels)



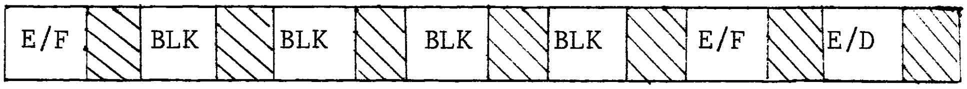
b. Final Reel of a Multi-Reel File (With Labels) or Single Reel File (With Labels)



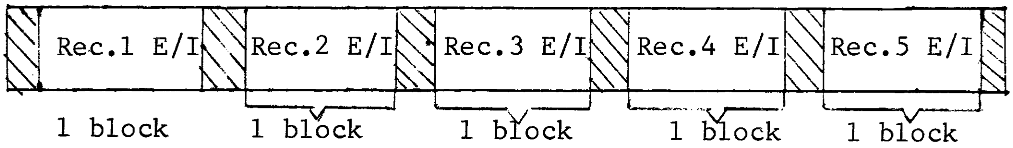
c. Initial and Intermediate Reels of a Multi-Reel File (With No Labels)



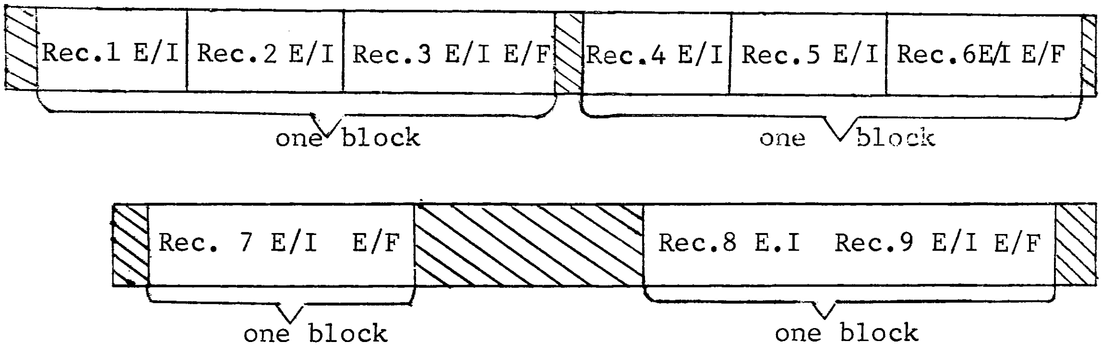
d. Final Reel of a Multi-Reel File (With No Labels) or Single Reel File (With No Labels)



e. Single Records on Tape



f. Batched Records on Tape



▨ represents an Interblock Gap.

B. PROGRAM STANDARDS

1. Standard Overlay Procedure - The TAPE INSERTION routine inserts only the first segment of each program into HSM. It is the responsibility of the object program to insert overlay segments. The following procedures are followed after reading the overlay into HSM:

   a. Place the segment number (two digits) into HSM location 0858-0859.

   b. Place the return address to the segment just inserted into HSM locations 0876-0879. If the return address is identical for each program overlay, it is necessary to complete this only once.

   c. Transfer control to HSM location 0860.

   This procedure is necessary for any type of interruption after the insertion of overlays into HSM; e.g., program testing with CONSOLIDATA or TRACER routines.

2. Program Insertion - The TAPE INSERTION routine is placed in HSM from 0250-0720 by the INSERTION BOOTSTRAP and operates from these locations for all processors. Locations 0576-0577 contain the HSM size indicator and these locations must be changed to enable the BOOTSTRAP to correctly insert the Print Table. These locations should contain:

99	for	10K
I9	for	20K
Z9	for	40K

   Provision is also made for specifying the reading device. Location 0574 is used for this purpose and should contain:

0 (zero)	for card input
2	for paper tape input.

   The above locations for HSM size and reader type are contained within the INSERTION BOOTSTRAP and the following HSM locations are set by the BOOTSTRAP within the TAPE INSERTION routine:

0256	Reading Device
0258 - 0259	HSM size indicator.

3. Program Segment Sizes - Program segments must not exceed the following limits:

5900 characters	for 10K
15900 characters	for 20K
35900 characters	for 40K

4. Program Segments - The first instruction of each program segment (block) must contain the letters PS as the first two characters and the MSC of the segment must be specified in the A Address portion. The B Address contains information to identify the program segment.

5. Standard Entrance Location - The second instruction of the first program segment is assumed as the standard entrance location of each program.

6. Standard Exit Location - All programs (with the exception of a subroutine) must exit to HSM location 0770 at the end of the program in lieu of stopping within the program. The TAPE INSERTION routine pre-stores a transfer to 0260 in this location and the reading of a STOP card will terminate the program.

7. Standard Date - The current date is stored in HSM at locations 0760-0767 (e.g., 03/15/64).

8. Standard Tape Table - A tape table is initiated and stored in HSM by the INSERTION BOOTSTRAP. All tape references within a program will reference the appropriate slot in this table, thus providing trunk switching capabilities. The table's MSC is 0835 and is composed as follows:

	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
08	1	2	3	4	5	6	A	B	C	D	E	F	J	N	L	P

9. PLT Trunk Number - Must be placed in HSM location 0200 prior to operating from the magnetic tape library.

C. ORGANIZATION OF PROGRAMS ON TAPE

1. Program Library Tape (PLT)

   a. The following first five (5) blocks on a PLT are automatically generated by the Tape Program Transcriber:

      Block 1 - Insertion Bootstrap

      Block 2 - Arithmetic Tables

      Block 3 - Standard Print Table

      Block 4 - Search Insertion

      Block 5 - Automatic Insertion

   b. Programs contained on a PLT are in alphanumeric order by the program name appearing in the Program Identification Tag. The Search Insertion would be used to load these programs. Programs may also be arranged in run to run sequence and the Automatic Insertion would be used for loading this type.

   c. All program segments appear as separate blocks on the PLT and adhere to all programming standards.

2. Program Identification Block

   Each program contained on the PLT is preceded by a tape block which provides a forty character Program Identification Tag for the program. This block appears on the following page:

2. (cont'd)

E/B _ N....NXXXX_NNN

where:

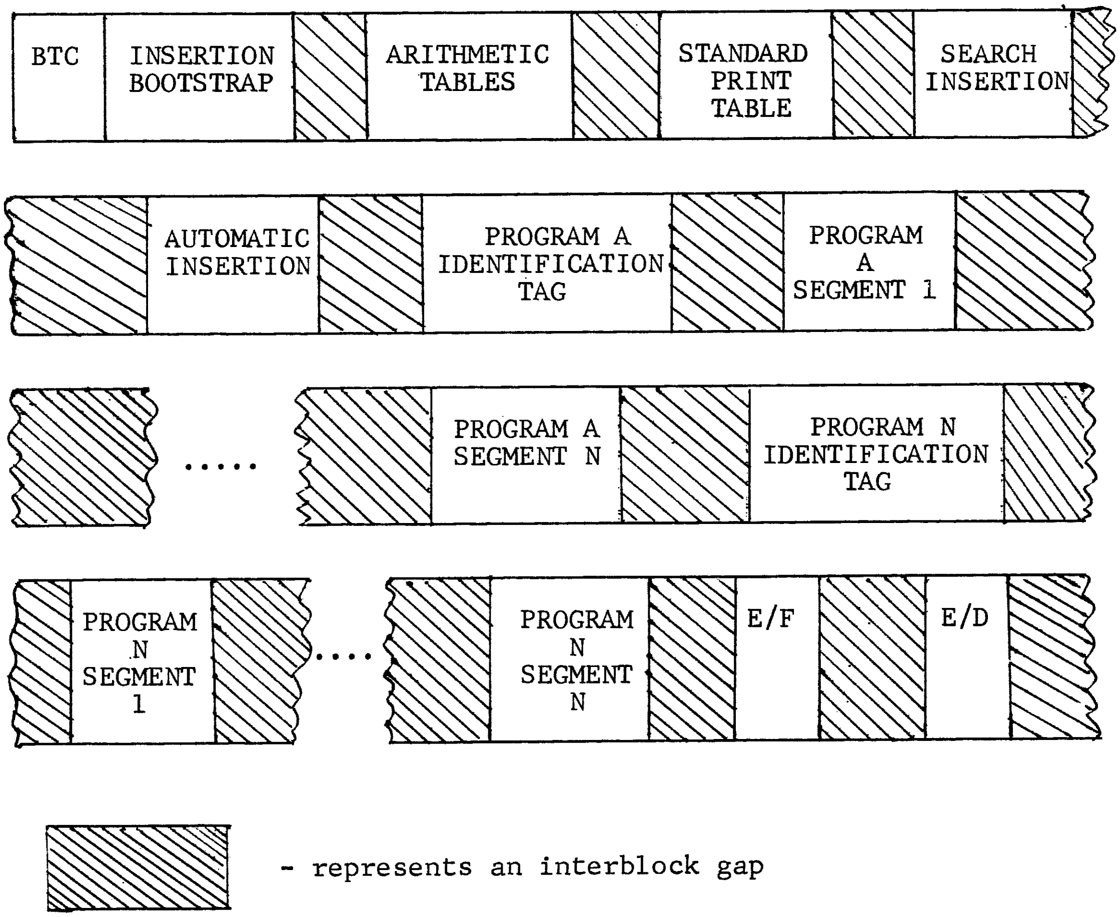
E/B _ is a two (2) character beginning tag sentinel

N....N is the thirty (30) character program name

XXXX is a four (4) character program MSC

_NNN is the library number of the PLT containing the latest updating of this program.

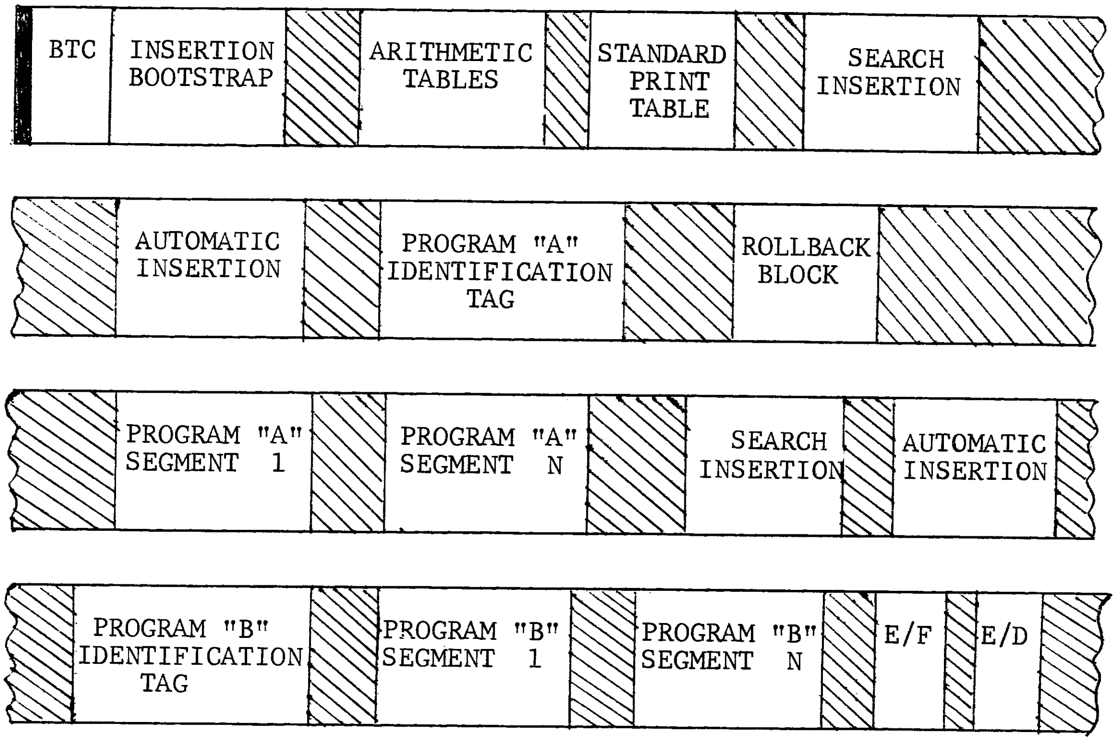
D. FORMAT OF PROGRAMS ON TAPE WITHOUT ROLLBACK



- represents an interblock gap

E. FORMAT OF PROGRAMS ON TAPE WITH ROLLBACK

(MODEL 381 TAPE STATIONS ONLY)



Program "A" utilizes Rollback and Program "B" does not.

▨ represents interblock gap.

F. STANDARD HALTS

N Character	Meaning
0	Normal Halts
1	Error Halts
4	Equipment Error Halts

The A Address contains numeric characters to further clarify the halt.

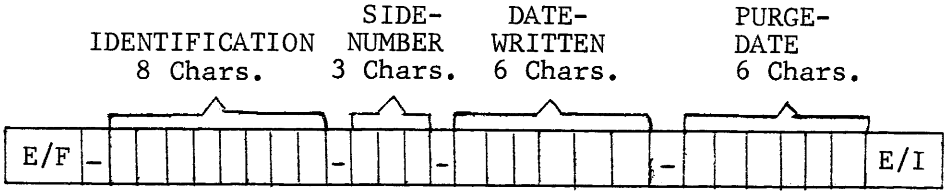
IX — DATA RECORD FILE STANDARDS

A. ORGANIZATION OF DATA

1. Standard Label Formats

a. Beginning Label (29 Characters)

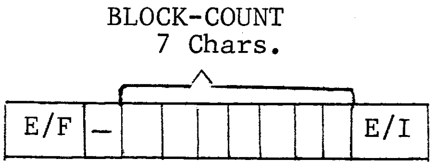
When Beginning Labels are used, they appear in the first cell of the initial band on each record side.



The SIDE-NUMBER is a count of the current record side in the file and starts at 001 for the first record side in the file.

b. Ending Label (10 Characters)

When an ending label is used, it appears only on the last record side of a file, and must always appear in the beginning of a cell. The ending label contains the count of the number of data blocks in the file. The format is as follows:



2. Data Record File Label Conventions

a. Standard Labels may be indicated.

b. Beginning and Ending Labels may be omitted.

c. A Beginning Label may be specified without an Ending Label; however, an Ending Label may not be specified without a Beginning Label.

d. The User may specify the format of the entire label(s) in which none of the standard items appear (own coding).

e. The E/B symbol must always follow the E/I symbol in a Beginning Label.

f. The E/FE/B or E/FE/D symbols must always appear following the E/I symbol in the Ending Label.

g. Additional information may appear in the label following all standard items. This information must precede the E/I symbol.

3. Standards for Data on the Data Record File

a. All data records are terminated by an E/I symbol.

b. No records may be split over cells.

c. All records must be fixed in size. If batching is used, the batch must contain a fixed number of fixed length records.

d. Data in each cell is terminated with an E/B symbol. Data is restricted to 897 characters per cell to allow for the writing of E/FE/DE/B in the same cell.

e. When labels are not present, the following sentinels can appear immediately following the E/I symbol of the last record of the last used cell, or in the very next cell of the band, as indicated. When labels are present, the following sentinels must always appear following the last character of the Ending Label, as indicated.

1. The sentinel E/DE/B terminates a partially filled record side. Additional information for the same file would appear on the next logical side to be processed. (Note that this sentinel is never used following an Ending Label.)

2. The sentinel E/FE/DE/B indicates the end of data for the last file to be processed.

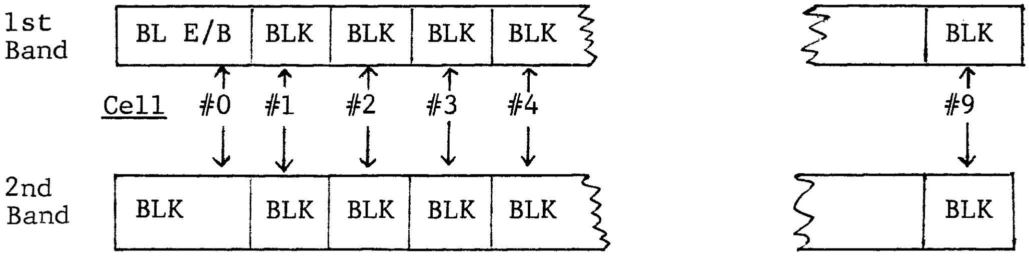
3. The sentinel E/FE/B indicates the end of data for a file.

f. The five character sentinel E/FENDE/B signifies the end of data in the first band of a record side and immediately follows the E/I symbol of the last data record. If there is insufficient space in the last data cell it will appear as the first five characters of the following cell. When this sentinel appears, the first cell in the second band must contain the next data block.

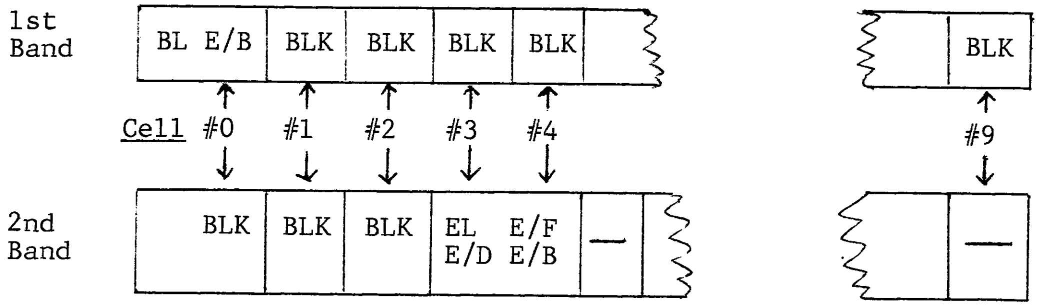
This sentinel is not used if data appears in all cells of a band or if the sentinels E/FE/B,E/DE/B, or E/FE/DE/B appear in the band.

4. Format of Data on the Data Record File

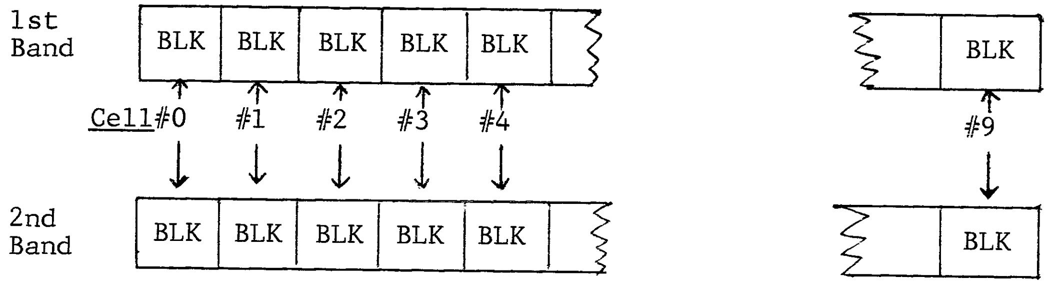
a. Initial and Intermediate Side of a Multi-Side File (With Labels)



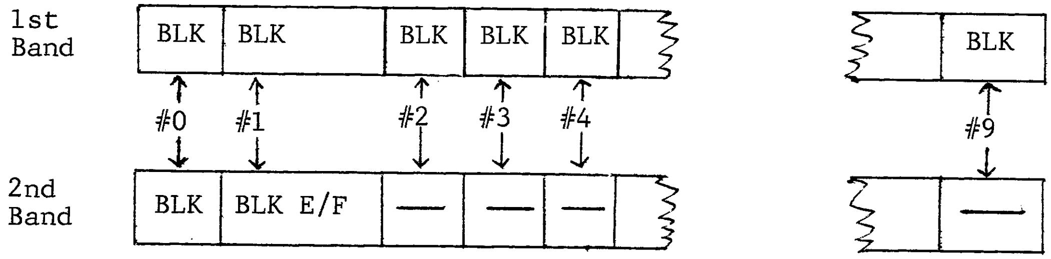
b. Final Side of a Multi-Side (With Labels) or Single Side (With Labels)



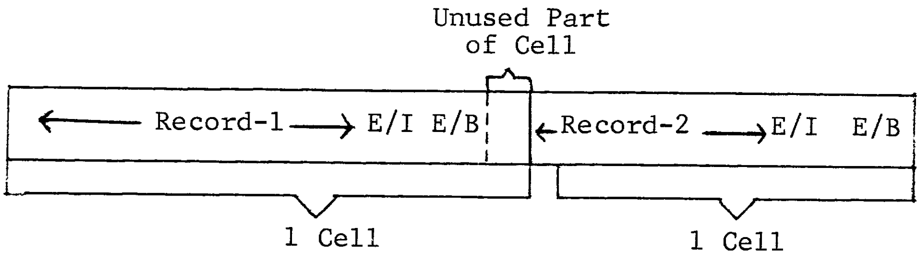
c. Initial and Intermediate Side of a Multi-Side File (With No Labels)



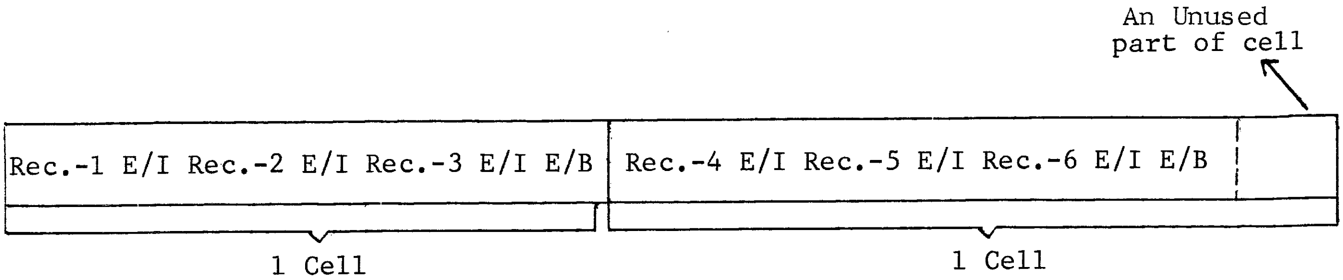
d. Final Side of a Multi-Side File (With No Labels) or Single Side File (With No Labels).



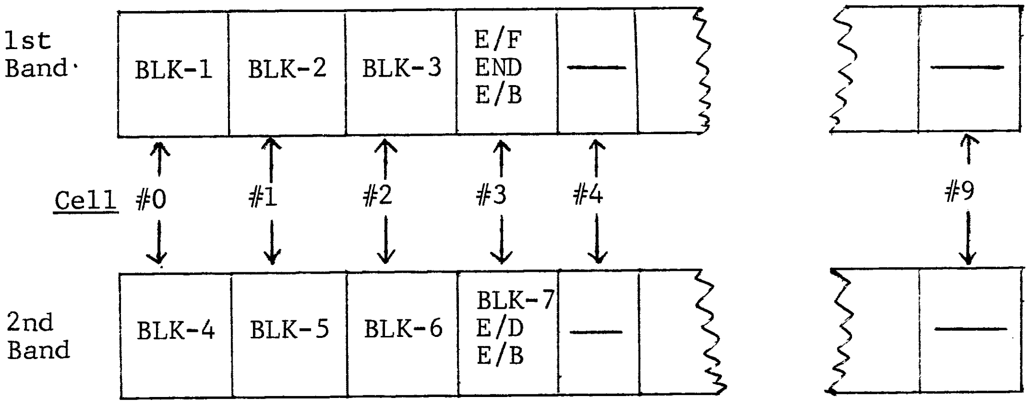
e. Single Records on the Data Record File



f. Batched Records on the Data Record File


An Unused part of cell
Rec.-1 E/I Rec.-2 E/I Rec.-3 E/I E/B
Rec.-4 E/I Rec.-5 E/I Rec.-6 E/I E/B
1 Cell
1 Cell

g. Partially Filled Record Side



B. PROGRAM STANDARDS

1. Standard Overlay Procedures

Since the INSERTION routine inserts only the first segment of each program into HSM, it is the responsibility of the object program to insert overlay segments. The INSERTION routine saves the initial band number of the last program inserted into HSM. This is available in HSM locations 0397-0399 and provides the User with the necessary information to perform overlays. The band number of a program overlay segment should be considered as zero relative to the band number of the programs initial segment ...i.e., 000, 001, 002 ... etc. The initial program band number located in HSM address 0397-0399, should be added to the relative band number before performing the band select. This allows the program disc to be placed anywhere in the Data Record File Cage for either Search or Automatic Insertions.

1. (cont'd)

The following example shows the band select set-up for a program segment overlay:

HSM ADDRESS	INSTRUCTION	COMMENT
4090	XX XXXX X001	Contains relative band number.
5000	M3 0397 5027	Move initial band number of program into Band Select Instruction.
5010	+3 5029 4099	Add relative band number to initial band number.
5020	D0 0000 0 000	Band Select Normal - the overlay segment.

The following procedure must be followed after reading the overlay into HSM:

a. Place the segment number into HSM location 0858-0859.

b. Place the return address XXXX to the segment just inserted into HSM locations 0876-0879. If the return address is identical for each program overlay, it is necessary to complete this only once.

c. Transfer control to HSM location 0860.

This procedure is necessary for any type of interruption after the insertion of overlays into HSM: e.g., program testing with the TRACER routine.

2. Program Insertion - The INSERTION routine will be placed in HSM from 0230-0599 by the INSERTION BOOTSTRAP and will operate from these locations for Model 303, 304 and 305 processors.

3. Program Segment Size - Program segments must not exceed the following limits:

a. 10K - 5900 characters

b. 20K -15900 characters

c. 40K -35900 characters

However, the first segment of a program may not exceed 8100 characters including E/B symbol for 20K and 40K. The E/B symbol must appear only as the last character of each program segment.

4. Band Number - When programming for the Data Record File, the band number of the disc must be stored in the standard HSM locations 0226-0228 after each band select. This requirement is only necessary for the standard record file, i.e., it does not apply for use of the optional Data Record File units. This standard is necessary for any type of program interruption.

C. ORGANIZATION OF PROGRAMS ON THE DATA RECORD FILE

1. The following program standards are automatically generated by the PROGRAM TRANSCRIBER Routine

   a. No more than one program may be stored on a disc.

   b. Each program must begin at the second cell (cell 1) of the initial band of the disc.

   c. The first cell (cell 0) of each even number band will contain a program identification tag in the first forty (40) positions. This label appears as follows:

      E/F_N......NXXCZZZZE/B

      where:

      E/F_ is a two (2) character beginning label sentinel.

      N....N is the thirty (30) character program name.

      XX is the side number of the disc within a multi-side program. The first side will be 01.

      C is the number of cells the first segment occupies.

      ZZZZ is the MSC of the program.

      E/B is the terminating sentinal of the program identification cell.

   d. All program segments must have an E/B symbol as its last character. No other E/B may appear in any segment. Each segment must start on a new cell.

   e. The cell following the last program segment of each program will contain as the first 5 characters E/FENDE/B.

   f. The first segment of a program may not exceed the first band of the first program side. This limits the first segment only, to 8100 characters. (Program Identification occupies the first cell of each even band.)

2. Organization of a Program Set in the Data Record File Library

   a. The first disc of a Record File Library set must be the Insertion Disc - containing in order; INSERTION BOOTSTRAP, Arithmetic Tables, Print Table, Search INSERTION and Automatic INSERTION (produced by the Insertion Generator).

   b. Programs must be stored in the record file cage in contiguous order.

c. The last disc of a Program Library Set must be a Terminating Disc. This disc will contain "E/FSTOPE/B" in the first cell (cell 0) of its initial band. This disc is not generated by the PROGRAM TRANSCRIBER, but may be produced with the EXECUTE Function.

D. FORMAT OF PROGRAMS IN A DATA RECORD FILE LIBRARY PROGRAM SET

1. First Disc of a Program Set (Insertion Disc)

Cell #	0	1	2	3	4	5	6	7-9
Band 008	Insertion ID Tag	Insertion Bootstrap	Arithmetic Tables	Print Table	Search Insertion	Automatic Insertion	E/F END E/B	NOT USED

Bands 009, 010, and 011 - Not Used.

2. Intermediate Disc of a Program Set (First Program)

Cell #	0	1-9
Band 012	Program A ID Tag (Side 01)	First Segment Program A Does Not Exceed the first band of Program Side 01, 8100 chars. ... E/B

Cell #	0-9
Band 013	Second Segment Program A 9000 Characters

Cell #	0	1-5	6-8	9
Band 014	Program A ID Tag (Side 02)	Second Segment Program A (cont.) ....E/B	Third (Last) Segment Program A ....E/B	E/FENDE/B

Band 015 - Not Used.

3. Intermediate Disc in a Program Set (Last Program)

Cell #	0	1-9
Band 016	Program B ID Tag (Side 01)	Program B - Single Segment Does not Exceed the first band of Program Side 01, 8100 chars. ....E/B

Cell #	0	1-9
Band 017	E/FENDE/B	NOT USED

3. Last Program (cont'd)

Bands 018 and 019 - Not Used.

4. Terminating Disc of Program Set

Cell #	0	1-9
Band 020	E/FSTOPE/B	NOT USED

E. STANDARD HALTS

N Character	Meaning
0	Normal Halts
1	Error Halts
4	Equipment Error Halts

The A Address will contain numeric characters to further clarify the halt.

X — DATA DISC FILE STANDARDS

A. ORGANIZATION OF DATA

1. Label Processing on the Data Disc File

   Labels appearing within the data area of the Data Disc will be treated as data records by the RCA 301 Data Disc File Software System. Therefore, label checking must be performed by the User through the use of own coding in both the Service Routines and the Sort for the Data Disc and through an appropriate user subroutine in the Assembly System.

2. FILEFILE

   a. The FILEFILE is an index of data files and work areas stored on the Data Disc. This information will be available for user label checking.

   b. FILEFILE entries will start at the beginning (first character) of a zone and will not overlap from one zone to the next. If an entry does not fit at the end of a zone, it will be placed (in its entirety) in the next zone.

   c. There can be a maximum of fifty-one FILEFILE entries per zone. The FILEFILE entries are of variable length.

   d. An E/I symbol will terminate a zone of FILEFILE entries and will follow the ISS of the last FILEFILE entry in a zone.

   e. An E/D symbol will terminate the FILEFILE and will fall in the place of the next logical FILEFILE entry.

   f. No two names in the FILEFILE may be the same.

   g. Entries must be made in the FILEFILE before writing data to the Data Disc to insure file protection.

   h. The most frequently used data files should be placed at the beginning of the FILEFILE for increased efficiency.

   i. The FILEFILE will be located in the Service Routine Track at a zone indicated by the User. The number of zones allocated for the FILEFILE is at the discretion of the User. The Service Routine Track designation, the zone location of the FILEFILE, and the number of FILEFILE zones are found in the Standard Area Extension in HSM.

   j. The items contained in each FILEFILE entry are as follows:

      1) File name - identifies the data file (8 characters).

      2) Record length - will define the record length which may be from 0003-1600 characters (4 characters).

      3) Data orientation - Sector or Zone (one character).

      4) Number of records per orientation - number of fixed records per sector or zone (3 characters).

      5) Blank - unused at present (1 character).

6) Unit word - identifies the unit and track and zone limits of the data file. There may be as many unit words as there are non-contiguous parts of a data file up to a maximum of 99 per entry (13 characters for each unit word).

7) End symbol (ISS) - each FILEFILE entry will be terminated with an ISS.

3. Standards for Data on the Data Disc File

a. Records will be fixed in length.

b. The maximum size data record may not exceed 1600 characters.

c. The minimum size data record may not be less than 3 characters.

d. Data orientation:

1) Sector Oriented Data - a file is considered sector oriented if it is formatted such that each sector contains an integral number of records (i.e., all records will end within the sector in which they begin).

2) Zone Oriented Data - a file is considered zone oriented if it is formatted such that there are an integral number of records per zone (i.e., records may be split between any or all sectors of a zone, but may not extend beyond the end of a zone).

e. Blank records will be designated by a character specified by the User. Service Routines will be written assuming this character to be a space $(12)_8$; however, documentation will be provided so that the User may replace the space with another character if he so desires.

f. A data file may be located in up to 99 non-contiguous areas on the Data Disc. Each non-contiguous area will be indicated by a unit word in the FILEFILE entry.

g. All data files and Data Disc areas will be considered as areas within four boundaries. Each data file will be defined by a beginning track and zone and an ending track and zone. Data files will always start with a new zone (in sector 0).

h. The RCA 301 Data Disc File Software will recognize one control symbol, the End Data Symbol (ED), when reading from the Data Disc. An E/D symbol appearing as the first character in the next position for a data record will signify that no more data within this data file is to be processed irrespective of the given end limits.

i. The RCA 301 Data Disc File Software will neither expect or make use of any chaining information within a data file. The appearance of chaining information which has not been defined as part of each record will cause the software to malfunction.

j. Data which is read or written from a media other than the Data Disc must conform to the Data Standards for that media.

k. Character restrictions:

1) The character (57)$_8$ may not appear within the data on the Data Disc. This character will be substituted by hardware for any character containing bad parity which is read from the Data Disc.

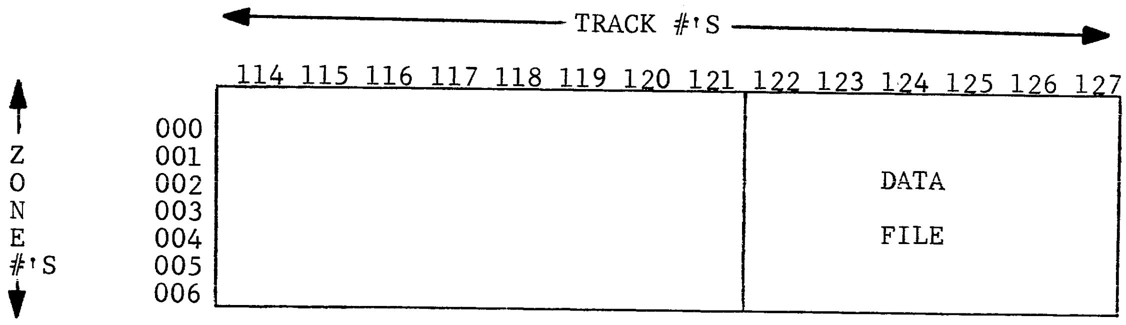
2) An (77)$_8$ will not be a legitimate data character. Data Disc File Service Routines which handle data will move records by using an (77)$_8$.

3) Other illegal data characters are (17)$_8$, E/I, E/D, E/F, and E/B.

1. A data record may not start with E/B_ in a system with tapes due to the conflict with file labels.

4. <u>Format of Data on the Data Disc File</u>

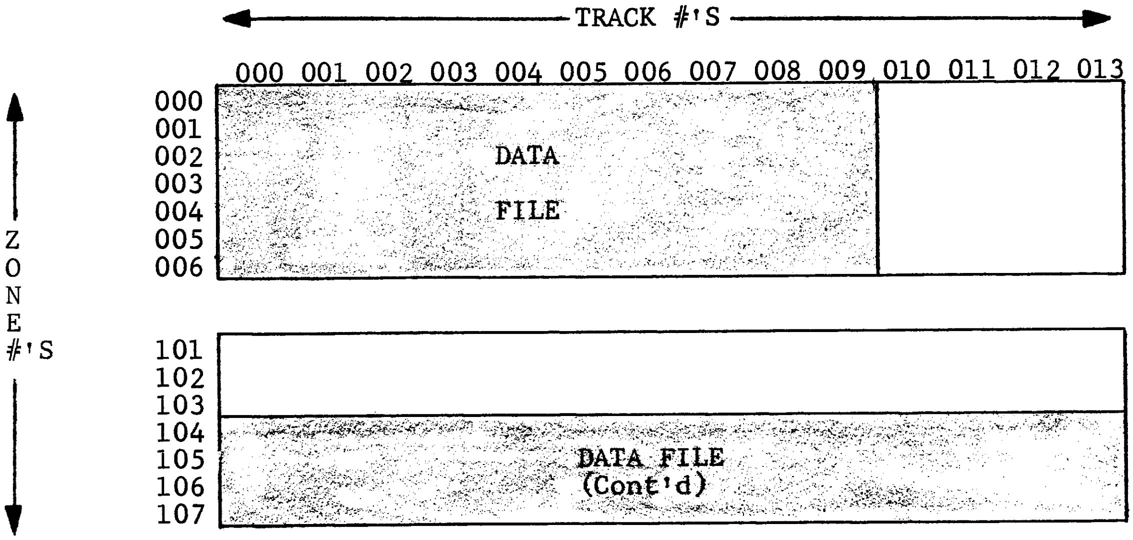
a. A four cornered concept of single Data File on the Data Disc.



Beginning on Track 122, Zone 000
Ending on Track 127, Zone 006

Processing would start with zone 000 in track 122 and will continue to zone 006 in track 122; the next zone to be processed would be zone 000 in track 123 and so on until the last zone (006) in the last track (127) were processed.

b. Four cornered concept of a Data File in non-contiguous areas on the Data Disc.



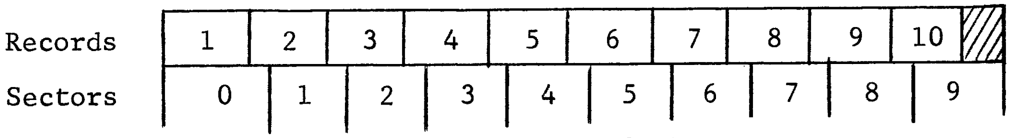
Beginning on Track 000, Zone 000
Ending on Track 009, Zone 006
} first data area

Beginning on Track 000, Zone 104
Ending on Track 013, Zone 107
} second data area

Processing would begin with zone 000 in track 000 and continue until the last zone (006) in the last track (009) for the first segment was completed. Processing would then continue with zone 104 in track 000 and would end with zone 107 in track 013. (There may be up to 99 unit words to a data file).

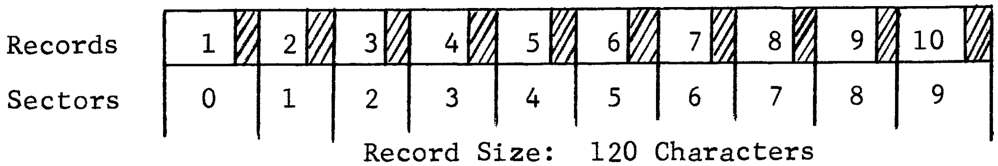
c. Records may be zone or sector oriented. This means that batches of records start at the beginning of a zone or sector. If records are a factor of 160 characters, then the records should be considered zone oriented, even though the records in fact are also sector oriented.

1) Zone Orientation



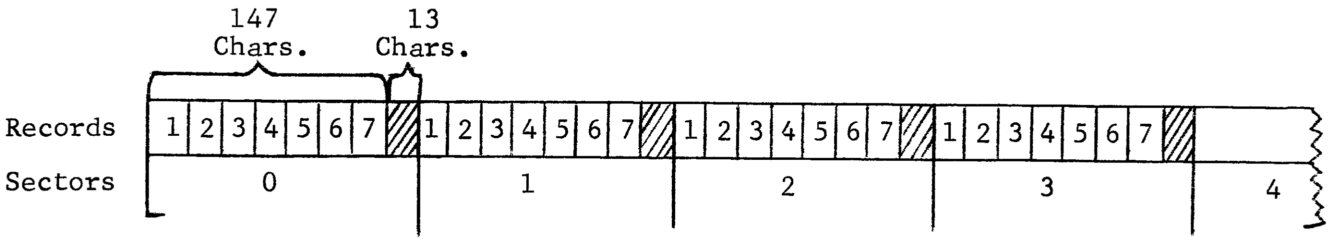
Record Size: 150 Characters

2) Sector Orientation



Record Size: 120 Characters

3) Sector Orientation



Record Size: 21 Characters

B. PROGRAM STANDARDS

1. Standard Overlay - Overlay segments are brought into HSM by the Disc to Memory Subroutine of the DATA DISC PROGRAM INSERTER. The Disc to Memory Subroutine does the following:

   a. Places the segment number into HSM locations 0858-0859.

   b. Places the return address to the segment just inserted into HSM locations 0876-0879.

   c. Transfers control to HSM location 0860 to enter the segment.

   d. This procedure allows interruption after the insertion of program segments by the TRACER routine.

2. Program Insertion - The DATA DISC PROGRAM INSERTER routine will be placed in HSM from 0250-0599 and will utilize another 840 floated HSM locations for the Disc to Memory subroutine and from 160 to 960 floated HSM locations for the Program Header. These memory requirements are for Model 303, 304, 305, 354 and 355 processors.

3. Program Segment Sizes - Program segments must not exceed the following limits:

Memory Size	Data Disc Storage
10K	7900 characters
20K	17900 characters
40K	37900 characters

4. Program Segments - The first instruction of each program segment must contain the letters PS as the first two characters and the MSC of the segment must be specified in the A Address. The B Address may contain information to identify the program segment.

5. Standard Entrance Location - The standard entrance location is the second instruction of the first program segment. Program segments other than the first segment do not have this restriction.

6. Standard Exit - All programs must exit to HSM location 0770 at the end of the program in lieu of stopping within the program. This is necessary for sequencing of programs for an automatic operation.

7. Standard Date - The date must always be put in HSM at location 0760 through 0767 (e.g., 09/07/62). This can be accomplished with the Execute function.

8. Standard Area Extension - HSM locations 0880-0999 will contain a parameter read subroutine, a standard error subroutine and constants defining the User's equipment configuration and HSM layout. The standard HSM area from 0880 thru 0999 will appear as follows:

Standard Pseudo Name	HSM Location	Instruction			Comment
PARAM	0880	M 5	0955	0941	Set Error Halt
	0890	S E/D	0002	0000	Set to continue on error
	0900	0 1			
		4 8	0600	0759	Read Parameter (Card or Pt)
	0910	S E/D	1002	0930	Sense Error
PEXIT	0920	V 1.	0219	0000	Read Subroutine Exit
ERROR	0930	V 0	0000	0000	User's Interrupt
	0940	. 0	0000	AABB*	Standard Error Halt
	0950	S E/D	0004	0001	Reset Error Indicators
EEXIT	0960	V 1	0219	0000	Error Subroutine Exit
	0970	R R	RZZZ	UUUU	Constants*
	0980	F F	FPPP	XXXX	Constants*
	0990	H 9	1TTT	ODDD	Constants*

*KEY - RRR - 3 characters. 107, 215, 323, or 431 appears in this location and represents the maximum zones available on any DDF unit under control of the first control panel (R).

ZZZ - 3 characters. 107, 215, 323, or 431 appears in this location and represents the maximum zones available on any DDF unit under control of the second control panel (S). This field will be blank when no Z unit is available.

UUUU - 4 character MSC and entrance of the Disc to Memory subroutine.

FFF - 3 character zone address of the FILEFILE.

PPP - 3 character zone address of the PROGFILE.

XXXX - 4 character HSM address of the Program Header.

H9 - 2 character HSM size flag = Z9 - 40K HSM
I9 - 20K HSM
99 - 10K HSM

1TTT - 4 characters indicating the track containing the Service Routine (TTT) and the unit containing this track (1).

ODDD - 4 character zone and sector address indicating the beginning of the Dump Area.

AA - 2 digit number indicating the number of zones allocated on the Service Routine Track for the PROGFILE.

BB - 2 digit number indicating the number of zones allocated on the Service Routine Track for the FILEFILE.

9. Service Routine Track - One track on the Data Disc will be designated as the Service Routine Track. On this track, which is designated by the User, the Program Insertion, Transcription routines, FILEFILE and PROGFILE are transcribed and the Dump Area is allocated.

C. ORGANIZATION OF PROGRAMS ON THE DATA DISC

1. A program may not run over one track to another track. Programs are restricted to one track in capacity.

2. Programs are limited to a maximum of 99 segments.

3. When a program is transcribed to the Data Disc by the DATA DISC PROGRAM TRANSCRIBER or the SCHEDULE GENERATOR, an entry will be made in the PROGFILE. A PROGFILE entry contains the following:

   a. Unit - The Data Disc Unit (1 character).

   b. Starting location - The track (3 characters) and zone (3 characters) where the Program Header starts.

   c. Total zones allocated - The total number of zones allocated for the program or schedule (3 characters).

   d. Program or Schedule Name - This name identifies a program or schedule.

4. PROGFILE entries will be merged to reflect the order of transcribed programs and schedules in disc unit, track and zone sequence.

5. There will be four PROGFILE entries per sector.

6. PROGFILE program and schedule names must be unique.

7. Program segments will be transcribed to the Data Disc in multiples of sectors. For example, if a program occupied 1000 locations, it would be transcribed to 7 sectors and would be enlarged to 1120 locations.

8. Each program will begin behind the Program Header. Since the Program Header always starts in sector 0 of a zone, the first program segment will be stored on the Data Disc according to the following characteristics of the program.

Number of Program Segments	The first program segment will start in sector*
1 - 17	1
18 - 37	2
38 - 57	3
58 - 77	4
78 - 97	5
98 - 99	6

*These numbers also indicate the numbers of sectors required by the program header for the indicated number of program segments.

9. A Program Header contains the following:

   a. Program Identification - Identifies the program which follows (8 characters).

   b. Data Disc Unit - Specifies the unit 1 (first unit) or A (second unit) (1 character).

   c. Track - Specifies the track where the program is stored (3 characters).

   d. MSC of first segment - Specifies the HSM address of the most significant character of the first program segment (4 characters).

   e. Number of segments transcribed - The number of segments transcribed.

   f. Size of Program Header - The number of sectors occupied by the Program Header minus one (1 character).

   g. Control Panel - 0 will specify the first unit; & will specify the second unit (1 character).

   h. Blanks - Not used at present (4 characters).

   i. Segment word entries - (eight characters for each segment)

1.	Program segment starting sector	1 character
2.	Program segment starting zone	3 characters
3.	Number of meaningful sectors in program segment	3 characters
4.	Total sectors of program segment minus one in the first zone (including blanks)	1 character
*5.	Number of blank sectors associated with program segment	2 characters

*These two characters will be "packed" in the zone bits of the preceding 4 characters. Thus, the segment word remains 8 characters.

10. The DATA DISC PROGRAM TRANSCRIBER determines the space required for the Program Header and generates it during a transcription.

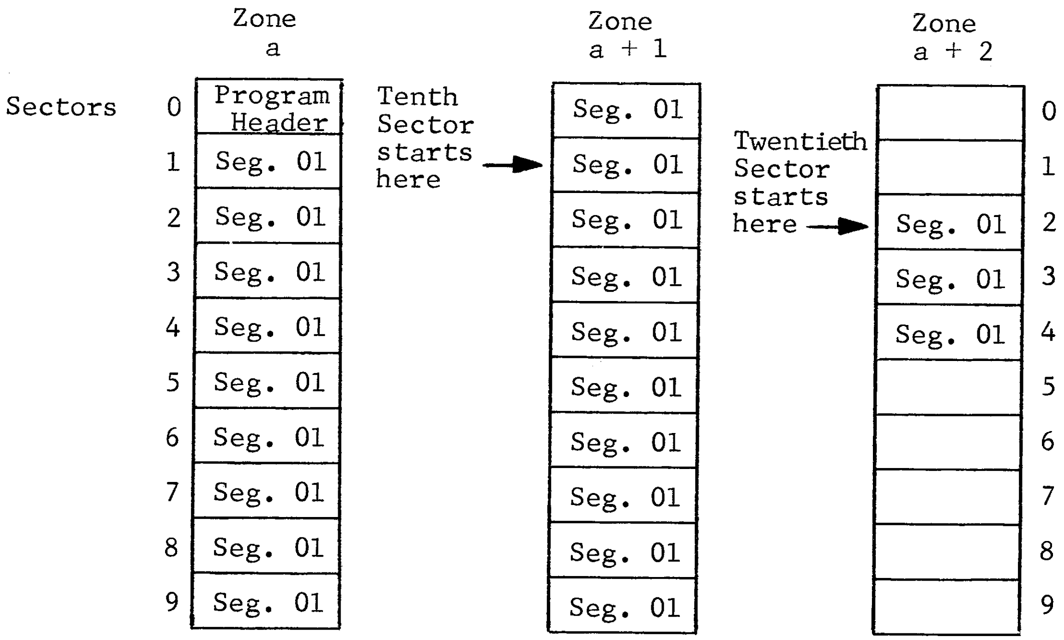
11. A program must be contained within one track and all program tracks are reserved for programs only. The Program Header will start in the first sector (0) of a zone. The program will start in the first sector following the Program Header. Successive segments will start in the next sector following the preceding segment, unless intervening blank sectors are specified. Each program will start in a new zone. The size of a track is the only limiting factor to the number of programs stored on a given track.

12. The Program Header will be placed in HSM when the program is inserted by the INSERTION routine. The header will be retained there during the operation of the program in order to expedite the insertion of program segments. If the number of program segments exceed 17, an allowance will have to be made for the larger Program Header required when inserting the program.

13. In one segment programs, the area occupied by the Program Header may be used for coding. In multi-segment programs all unused portions of the Program Header may be used by the program.

14. Only one program may be stored in a zone. This means that if a program occupies only one sector of a zone, no other programs may be transcribed to that zone and all other sectors in that zone except the one containing the Program Header are considered unused, unless designated blank.

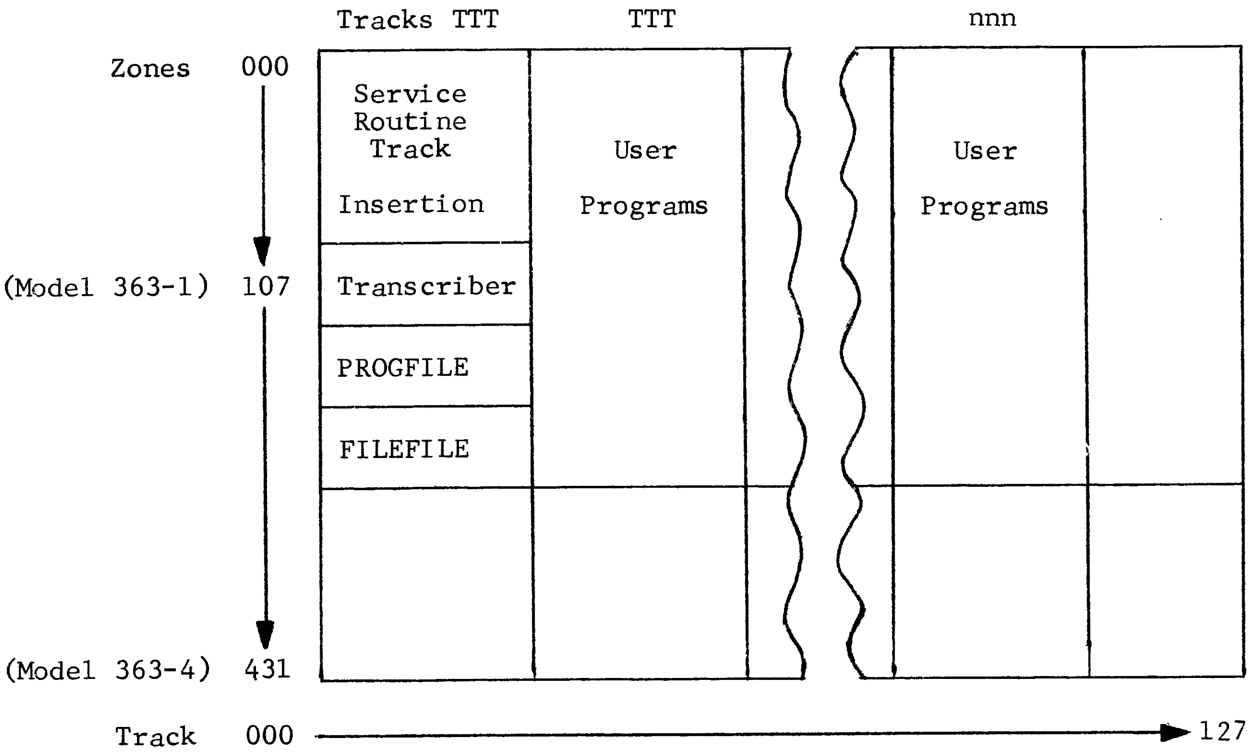
15. When a program segment is transcribed to the Data Disc a spiral pitch of one is utilized so that when the segment is inserted into HSM latency is reduced. This means that when a program segment covers two or more zones there is a stepping up of the sector number for each succeeding zone of the program segment. For example, consider the first program segment contains 3520 locations which would be 22 sectors long. It would be stored as follows:



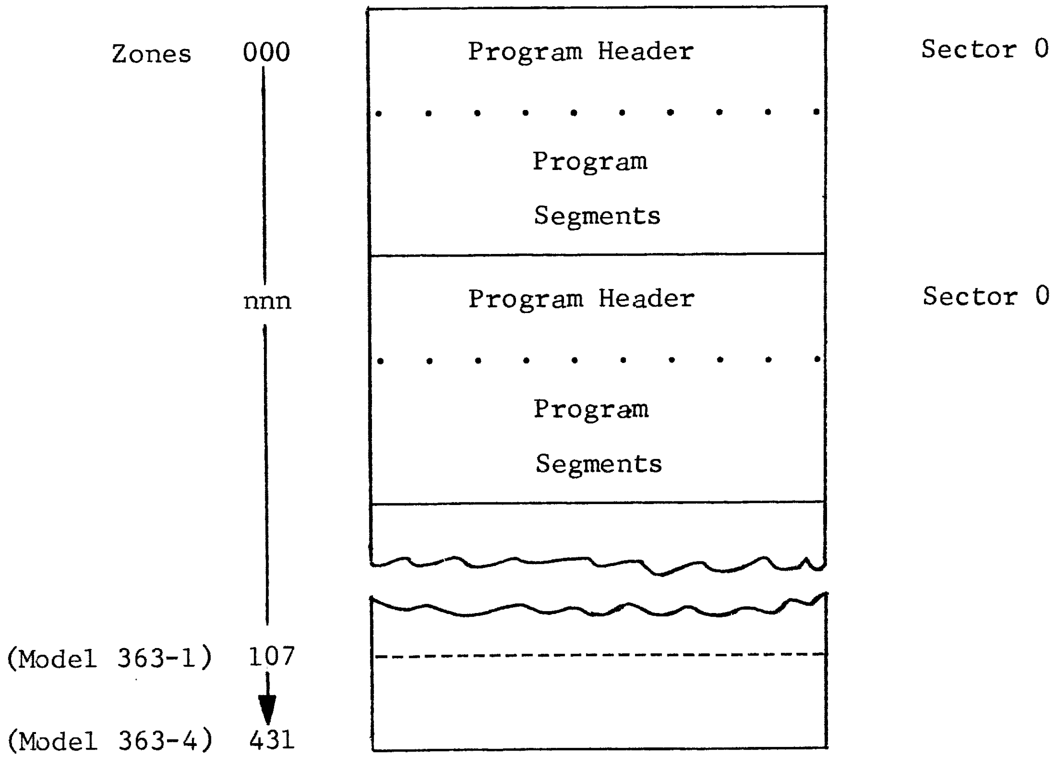
16. Program tracks may not be mixed with data tracks. That is, a program and data may not appear in the same track.

17. It is the User's responsibility to allocate the Data Disc for program as well as data storage. The Data Disc Service Routines will check to insure that there is no conflict. However, the Service Routines do not automatically allocate program or data storage.

D. FORMAT OF PROGRAMS ON THE DATA DISC



PROGRAM TRACK



E. STANDARD HALTS

N Character	Meaning
0	Normal Halts
1	Error Halts
4	Equipment Error Halts

The A Address will contain numeric characters to further clarify the halt.

XI — RANDOM ACCESS COMPUTER EQUIPMENT STANDARDS

A. ORGANIZATION OF DATA

1. 3488 Unit of Storage

   a. Record - A record is the smallest unit of information handled by 3488 software. To facilitate storage and retrieval, records are grouped into logical units of storage which are referred to as Buckets.

   b. Bucket - A bucket is the basic unit of information storage in a 3488 System. A bucket has many of the properties of a "batch" on tape.

   Whereas the physical 3488 card is subdivided into blocks and bands, the logical 3488 card is subdivided into buckets. Therefore, the bucket is the logical addressing unit. It is composed of one or more consecutive 650-character blocks as specified by the user. The physical address of a bucket is the hardware address of the first block within it.

2. Standards for Data

   a. All data records are terminated by an E/I symbol.

   b. A bucket may contain:

      1) A fixed number of fixed-length records.

      2) A fixed number of variable-length records.

      3) A variable number of fixed-length records.

      4) A variable number of variable-length records.

   c. The $(77)_8$ character is a restrictive character used by 3488 software and can not be used as a data character.

3. Format of Bucket

   A bucket (Figure 1) contains a Bucket Header (B) and Data Records (R). If accessing in either the Sequential or Random Mode, a bucket (Figure 2) may also contain Tag Address Records (T).

   A Bucket Header contains bucket control information (See System Control Formats). A Data Record is a unit of file information. A Tag Address Record is an entry representing a data record that has been placed in another (visitor) bucket.

   Each record must be fully contained within a bucket.

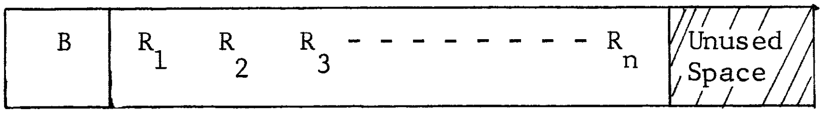


Figure 1 - Bucket Layout (Serial Mode)

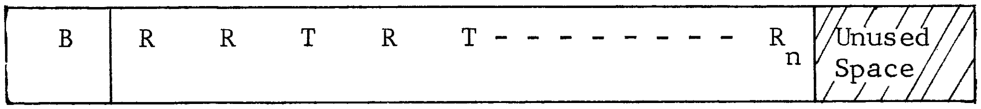


Figure 2 - Bucket Layout (Sequential or Random Mode)

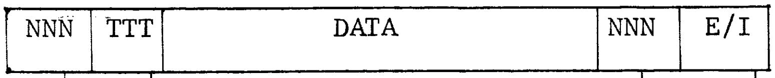
4. Format of Data and Tag Address Records

The following record formats are applicable to all 301-3488 software.

a. 301 Standard Record Format (Figure 3) - A record consists of a fixed or variable number of characters followed by an E/I $(52)_8$.

Figure 3 - 301 Standard Record Fromat

b. 301/3301 Record Format (Figure 4) - This format is designed to provide a compatible record format that is acceptable for processing on both the RCA 301 and the RCA 3301. All records of this format must consist of a group of 10-character units. The record consists of a 6-character control field followed by data that is followed by a 3-character control field and an E/I symbol. The maintenance of these control fields is the responsibility of the user when processing on the RCA 301.



TTT - a 3-character field containing additional characters used in moving the record.

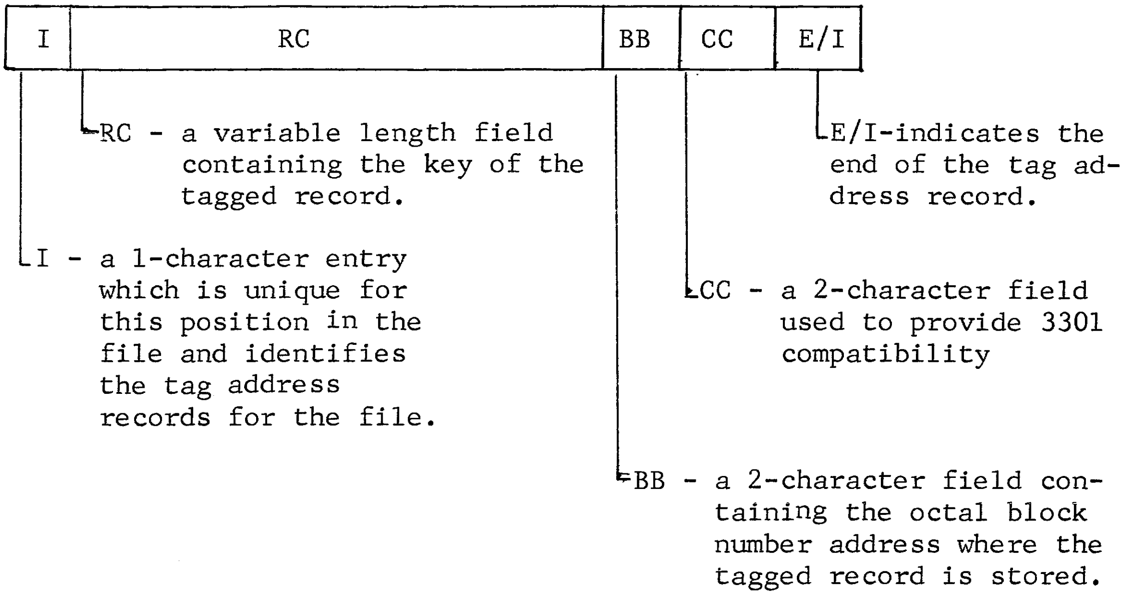
E/I - indicates the end of the record.

NNN - a count of 10-character units in the record. This count is required only if the record is to be processed on the 3301 with tape that is reading reverse. Otherwise, it may be used for data.

NNN - a 3-character count of the number of 10-character units in the record.

Figure 4 - 301/3301 Record Format

c. Tag Address Record Format (Figure 5) - A Tag Address Record consists of a Tag ID character (I), record key (RC), and the visitor bucket address (BB) where the tagged record is stored. The Tag is identified by a unique character specified by the user.



RC - a variable length field containing the key of the tagged record.

E/I-indicates the end of the tag address record.

I - a 1-character entry which is unique for this position in the file and identifies the tag address records for the file.

CC - a 2-character field used to provide 3301 compatibility

BB - a 2-character field containing the octal block number address where the tagged record is stored.

Figure 5 - 301 Tag Address Record Format

5. File Mapping (Unit Words)

File Mapping is the process by which a user describes the storage of information files within the 3488 memory. Because of the division of 3488 memory into cards, magazines, etc., it is necessary to describe the storage of a file in terms of sets of these components. The file areas are defined by the unit words that are contained in the description of the file in the System Control Card.

Each unit word defines one file area within one magazine. A file area consists of sets of buckets occupying the same physical positions on each of a set of cards. Both the buckets within a card and the cards themselves are each contiguous. Therefore, one or more unit words define all the file areas required.

A unit word contains the following elements:

- Consecutive Card Numbers within a magazine
- Consecutive Block Numbers within a Card
- Number of buckets defined by this unit word.
- The Logical Address of the last bucket within the file area defined by the unit word.

The following are some guidelines for defining unit word:

- A magazine may contain up to a maximum of 65 unit words.

- A unit word can start at any block position within a card.

- Individual unit words for a file may be located on different magazines, on different units, and on different channels. One file may have up to 86 different unit words.

See page XI-10 for detailed format of unit word.

B. BASIC STORAGE AND ACCESS METHODS

The Basic Storage and Access Methods describe the manner and data unit in which information is introduced, held, and then extracted at a later time.

Once the storage pattern has been established for a particular file and the information has been placed on the device, the information may be accessed by bucket (device) or record level in any one of three modes. Also, data transfer may be effected in any one of three 3488 Input-Output methods.

1. Unit of Data Information

The user may handle data information within the 3488 System in one of two levels:

a. Bucket (device) level - Data information may be handled by buckets, using either the hardware instructions or the 3488 hardware macros. This level of information processing requires the user to batch or unbatch his information for record porcessing.

b. Record level - Data information will be delivered to the user one logical record at a time by using the 3488 Record Processing macros. At this level the System Control Information (System Control Card, Magazine Control Cards, Bucket Headers) is established and maintained by the 3488 Utility and Macro routines.

2. Data Storage

Data may be stored on the 3488 in three methods.

a. Serial Storage - Data is loaded on the 3488 in a serial fashion, consecutive buckets or records one after another, disregarding any criteria.

b. Sequential Storage - Data is loaded on the 3488 in a sequential fashion, one record after another by primary key.

c. Random Storage - Data is loaded on the 3488 in a random fashion with each bucket or record location determined by a random or pseudo-random code or address.

3. Access Methods

The accessing method defines the order in which 3488 memory locations are selected for the purpose of transferring information.

a. Serial Access - provides for the selection of 3488 memory locations in physical ascending order. Each reference to the file is to the next higher location within that file. This method is normally applied to a serially-stored file.

b. Sequential Access - provides for the selection of 3488 memory locations in order by primary key of the records. Each reference to the file will be to the location containing the record with the next higher primary key. This is accomplished through the use of chaining records (tag linkage) from home bucket to visitor bucket. Chaining results when a record will not fit into a bucket in its sequential order.

c. Random Access - provides for the selection of 3488 memory locations where each location is independent for each access. Reference can be made through indices, random-address techniques, or pseudo-random code generation. A record key is the link to the logical bucket address.

4. Input-Output Methods

The direction of information transfer between the 3488 and HSM is as follows:

a. Input - This method transfers information from the 3488 to HSM. The Serial Read, Sequential Read, and Logical Read macros are three macros that will accomplish information retrieval.

b. Output - This method transfers information from HSM to the 3488 (e.g., the Serial Write and Random Write macros are two macros that will accomplish information storage).

c. Overlay - This method transfers information between the 3488 and HSM for updating purposes. For example, the Serial Overlay macro will retrieve a record, and then transfer it back to the same location on the 3488.

C. SYSTEM CONTROL FORMATS

1. System Control Card

The System Control Card contains a description of every data file and every magazine contained in the system. The System Control Card is normally but not necessarily the first card of the first magazine; there may be more than one System Control Card per system.

DATA FILE INDEX	(8 BLOCKS)
DATA FILE DESCRIPTION	(138 BLOCKS)
ON-LINE CATALOGUE	(2 BLOCKS)
MAGAZINE SERIAL NUMBER INDEX	(8 BLOCKS)
MAGAZINE DESCRIPTION	(100 BLOCKS)

Figure 6 - System Control Card Layout

a. Data File Index - For each data file entered into the system, its File ID is entered into the Data File Index and a complete description of this data file is entered into the Data File Description.

The Data File Index consists of eight blocks; each block containing a 10-character header. Each entry in the Data File Index contains the eight-character alphanumeric File ID and a two-character binary address of its associated Data File Description.

The format of the Data File Index is:

HEADER (10)*	CONTROL CARD LABEL (10)	DATA FILE ID (8)	ADDR. OF DATA FILE DESC. (2)	DATA FILE ID (8)	ADDR. OF DATA FILE DESC. (2)	
		Data File 1		Data File 2		

* - number of characters

b. Data File Description - The Data File Description consists of 138 blocks, each containing a 10-character header. Each entry consists of 35 characters of fixed information, a variable-size magazine table, and an optional file label area of any size as long as it fits into the block.

The format for a Data File Description is:

HEADER (10)	FIXED INFORMATION (35)	MAGAZINE TABLE (variable)	FILE LABEL AREA (optional)	E/I

A Data File Description entry can not exceed a block nor cross blocks; however, there may be more than one entry per block. It should be noted that <u>each</u> Data File Description entry <u>must</u> be decade oriented with the terminal E/I symbol as the last character of the entry.

The contents of each field are defined below:

1) Fixed Information

Contents	No. of Chars.
Number of 10-character words in this entry	2
Data File ID	8
Number of blocks per bucket	2
Number of characters for File Label Area	3
Maximum number of characters per bucket	5
Tag Identification Character	1
Record Type	1
F = Fixed Length Record	
V = Variable Length Record	
Record Format	1
S = 301 Standard Record Format	
C = 301/3301 Record Format	
Maximum number of characters per record	4
Key Length	2
Key position from LHE of record	3
File Organization	1
S = Serial	
Q = Sequential	
R = Random	
Number of unit words	2
Total Number of Characters	35

2) Magazine Table - The Magazine Table contains the information necessary to locate the Unit Word(s) for a given file contained within the system.

Contents	No. of Chars.
Magazine Serial Number	5
Logical Sequence Number of unit words*	2
Total Number of Characters	7n, where n = the number of unit words

*This entry defines the order in which the unit word(s) are entered into the file. For example, if a file is allocated across three magazines, the unit words needed to describe the area allocated for this file would be logical numbers 01, 02, and 03. If another unit word is necessary, it would be designated as logical sequence number 04.

3) File Label Area - This area is reserved for unique information requirements of each installation. The size of this area is specified by the user and the information is maintained and/or processed by the user.

c. On-line Catalogue - The On-line Catalogue consists of two blocks, each block containing a 10-character header. Each entry consists of seven characters for each magazine on the system. Its purpose is to keep a record of all magazines on the system at object time.

Contents	No. of Chars.
Magazine Serial Number	5
R/W Station (Octal Value) (0-7)	1
Magazine Number (Binary Value) (0-15)	1
Total Number of Characters	7n, where n = the number of magazines on-line

HEADER (10)	MAGAZINE 1 (7)	MAGAZINE 2 (7)	MAGAZINE n (7)

Entries on the On-line Catalogue are grouped by R/W Station; the first 16 entries represent R/W Station 1 [Device Number $(60)_8$], the second 16 entries represent R/W Station 2, etc.

d. Magazine Serial Number Index - The Magazine Serial Number Index consists of eight blocks, each block containing a 10-character header. Each entry consists of the magazine serial number, the allocator control indicator, the blank magazine indicator, and the binary address of its associated magazine description. Every magazine in the entire system must be entered into the Magazine Serial Number Index.

Contents	No. of Chars.
Magazine Serial Number	5
Allocator Control Indicator (0=301 Allocation)	1
Blank Magazine Indicator* (B=Blank)	1
Binary Address of the associated Magazine Description	2
Total Number of Characters	9

*This entry indicates that the entire magazine is blank; i.e., no area has been allocated in the magazine.

e. Magazine Description - The Magazine Description consists of 100 blocks, each block containing a 10-character header. Each entry contains a description of all files on a magazine and their allocated area(s).

Contents		No. of Chars.
Number of 10-character words in this entry		3
Magazine Serial Number		5
File Information Table		36N
File ID	8	
Logical Sequence Number of unit word	2	
File Type (A=Data File)	1	
Unit Word*	25	
	36	
Total Number of Characters		8+36N, where N=the number of file areas in the magazine

HEADER (10)	MAGAZINE DESCRIPTION 1	E/I	MAGAZINE DESCRIPTION 2	E/I	

Note that each Magazine Description must be decade-oriented with the terminal E/I symbol as the last character of the entry.

*All entries in the unit word are decimal.

*Unit Word

The unit word defines a contiguous file area within one magazine.

Contents	No. of Chars.
Starting Card Number	3
Ending Card Number	3
Starting Block Number	3
Ending Block Number	3
Number of buckets defined by Unit Word	5
Last Logical Bucket Number	8
Total Numbered Characters	25

*All entries in the unit word are decimal.

2. Magazine Control Card

The Magazine Control Card is normally the last card of the magazine; there may be more than one Magazine Control Card with the linkage to the next Magazine Control Card contained in the first entry of the Flaw and Overflow Index. The Magazine Control Card maintains an index of all flaws and overflows, and storage area for flaws and overflows.

The format is:

FLAW/OVERFLOW INDEX	(8 BLOCKS)
FLAW/OVERFLOW AREA	(248 BLOCKS)

a. Flaw/Overflow Index - The size of the Flaw/Overflow Index is eight blocks, each block containing a 10-character header. The Flaw/Overflow Index is a 17-character entry for each flaw bucket, tag overflow bucket, or card overflow bucket. Its purpose is to locate the flaw or overflow area which has been allocated for a flaw bucket, tag overflow bucket, or card overflow bucket. The format is:

HEADER (10 Chars)	CONTROL CARD LABEL (17 Chars)	FLAW/OVER-FLOW INDEX (17 Chars)	FLAW/OVER-FLOW INDEX (17 Chars)	

Contents	No. of Chars.
File ID	8
Binary Card Number in which Flaw/Overflow occurred	2
Binary Address of Flaw/Overflow Bucket on Home Card	2
Binary Address of Bucket on this card where the Flaw/Overflow Bucket is stored	2
Number of Blocks per Bucket	2
Type Entry F = Flaw Bucket 0 = Card Overflow E = Empty (Flaw or Overflow Bucket has been deleted) T = Tag Overflow	1
Total Number of Characters	17

b. Flaw/Overflow Area - The size of the Flaw/Overflow Area is 248 blocks. Its purpose is to provide storage area for all buckets that can not be stored on the card where they belong (home card) within the magazine. The buckets are of three (3) types:

Flaw Buckets
Tag Overflow Buckets
Card Overflow Buckets

Bucket sizes appearing in this area are dependent upon the size of the original bucket within the file encountering this condition.

3. Control Card Label

The Control Card Label entry provides the software system with a means of identifying particular control cards. On the System Control Card(s) it appears as the first Data File Index entry (10 characters). On the Magazine Control Card(s), the Control Card Label appears as the first entry in the Flaw/Overflow Index (17 characters).

Contents	No. of Chars.
Magazine Serial Number	5

Type of Control Card
S = System Control Card
M = Magazine Control Card
	1

Logical Number of Control Card (01, 02, 03, ... NN)	2
Binary Address of Next Control Card (** = last Control Card)	2
Total Number of Characters	10

4. Bucket Header

The Bucket Header (the first 10 characters of every bucket) is used to indicate:

a. the status of the bucket (tag, overflow, flaw), and

b. the data (records and tags) limits within this bucket.

The format is:

FLAW/OVERFLOW INDICATOR (1 Char)	FILE CONTROL CHARACTER (1 Char)	A FIELD (4 Chars)	B FIELD (4 Chars)

Contents	No. of Chars.
Flaw/Overflow Indicator	1

0 = Card/Tag Overflow in this bucket. Go to Magazine Control Card.
1 = No Flaw or Card/Tag Overflow in this bucket.
& = Flaw and Card/Tag Overflow in this bucket. Go to Magazine Control Card.
A = Flaw in this bucket. Go to Magazine Control Card. (Flaw Bucket Replacement technique)
/ = Flaw detected on Read with Flaw Bucket not on Magazine Control Card (Skip Bucket technique).
- = (minus) Overflow not permissible.

File Control Character

E/D = $(55)_8$ - specifies the last home bucket on this card for a sequentially organized file within the current Unit Word. May be followed by overflow area.
E/B = $(72)_8$ - specifies that this is the last card containing information for this file within the Unit Word currently being operated on.
E/F = $(56)_8$ - specifies that this is the last card and bucket containing information for this file.
1 = $(01)_8$ - no control character is present in this bucket.

	Contents	No. of Chars.
A Field -	Actual size of area (in characters) currently containing records and tags in this bucket	4
B Field -	Maximum size of area (in characters) to contain records and tags in this bucket	4
	Total Number of Characters	10

D. PROGRAM STANDARD

Standard HSM allocation - the address of the System Control Card must be placed in HSM location 0853-0856 prior to running 3488 Service Routines or opening a 3488 file. Allocation is as follows:

0853 - Device Number (octal)
0854 - Magazine Number (octal)
0855-0856 - Card Number (binary) - See following table for binary equivalents of card number.

301/3488 ADDRESSES

BLOCK OR CARD NUMBER	0	1	2	3	4	5	6	7	8	9
0	00	01	02	03	04	05	06	07	08	09
1	0 sp	0#	0@	0(	0)	$0(17)_8$	10	11	12	13
2	14	15	16	17	18	19	1 sp	1#	1@	1(
3	1)	$1(17)_8$	20	21	22	23	24	25	26	27
4	28	29	2 sp	2#	2@	2(	2)	$2(17)_8$	30	31
5	32	33	34	35	36	37	38	39	3 sp	3#
6	3@	3(	3)	$3(17)_8$	40	41	42	43	44	45
7	46	47	48	49	4 sp	4#	4@	4(	4)	$4(17)_8$
8	50	51	52	53	54	55	56	57	58	59
9	5 sp	5#	5@	5(	5)	$5(17)_8$	60	61	62	63
10	64	65	66	67	68	69	6 sp	6#	6@	6(
11	6)	$6(17)_8$	70	71	72	73	74	75	76	77
12	78	79	7 sp	7#	7@	7(	7)	$7(17)_8$	80	81
13	82	83	84	85	86	87	88	89	8 sp	8#
14	8@	8(	8)	$8(17)_8$	90	91	92	93	94	95
15	96	97	98	99	9 sp	9#	9@	9(	9)	$9(17)_8$
16	sp 0	sp 1	sp 2	sp 3	sp 4	sp 5	sp 6	sp 7	sp 8	sp 9
17	sp sp	sp #	sp @	sp (	sp)	$sp(17)_8$	#0	#1	#2	#3
18	#4	#5	#6	#7	#8	#9	#sp	# #	#@	#(
19	#)	$\#(17)_8$	@0	@1	@2	@3	@4	@5	@6	@7
20	@8	@9	@sp	@#	@ @	@(	@)	$@(17)_8$	(0	(1
21	(2	(3	(4	(5	(6	(7	(8	(9	(sp	(#
22	(@	((	( )	$((17)_8$	)0	)1	)2	)3	)4	)5
23	)6	)7	)8	)9	) sp	)#	)@	) (	))	$)(17)_8$
24	$(17)_8 0$	$(17)_8 1$	$(17)_8 2$	$(17)_8 3$	$(17)_8 4$	$(17)_8 5$	$(17)_8 6$	$(17)_8 7$	$(17)_8 8$	$(17)_8 9$
25	$(17)_8$sp	$(17)_8\#$	$(17)_8@$	$(17)_8($	$(17)_8)$	$(17)_8(17)_8$	——	——	——	——

THE MOST TRUSTED NAME IN ELECTRONICS

Radio Corporation of America