SPERRY RAND

UNIVAC

9400 SYSTEM

UTILITY AND SERVICE ROUTINES

PROGRAMMERS
REFERENCE

UP-7713, Rev. 1

This document contains the latest information available at the time of publication. However, the Univac Division reserves the right to modify or revise its contents. To ensure that you have the most recent information, contact your local Univac Representative.

PRINTED IN U.S.A.

CONTENTS

FIGURES

TABLES

1. INTRODUCTION

1.1. GENERAL

This document describes the utility and service routines provided for the UNIVAC 9400 System. It includes descriptions of the acceptable conventions and the linkage commands required to perform a particular utility or service operation. Use of this document assumes a knowledge of the contents of the *UNIVAC 9400 System Job Control Programmers Reference, UP-7793* (current version), *UNIVAC 9400 System Assembler/Central Processor Unit Programmers Reference, UP-7600* (current version), *UNIVAC 9400 System Supervisor Programmers Reference, UP-7689* (current version), and *UNIVAC 9400 System Data Management System Programmers Reference, UP-7629* (current version).

Utility and service routines are provided to assist the user in accomplishing common functions in operating the UNIVAC 9400 System. Some of the common functions are not described here but are described fully in their own programmers reference documents. These functions include sorting data according to a specified order and merging of data to facilitate processing, maintaining files on magnetic tape, and linking output modules of language processors into a single executable program. The manuals are: *UNIVAC 9400 System Sort/Merge Programmers Reference, UP-7664* (current version); *UNIVAC 9400 System Tape Librarian Programmers Reference, UP-7667* (current version); *UNIVAC 9400 System Linkage Editor Programmers Reference, UP-7703* (current version).

This document is therefore concerned primarily with describing an assortment of many of the other service and utility routines which are needed to perform miscellaneous day-to-day functions. For example, some of the routines transfer data from one media to another, or from one area of a peripheral unit to another, or create, alter, or delete files according to the needs of the user. The control information required by most of these routines is furnished as parameters by means of Job Control, but in some cases are supplied by the problem program itself.

1.2. STATEMENT CONVENTIONS

The conventions used to illustrate statements in this manual are as follows:

- Capital letters and punctuation marks (except braces, brackets, and ellipses) indicate information that must be coded exactly as shown.
- Lowercase letters and terms represent information that must be supplied by the user.
- Information contained within braces represents necessary entries, one of which must be chosen.
- Information contained within brackets represents optional entries that are included or omitted, according to program requirements.
- Braces within brackets signify that one of the entries must be chosen if that operand is included.
- Ellipses (series of three periods) indicate that the number of entries is not defined.
- Commas are required when positional parameters are omitted, except for trailing positional parameters.

2. UTILITY AND SERVICE ROUTINES

2.1. MAGNETIC TAPE PREPARATION ROUTINE

The magnetic tape preparation routine (UTPREP00) initializes tapes in standard label format by writing an initial volume label (VOL1), a dummy file header label (HDR1), and a tape mark. The routine operates under control of Job Control in the UNIVAC 9400 Operating System. A series of job control statements defines the particular aspects of the routine and directs its execution.

A succession of tapes may be prepared in one execution of the routine, and up to nine tapes can be prepared simultaneously, depending on the number of tape units available.

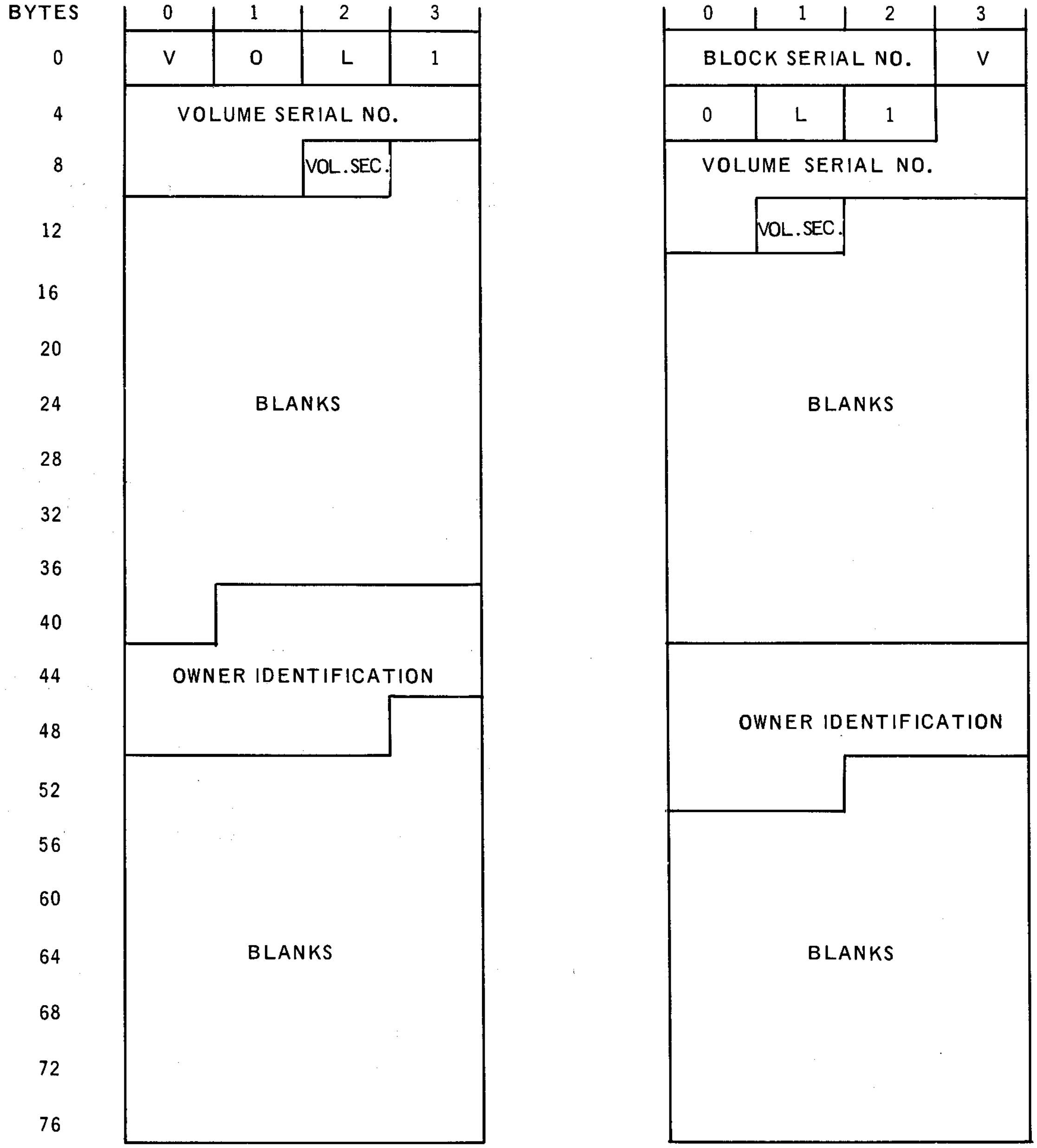
The VOL1 label identifies the tape reel and its owner, and is used to check that the proper reel is mounted. When a tape is first used at an installation, the serial number and other volume information are specified by parameter cards supplied to the magnetic tape preparation routine. The serial number should also be written on the exterior tape label of the reel for visual identification.

2.1.1. Tape VOL1 Label Format

The VOL1 label format is shown in Figure 2–1. Although six bytes are reserved for the volume serial number, and 10 bytes for owner identification, the user is not required to specify these parameters. If not specified, a volume serial number of 000010 in EBCDIC code is assumed, while the owner identification field is filled with blanks.

The volume serial number, given or implied, can be specified as the first in a series by means of an additional parameter which indicates the number of tapes in the series. In this case, the user must also supply a value to be used as an increment to the initial volume serial number. (If an alphanumeric serial number is to be incremented, only the first numerics encountered in a right-to-left scan will be incremented. For example, 10R15C incremented by 7 becomes 10R22C.) If no series of tapes is involved, the above two parameters are not specified. All other bytes in the 80-byte VOL1 label contain blanks.

If the console option for specifying block serial numbers is employed, then a three-byte block serial number appears in the first three bytes of the VOL1 label. The volume serial number is then moved back by three bytes; that is, to bytes 7 through 12. Similarly, the owner identification field is moved to bytes 44 through 53.



WITHOUT BLOCK SERIAL NUMBER

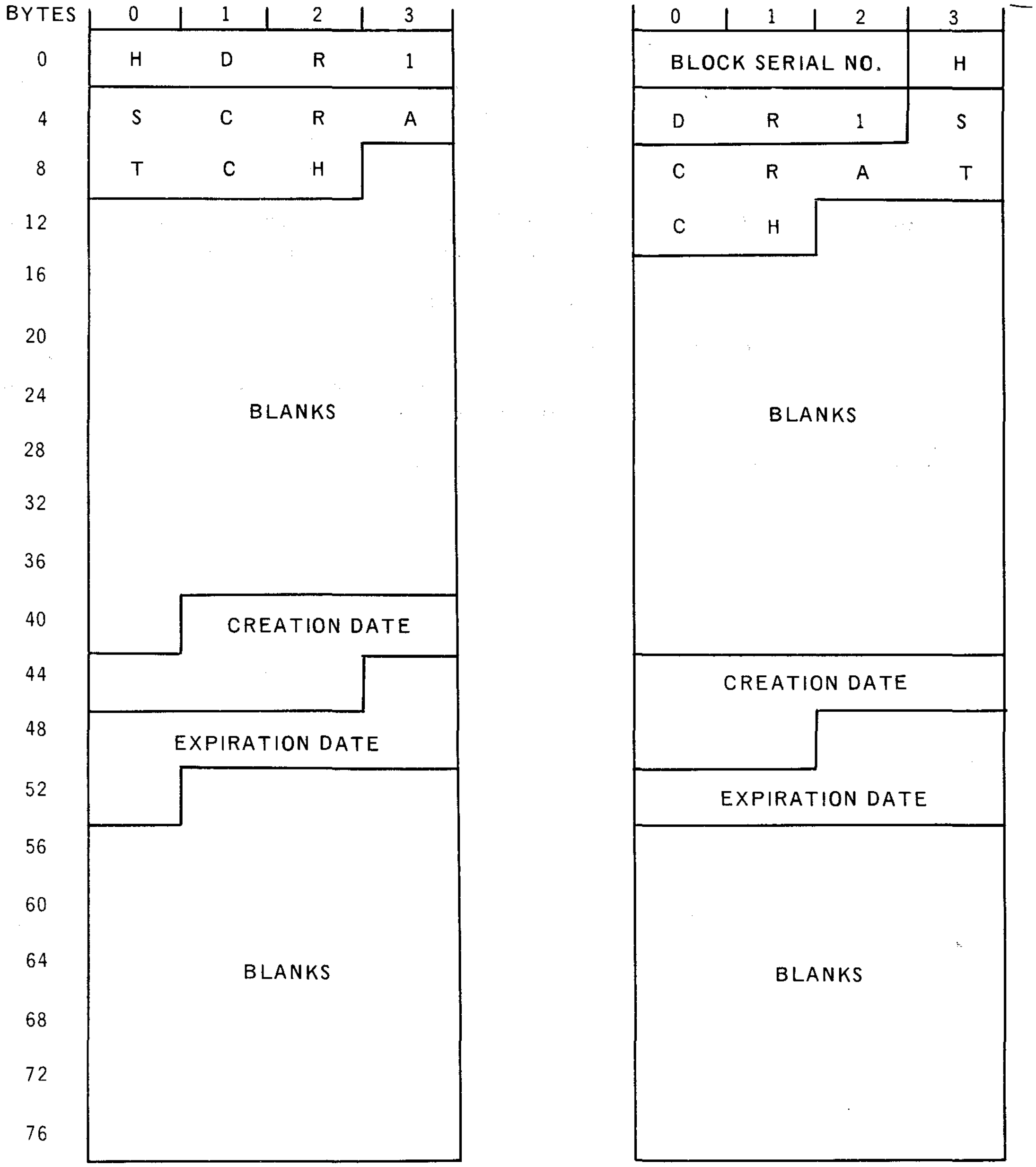
WITH BLOCK SERIAL NUMBER

Figure 2-1. Tape VOL1 Label Format

Byte Numbers		
Without Block Serial No.	With Block Serial No.	
	0 – 2	Block Serial Number
0 – 2	3 – 5	Label Identifier – Contains VOL to indicate that this is a volume label.
3	6	Label Number – Always 1, for the initial volume label.
4 – 9	7 – 12	Volume Serial Number – Unique identification assigned to a reel when it enters the system. This number should also be written on the outside of the reel. It is usually numeric but may be any six alphanumeric characters.
10	13	Volume Security – Reserved for future use by installations requiring security at the reel level.
11 – 40	14 – 43	Contains blanks.
41 – 50	44 – 53	Owner Identification – Unique identification of the owner of the reel. It may be any combination of 1 to 10 alphanumeric characters.
51 – 79	54 – 79	Contains blanks.

2.1.2. Tape HDR1 Label Format

The HDR1 label has 80 bytes reserved for it, but only a portion of these are used for the dummy header format; see Figure 2–2. The characters SCRATCH appear in bytes 4 through 10, while the creation and expiration dates (obtained from the System Information Block) appear in bytes 41 through 46 and 47 through 52, respectively. All other bytes contain blanks. If the block serial number (console) option is employed, then these byte assignments will be shifted back by three bytes, as explained for VOL1 label format in 2.1.1.



WITHOUT BLOCK SERIAL NUMBER

WITH BLOCK SERIAL NUMBER

Figure 2–2. Tape HDR1 Label Format

Byte Numbers		
Without Block Serial No.	With Block Serial No.	
	0 – 2	Block Serial Number
0 – 2	3 – 5	Label Identifier – Contains HDR to indicate a file header label.
3	6	Label Number – Always 1.
4 – 10	7 – 13	Dummy File Identifier – A 7-byte field that uniquely identifies the dummy file SCRATCH.
11 – 40	14 – 43	Contains blanks.
41 – 46	44 – 49	Creation Date – The date on which the file was created. The date is expressed in the form yyddd and is right-justified. The leftmost position is a blank.
47 – 52	50 – 55	Expiration Date – The date on which the file may be written over or used as scratch. The date is in the same form as the creation date.
53 – 79	56 – 79	Contains blanks.

2.1.3. Parameter Statement

The parameter (PARAM) statement, which is used to submit operational parameters to the magnetic tape preparation routine by means of the job control stream, includes the following positional parameters: the volume serial number, an increment value, the number of tapes in a series to be prepared, and the owner's identification. Normally, the PARAM statement or statements immediately follow the EXEC statement in the job control stream. To properly identify the magnetic tape preparation routine, the routine's code name UTPREP00 must appear in the EXEC statement.

The format of the PARAM statement is:

OPERATION ƀ	OPERAND
// PARAM	[volume-serial-number][,increment][,number-of-tape-in-a-series][,owner-identification]

POSITIONAL PARAMETER 1

volume-serial-number – may be any six alphanumeric characters.

if blank – a value of 000010 is assumed.

POSITIONAL PARAMETER 2

increment – may be one to four decimal digits; this value is used to increment volume serial number when a series of tapes is to be prepared.

if blank – a value of 0010 is assumed.

POSITIONAL PARAMETER 3

number-of-tapes-in-a-series – may be one to three decimal digits.

if blank – assumes that only one tape is to be prepared.

POSITIONAL PARAMETER 4

owner-identification – may be 1 to 10 alphanumeric characters, left-justified.

if blank – no identification given.

Example:

```
1        10        20        30        40
// PARAM 134,,,WØRKTAPE
```

Volume serial number is 134, only one tape is to be prepared, and owner's identification is WORKTAPE。

2.1.4. Job Control Stream

Entry into the magnetic tape preparation routine is accomplished by means of the job control stream. A sample sequence of such statements is given below. Parameters are entered into the magnetic tape preparation routine by means of the PARAM statement, as described in 2.1.3. For a full description of the control statements, refer to the *UNIVAC 9400 System Job Control Programmers Reference, UP-7793* (current version).

Sample job control stream:

```
// JOB   jobname
// DVC   6
// LFD   TAPEOT01
// DVC   7
// LFD   TAPEOT02
// DVC   8
// LFD   TAPEOT03
// EXEC   UTPREP00
// PARAM   50,,1,UTILITY
// PARAM   75,20,3,ASSEMBLY
// PARAM   134,,,WORK TAPE
/&
```

NOTE: Logical File Definition (LFD) statements can be TAPEOT01 through TAPEOT09; any other LFD names will not be recognized.

2.1.5. Console Messages

There are three messages, or inquiries, which may be printed at the console, at various points during the running of the routine. A description of each of the console messages follows.

```
¡¡ UP01  BLOCK NUMBER REPLY (Y,N).
¡¡ R Y Ⓢ
```

If the reply is Y, VOL1 and HDR1 labels will be 83 bytes each in length. VOL1 will be block number 1 and HDR1 will be block number 2. If the reply is other than Y, VOL1 and HDR1 labels will be 80 bytes each in length and the blocks will not have block serial numbers.

```
¡¡ UP02 DISMOUNT TAPES.
¡¡ UP02 TAPEOT01 = VOL. SER. NO. 000000 = UTILITY
```

jj UP02 REPLY (C) CONTINUE
jj R C Ⓢ

This message is typed after all tapes mounted for labeling have been written and rewound with interlock. The second line starts a list of the logical unit names (TAPEOT01 ... TAPEOT09), volume serial numbers, and owner identification (for example, UTILITY) for each tape labeled. At this point, the operator may label the tapes before mounting the next series.

If a reply other than C is given, the routine will reinitialize immediately and begin reading the cards from the job control stream.

jj UP03 MOUNT BLANK TAPES ON ALL AVAILABLE UNITS. REPLY (C)CONTINUE
jj R C Ⓢ

If a reply other than C is typed in, the magnetic tape preparation routine goes directly to the end-of-job control statement (/&).

NOTE: jj is the job number assigned by job control.

2.2. REPRODUCER ROUTINE

The Reproducer routine (UTREPR00) is a means of copying programs or data from an input card deck to an output card deck. A printed listing of the card decks processed may be produced. This routine can also be used to list cards only. The routine operates under control of Job Control in the UNIVAC 9400 Operating System. A series of job control statements define the particular aspects of the routine and direct its execution.

Optionally, the information contained in card columns 73 through 80 may be used for changing the program identification and updating the sequence numbers.

2.2.1. Card Conventions

The card input information may be either source program code or data. But in order to be reproduced correctly, only printable EBCDIC code can be used for input. Any columns containing nonprintable code are not punched; when the list option is specified in the *P parameter statement (2.2.2), the line containing nonprintable code is flagged and a blank is shown on the listing for the column containing the error. Otherwise, card output will be the same as the input, except for any desired changes in columns 73 through 80:

Columns 73,74	– program identification
Column 75	– revision number
Columns 76...80	– sequence numbers

If the sequence numbers are revised, they will begin with 00010 and be incremented by 10. This will allow for 9999 records in the input card deck, with up to nine insertions between records.

Each input card deck to be processed must be preceded by a special parameter (*P) card as described in 2.2.2 and be ended with a blank card.

2.2.2. Parameter Statements

Two types of parameter statements are required to submit operational parameters to the Reproducer routine. The first of these is the normal PARAM statement used by Job Control following an EXEC statement. As used here, the PARAM statement includes three positional parameters, defining the input device, output device, and number of files.

The format of the PARAM statement is:

OPERATION ƀ	OPERAND
// PARAM	CARD, $\left\{\begin{array}{l}\text{CARD}\\\text{NONE}\end{array}\right\}$,number-of-files

POSITIONAL PARAMETER 1

CARD – input device must be specified as CARD.

POSITIONAL PARAMETER 2

CARD – specifies a card-to-card reproduce operation.

NONE – specifies a printed listing of the input deck only.

POSITIONAL PARAMETER 3

number-of-files – may be one or two decimal digits (1–99), indicating the number of files (card decks) to be processed.

Examples:

```
   1         10         20         30         40
1. // PARAM CARD,CARD,3
2. // PARAM CARD,NONE,1
```

1. Card-to-card reproduction of three files.

2. Printed listing of one card input file.

The second parameter (*P) statement must be included as the first card of each input file to be processed.

The format of the *P statement is:

OPERATION ƀ	OPERAND
*P	$\left\{\begin{matrix}Y\\N\end{matrix}\right\}$ [,identification] [,revision-number] [,LIST]

POSITIONAL PARAMETER 1

Y – indicates that the following file is to be resequenced (columns 76 through 80).

N – indicates that the following file is not to be resequenced.

POSITIONAL PARAMETER 2

identification – specifies the new two-character program identification to be used in columns 73 and 74.

if blank – no change in program identification is required.

POSITIONAL PARAMETER 3

revision-number – specifies the new revision number to be used in column 75.

if blank – no change in revision number is required.

POSITIONAL PARAMETER 4

LIST – specifies that a printed listing of the output file is desired.

if blank – no listing is desired.

NOTE: Each file (deck) is copied, as specified by the *P statement, until a blank card is reached. If the next card following the blank card is a new parameter card, the routine is reinitialized and the next file is copied. The program is terminated by end-of-data (/*) and end-of-job (/&) control statements.

Examples:

```
   1        10        20        30        40
1. *P N,,,LIST
2. *P N
```

1. File is to be reproduced exactly as it appears in the input card deck, and the output is to be listed on the printer.

2. Same as 1 with no listing.

2.2.3. Job Control Stream

Entry into the Reproducer routine is accomplished by means of the job control stream. A sample sequence of such statements is given below. Parameters are entered into the Reproducer routine by means of the PARAM and *P statements, as described in 2.2.2. For a full description of the control statements, refer to the *UNIVAC 9400 System Job Control Programmers Reference, UP-7793,* (current version).

Sample job control streams:

- *CARD-TO-CARD*

```
// JOB jobname

// DVC 2

// LFD PUNCH

// DVC 3

// LFD PRNTR

// EXEC UTREPR00

// PARAM CARD,CARD,3

/$

*P Y,ID,1,LIST  \
                 |
     .           |
     .           >  FILE 1
     .           |
                 |
blank card      /

*P N,AB,,LIST   \
                 |
     .           |
     .           >  FILE 2
     .           |
                 |
blank card      /
```

```
*P Y,,9

        .
        .          FILE 3
        .

blank card
/*
/&
```

- *LIST ONLY*

```
// JOB jobname
// DVC 3
// LFD PRNTR
// EXEC UTREPR00
// PARAM CARD,NONE,1
/$
*P N,,,LIST

        .
        .          FILE
        .

blank card
/*
/&
```

2.3. SYMBOLIC ANALYZER ROUTINE

The Symbolic Analyzer routine (UTSYMB00) reads a source program card deck in assembler language and produces as output an alphabetic listing of the symbolic tags used in the source program. If there is sufficient available storage, the listing can be obtained in one continuous reading of the source program, or in segments in which upper and lower limits are assigned to determine the range of each program segment. The routine operates under control of Job Control in the UNIVAC 9400 Operating System. A series of job control statements define the particular aspects of the routine and direct its execution.

Certain options are available for supplementing the output with a listing of all the source cards, and for handling nonstandard data cards. Diagnostic error message codes (flags) are also provided by the analyzer.

2.3.1. Output Format

The output listing appears with references given in terms of assumed input card numbers. Each entry on the list includes the symbol, the input card number where the symbol was defined, and a list of all the card numbers where the symbol was referenced. In the case of multiple definitions, any definitions after the first are listed on succeeding lines in the definition field, beginning at the line following the last reference.

Two special cases are indicated in the output format as follows:

- An * appears in place of a definition for an EXTRN directive in the source program.
- An S precedes the card definition for a SET symbol.

If the storage available is not sufficient to hold all symbols and references at one time, the tables are printed out when they are full. The Symbolic Analyzer then continues reading the source program, filling the tables again. In this case, it is necessary for a complete reference to look up each symbol listed in each section of the program.

If the storage available is quite small, it is still possible to obtain a listing in which each symbol appears only once by limiting the range of the symbols analyzed in one pass of the source code. To do this the source code deck must be preceded by a card specifying the lower and upper alphanumeric limits of the symbols which are to be analyzed in this pass of the source code. The format of this special limits card is:

OPERATION ␢	OPERAND
–	P_1,P_2

where column 1 is a hyphen, column 2 is a space, and P_1, P_2 are the two positional parameters that specify the lower and upper limits, respectively.

POSITIONAL PARAMETER 1

P_1 – a field up to eight characters in length which specifies the lower alphanumeric limit of the symbols to be analyzed.

POSITIONAL PARAMETER 2

P_2 – a field up to eight characters in length which specifies the upper alphanumeric limit of the symbols to be analyzed.

A listing of all the source cards (with assumed card numbers) is also provided to facilitate cross referencing between the source program and the output listing. If this list is not needed, it can be deleted by placing before the source deck a special parameter card which has the following fixed format:

```
NO LIST
```

where the N is in column 1 and there is only one space between the two words.

An example of the output format produced by the Symbolic Analyzer routine is given below. In this example, both the source program listing and the symbol output listing are included, with the latter limited to those symbols beginning with the character A and being equal to or less than the character string GA.

- Source Program Card Listing

```
0001   ABLE    LH      R6$,GA
0002   FRAME   MVC     CHARGE(4),ABLE
0003   SMART   DC      A(CHARGE)
0004           EXTRN   BANDAID
0005   ABLE    DC      H'0'
0006           CLC     FRAME(6),SMART
```

- Symbolic Analyzer Output Listing (A to GA)

```
ABLE       0001      0002
           0005

BANDAID    *         0004

CHARGE               0002      0003

FRAME      0002      0006

GA                   0001
```

2.3.2. Nonstandard Input Cards

By using an Input Format Control (ICTL) statement, it is possible to analyze a deck of cards which does not conform to the standard starting (column 1), ending (column 71), and continue columns. If used, the ICTL statement card must immediately precede the source code deck.

The format of the ICTL statement is:

LABEL	ƀ OPERATION ƀ	OPERAND
not used	ICTL	[b] [,e] [,c]

where: b is an unsigned decimal integer specifying the begin column.

e is an unsigned decimal integer specifying the end column, and must be greater than b+5 and less than or equal to 80.

c is an unsigned decimal integer specifying the continue column, and must be greater than 6 and less than e.

If b is omitted, it is assumed to be 1. If e is omitted, it is assumed to be 71. If c is omitted or if e is equal to 80, continuation records are not allowed.

2.3.3. Job Control Stream

Entry into the Symbolic Analyzer routine is accomplished by including certain statements in the job control stream. A sample sequence of such statements is given below. Special parameters are entered into the routine by means of the statement cards, as described previously. For a full description of all job control statements, refer to the *UNIVAC 9400 System Job Control Programmers Reference, UP-7793* (current version).

Sample job control stream:

```
// JOB jobname

// DVC 3

// LFD PRNTR

// EXEC UTSYMB00

/$

- A,G

NO LIST

    ICTL   5,60,20

    (Source deck)

/*

/&
```

This example indicates a job in which no source listing is desired, only symbols between and including A and G are to be analyzed, and the source deck cards are not in standard format.

2.3.4. Error Message Codes

Diagnostic error codes (flags) signify incorrectly specified source code statements. The appropriate codes are printed on the program listing on the same line as the statements containing errors; that is, in the margin to the left of the card images.

The following error codes are used in the Symbolic Analyzer routine:

E – indicates an expression error.

G – indicates a statement too large; that is, the number of characters in the statement exceeds the size of the buffer from which the statement is processed.

X – indicates a continuation error.

U – indicates an unrecoverable expression error.

The E, G, and X error codes indicate that there is an error in the statement, but that all the symbols in the statement except the symbol in error have been analyzed. A U error code indicates that the symbol in error and the remaining portions of the statement have not been analyzed.

2.4. TAPE-TO-PRINT DUMP ROUTINE

The Tape-to-Print Dump routine (UTTPPR00) produces a listing of the contents of an entire tape, or of specified portions of a tape. The printout can be in the form of hexadecimal digits or alphanumeric characters or both. The routine operates under control of Job Control in the UNIVAC 9400 Operating System. A series of job control statements define the particular aspects of the routine and direct its execution.

Certain options are available for listing the contents of a tape with nonstandard labels, and for using the system console to determine the kind and range of the listing.

2.4.1. Output Format

A block number (BLOCK n) precedes every block that is printed out. The numbers given are relative to the first block on the tape (BLOCK 1), and options are provided for specifying the first and last blocks to be printed. Each line of the printout begins with a six-digit hexadecimal number which has the following significance: on the first line of a block, it signifies the total byte count of the block; on all succeeding lines it represents the relative hexadecimal position from the start of the block. Two asterisks (**), placed to the right of the hexadecimal byte count, are used to denote that one or more following duplicate lines have been deleted from the printout.

When a tape mark is detected on the tape, the dump routine prints out FILE MARK. Any bad blocks on the tape result in the words UNREADABLE BLOCK and the block number being printed. If two consecutive unreadable blocks are encountered, the routine is terminated. The words UNREADABLE BLOCK and the block number are printed when any unreadable blocks are passed before the first block specified to be printed is reached.

2.4.2. Parameter Statement

The parameter (PARAM) statement includes the following positional parameters: type of printout, first block to be printed, last block to be printed, and a nonstandard label indicator. Normally, the PARAM statement or statements are grouped immediately following the EXEC statement in the job control stream. To identify the Tape-to-Print Dump routine, the routine's code name UTTPPR00 must appear in the EXEC statement.

OPERATION ɓ	OPERAND
// PARAM	$\begin{Bmatrix}\text{A}\\\text{X}\\\text{B}\end{Bmatrix}\left[,\begin{Bmatrix}\text{starting-block-number}\\0\end{Bmatrix}\right]\;[\text{,ending-block-number}]\;[\text{,N}]$

POSITIONAL PARAMETER 1

A	– signifies alphanumeric representation for the printout.
X	– signifies hexadecimal representation for the printout.
B	– signifies both alphanumeric and hexadecimal representation for the printout.

POSITIONAL PARAMETER 2

starting-block-number	– may be one to six decimal digits designating the first block to be printed.
0 or blank	– BLOCK 1 is assumed to be the first block printed.

POSITIONAL PARAMETER 3

ending-block-number	– may be one to six decimal digits designating the last block to be printed.
if blank	– all blocks are printed, if standard labels are used. If nonstandard labels are used, the entire tape is printed (see positional parameter 4).

POSITIONAL PARAMETER 4

N	– specifies nonstandard tape labels. Enables the printout to continue beyond two consecutive tape marks. In this case, a runaway tape condition ends the routine.
if blank	– standard tape labels are assumed and the routine will not print any blocks after the two tape marks.

Example:

```
1        10        20        30        40
// PARAM X,10,20
```

This statement causes the hexadecimal printout of tape blocks 10 through 20, with standard labels assumed.

2.4.3. Job Control Stream

Entry into the Tape-to-Print Dump routine is accomplished by means of the job control stream. A sample sequence of such statements is given below. Parameters are entered into the routine by means of the PARAM statement, as described in the preceding section. For a full description of all job control statements, refer to the *UNIVAC 9400 System Job Control Programmers Reference, UP-7793 (current version).*

Sample job control stream:

```
// JOB jobname
// DVC 3
// LFD PRNTR
// DVC 5
// LFD TAPEIN
// EXEC UTTPPR00
// PARAM X,,10
// PARAM A,11,20
// PARAM X,25,26
/&
```

The use of more than one PARAM statement, as shown in the preceding example, allows the user to print out any desired areas of the tape. The PARAM statements must be placed in ascending order according to the starting block number parameter. Moreover, on any one PARAM statement, the starting block number cannot be greater than the ending block number.

2.4.4. Console Messages

If the parameters are not supplied by means of the job control stream, the following message is printed on the console:

ii UT01 TYPE IN: TYPE(A,X,B), FIRST BLOCK,LAST BLOCK, N.

When the operator replies, these parameters should be separated by commas. For example, if an alphanumeric printout of blocks 11 through 20 is desired, the operator should type in the following:

ii R A,11,20ⓢ

When no additional parameters are to be typed in, the operator should type in ENDⓢ to answer the typeout and terminate the job.

2.4.5. Termination while Processing

To terminate the job at any time, the operator may submit an unsolicited typein. The format is:

iiⓢ, ENDⓢ

2.5. DISC-TO-PRINT DUMP ROUTINE (UNIVAC 8411 SUBSYSTEM)

The Disc-to-Print Dump routine (UTDIPR00) produces a listing of the contents of an entire disc pack, or of specified areas of the disc pack. The listing is printed in hexadecimal digits. The routine normally operates under control of Job Control in the UNIVAC 9400 Operating System. A series of job control statements defines the particular aspects of the routine and directs its execution.

2.5.1. Addressing Scheme

Two five-digit hexadecimal numbers are used to specify the starting and ending addresses of the disc area to be printed. The first three digits of the number define the cylinder (track) number, while the last two digits define the head number. Since the highest cylinder number of a disc pack for the UNIVAC 8411 Subsystem is hexadecimal CA and the highest head number is 9, the leading digit in each case is ignored. The routine makes a test on the remaining digits to determine if they represent a valid disc address. The extra digits permit future expansion.

2.5.2. Parameter Statement

The parameter (PARAM) statement includes the following positional parameters: type of printout, starting address, and ending address. Normally, the PARAM statement or statements are grouped immediately following the EXEC statement in the job control stream. To identify the Disc-to-Print Dump routine, the routine's code name UTDIPR00 must appear in the EXEC statement.

The format of the PARAM statement is:

OPERATION ␢	OPERAND
// PARAM	X,starting-address,ending-address

POSITIONAL PARAMETER 1

X – signifies hexadecimal representation for the printout.

POSITIONAL PARAMETER 2

starting-address – must be five hexadecimal digits, with first three digits indicating cylinder number and last two digits indicating head number.

POSITIONAL PARAMETER 3

ending-address – must be five hexadecimal digits, with first three digits indicating cylinder number and last two digits indicating head number.

NOTE: The leading digit of both the cylinder number and the head number must be a zero.

Example:

```
1        10        20        30        40
// PARAM X,06A00,06A04
```

This statement causes the hexadecimal printout of disc area beginning at address specified by cylinder number 6A, head 0, and ending at address specified by cylinder number 6A, head 4.

2.5.3. Job Control Stream

Entry into the Disc-to-Print Dump routine is accomplished by means of the job control stream. A sample sequence of such statements is given below. Parameters are entered into the routine by means of the PARAM statement, as described in 2.5.2. For a full description of all job control statements, refer to the *UNIVAC 9400 System Job Control Programmers Reference, UP-7793 (current version).*

Sample job control stream:

```
// JOB jobname
// DVC 3
// LFD PRNTR
// DVC 9
// LFD DISKIN
// EXEC UTDIPR00
// PARAM X,06A00,06A04
// PARAM X,00000,00009
// PARAM X,00B02,00C01
/&
```

The use of more than one parameter card, as shown in the above example, allows the user to print out areas of the disc pack by tracks, in any sequence.

2.5.4. Console Messages

The following messages may be printed at the console at various times during the running of the routine:

UT02 RDFCB ERROR

Explanation:

The file control block for the input or output device could not be found.

Action:

The job is terminated.

UT03 UNRECOGNIZABLE TYPEIN

Explanation:

An unsolicited typein was made to the program which was not recognized by UTDIPR00. The typein was ignored.

Action:

If the operator was attempting to end the job he should type in @hh:mm jjⓈ, ENDⓈ. Otherwise no action is required.

2.6. SYSTEM DISC DUMP AND RESTORE ROUTINES

The System Disc Dump and Restore routines consist of two separate routines: the Disc-to-Tape Copy routine (Disc Dump) and the Tape-to-Disc Copy routine (Restore). These routines are used together to copy files from one disc to another by using tape as intermediate storage.

2.6.1. Disc-to-Tape Copy Routine

The Disc-to-Tape Copy routine (UTDCTP00) copies disc files onto a tape. Any number of the files contained on the disc may be copied.

2.6.1.1. COPY Statement

The COPY statement, which is used to submit operational parameters to the Disc-to-Tape Copy routine, are passed to the routine in the form of data cards.

The format of the COPY statement is:

OPERATION ƀ	OPERAND
COPY	{ALL [,$Y$POOL] filename [,ABS}}

POSITIONAL PARAMETER 1

ALL – specifies that all the files of the input disc are to be copied onto tape. If this parameter is used, all of the data records of the files on the input disc are allocated absolutely when UTTPDC00 restores them to the output disc. Only one COPY statement may be present if this parameter is used; no other utility control statements are allowed.

filename – the name of the file to be copied.

POSITIONAL PARAMETER 2

$Y$POOL – used only if positional parameter 1 is ALL. If this parameter is present, all of the extents of the files including $Y$POOL are copied. Without this optional parameter, all the extents of all the files are copied except $Y$POOL; only the first three extents of $Y$POOL are copied. When $Y$POOL is omitted, it must be mapped after running the Restore routine (see 2.6.2) to ensure correct formatting.

ABS —used only when positional parameter 1 is filename. This optional parameter indicates that the file is to be reallocated absolutely; that is, the extents of the file on the output disc, which receives the copied extents, must be in the same physical location as that of the disc which is dumped.

All of the files which are to be made absolute must be specified before those which do not need to be absolute.

The following files must be copied absolutely: $Y$TRAN, $Y$CTRL, $Y$POOL, $Y$ABS, $Y$EXEC, and all indexed-sequential files. Note that if $Y$POOL is specified as one of the files to be copied, only the first three extents are copied.

Examples:

```
   1        10        20        30        40
1.    CØPY ALL,$Y$PØØL
2.    CØPY $Y$PØØL,ABS
```

1. This statement causes all extents of all files on a disc to be dumped on a specified tape.

2. This statement causes the first three extents of the file $Y$POOL to be allocated on the output disc absolutely.

2.6.1.2. Job Control Stream

Entry into the Disc-to-Tape Copy routine is accomplished by means of the job control stream. Two sample sequences of such statements are given below. Parameters are entered into the routine by means of the COPY statement, as described in 2.6.1.1. For a full description of all job control statements, see *UNIVAC 9400 System Job Control Programmers Reference, UP-7793 (current version).*

Example 1:

```
// JOB jobname
// DVC 3
// LFD PRNTR
// DVC 6
// VOL SP0000
// LFD TAPEOT
// DVC 20
// VOL DSP001
// LFD DISKIN
// EXEC UTDCTP00,LOAD$LIB,,REL
/$
   COPY ALL,$Y$POOL
/*
/&
```

The control stream in this example causes all extents of all files on the input disc to be copied onto tape.

Example 2:

```
// JOB jobname
// DVC 3
// LFD PRNTR
// DVC 6
// VOL SP0000
// LFD TAPEOT
// DVC 20
// VOL DSP001
// LFD DISKIN
// EXEC UTDCTP00,LOAD$LIB,,REL
/$
   COPY $Y$TRAN,ABS
   COPY $Y$EXEC,ABS
   COPY $Y$CTRL,ABS
   COPY $Y$POOL,ABS
   COPY $Y$ABS,ABS
   COPY LOAD$LIB
   COPY DATAFILE
/*
/&
```

The control stream in this example causes the files $Y$TRAN, $Y$EXEC, $Y$CTRL, and $Y$ABS to be copied onto tape. These files are allocated on the output disc absolutely. $Y$POOL is also allocated absolutely, but only the first three extents of the file are copied. Files LOAD$LIB and DATAFILE are copied onto tape but when they are copied to the output disc, their physical location may differ.

In both examples, the COPY statements are considered to be data cards and must be preceded by /$ and followed by /*. The VOL statement is required for TAPEOT and DISKIN.

2.6.1.3. Printer Output

Printer output of the Disc-to-Tape Copy routine is produced in two parts:

- A printout of the volume table of contents of the input disc.
- A printout of the control stream. (An error on the COPY statement is indicated by an E printed to the left of the reprinted card image.)

The heading contains the volume serial numbers of the input disc and the tape. The date and time are also printed if the operator sets them.

Four error messages which appear on the console are also shown on the printer. They are:

UC08 FILE ERR filename

UC11 DATA CHECK CC=C'C HH=H'H TRACK NOT COPIED filename

UC14 UTDCTP00 COMPLETED

UC15 UTDCTP00 CANCELLED

See 2.6.1.4. for an explanation of these four messages.

2.6.1.4. Console Messages

The following messages may be printed at the console during the running of the routine:

UC01 FCB ERR

Explanation:

The file control block could not be found.

Action:

Check the control stream. Job is cancelled.

UC02 TAPE I/O ERR

Explanation:

Unrecoverable tape error.

Action:

Rerun job.

UC03 MOUNT NEW TAPEOT, REPLY (R) READY

Explanation:

An additional tape is required to complete copy function.

Action:

Mount the additional tape on the same logical unit as the previous TAPEOT. Type in R as follows: @ hh:mm jjR ⓢ

UC04 NO START/END DATA

Explanation:

A /$ or /* card is missing from control stream.

Action:

If /$ is missing, the job must be rerun.

UC05 TRACK MISSING R0 CC=c_1c_1 HH=h_1h_1

Explanation:

The track descriptor record is missing from the specified track; c_1c_1 is the cylinder number and h_1h_1 is the head number of the missing track descriptor record.

Action:

Job is cancelled.

UC06 NO DISC VOL1

Explanation:

No VOL1 label exists on the disc to be copied.

Action:

Job is cancelled.

UC07 VTOC ERR

Explanation:

An error detected in the volume table of contents has made it impossible to copy the disc.

Action:

The VTOC must be corrected before a copy can be completed.

UC08 FILE ERR filename

Explanation:

Either the filename on the COPY card could not be found on the input disc and processing proceeds to the next COPY card; or an error in the file has caused the termination of the copying of that file.

Action:

Check the filename on the input card against the VTOC print or disc print of file for possible file errors.

UC09 BLOCK SIZE ERR

Explanation:

A record which exceeds 4086 (total number of bytes in the count, key, and data area) has been encountered while using a UNISERVO VI-C Tape Unit as TAPEOT.

Action:

Rerun the job using a UNISERVO 12 Tape Unit.

UC10 LABEL TAPE(S): SELECTOR CHANNEL INPUT REQUIRED

Explanation:

Indicates that the operator should label the output tape with SELECTOR CHANNEL INPUT REQUIRED to ensure that the tape is mounted on a UNISERVO 12 Tape Unit for the UTTPDC00 run.

Action:

Label output tape SELECTOR CHANNEL INPUT REQUIRED.

UC11 DATA CHECK CC=C'c' HH=H'h' TRACK NOT COPIED filename

Explanation:

A data check was encountered at cylinder C'c' and head H'h' in the named file. The track was not copied.

Action:

None.

UC12 VTOC DATA CHECK CC=C'c' HH=H'h'

Explanation:

A data check has been encountered in the area of the volume table of contents at cylinder C'c' and head H'h'.

Action:

This is an unrecoverable error; the job is cancelled.

UC13 MISSING ADDRESS MARKER CC=c_1c_1 HH=h_1h_1

Explanation:

A missing address marker was encountered at cylinder c_1c_1 and head h_1h_1.

Action:

The job is cancelled if this condition was encountered while reading the volume table of contents. If this condition occurs while reading a file, the file processing is terminated and the file is scratched when UTTPDC00 is run.

UC14 UTDCTP00 COMPLETED

Explanation:

Job has terminated under normal conditions.

Action:

None.

UC15 UTDCTP00 CANCELLED

Explanation:

The job has been cancelled.

Action:

None.

UC16 UNEXPIRED TAPE

Explanation:

The data management expiration date has not been met on the TAPEOT. The tape cannot be written on until this condition is changed.

Action:

Rerun UTDCTP00 using a tape which has met the data management expiration date.

UC17 UNFORMATTED TRACK IN VTOC

Explanation:

A track which has not been preformatted is discovered in the VTOC of the input disc. This is a warning message that does not mean that the copy was not completed successfully.

Action:

None.

2.6.2. Tape-to-Disc Copy Routine

The Tape-to-Disc Copy routine (UTTPDC00), also referred to as the Restore routine, restores on disc the contents of a tape containing the output of the Disc-to-Tape Copy (Disc Dump) routine. Four utility control statements are available to the user of the Restore routine: INITIALIZE, COPY, DELETE, and ADD. By selective use of these control statements, the user may effectively reorganize and/or relocate the files he intends to write, or he may copy files from a disc to appear similar to the original except for possible relocation of format records in the VTOC. (Processing is done on a file basis although input/output operations are conducted on the record level.) The utility control statements are passed to the Restore routine on individual data cards.

The Restore routine employs volume checking; therefore, the output disc (DISKOT) must be prepared with a Standard Volume Label (SVL). In addition to volume checking, the Restore routine changes the volume serial number in the SVL, Format 1 label records, and SYSPOOL pool table to reflect the output disc.

The address of the Format 4 label in the SVL and the VTOC extent in the Format 4 record must be identical on the original disc, from which the files are copied, and the current output device. If not, a printer error message occurs and abnormal termination results (see 2.6.2.6).

The Restore routine calls upon the space management allocate, obtain, and scratch transient routines. When the allocate routine is called, the Restore routine prepares an extent request block which always specifies contiguous allocation and absolute or relocating cylinder or track addressing schemes. These addressing schemes depend on the information passed from the Disc Dump routine. Each of the three space management transient routines returns any error conditions to the Restore routine. These error conditions are indicated by a printout of specific alphanumeric codes (see 2.6.2.6).

If DISKIN (disc dump program) was an initial program load (IPL) volume, DISKOT must also be an IPL volume; if DISKIN was not an IPL volume, DISKOT must be the same.

NOTE: Certain system files must be copied to absolute addresses. These files include $Y$TRAN, $Y$ABS, $Y$CTRL, $Y$EXEC, and $Y$POOL. All files which are organized using indexed-sequential methods and employing Format 2 records must be copied to absolute addresses.

If $Y$POOL is one of the files copied, it is necessary to run DACMAP and remap $Y$POOL following UTTPDC00 (see 2.6.1.1).

2.6.2.1. INITIALIZE Statement

This statement causes the entire disc to be scratched and the following records to be written onto disc from tape: IPL1, IPL2, SVL, user volume labels (if present), bootstrap, and Format 4 labels. These records are written to the addresses from which they originated on input disc for the Disc Dump routine. The INITIALIZE statement must be followed by one or more of the other utility control statements (COPY ALL or DELETE statements).

The format of the INITIALIZE statement is:

OPERATION ƀ	OPERAND
INITIALIZE	

No operand is specified.

2.6.2.2. COPY Statement

The COPY statement causes all files put on tape by the prior execution of the Disc Dump routine to be copied onto disc. This statement also permits the user to create a new volume consisting of selected files from TAPEIN.

The format of the COPY statement is:

OPERATION ƀ	OPERAND
COPY	{ ALL filename }

POSITIONAL PARAMETER 1

ALL – specifies that all files put on tape by prior execution of the Disc Dump routine are copied onto disc. Data records for those files designating absolute allocation (specified by input parameters to the Disc-to-Tape Copy routine – see 2.6.1.1) are copied to the same addresses. Those files not designated absolute have their data records written where space is next available. Console message UB14 prints the name of the first file with non-absolute addresses.

NOTE: All files following the first file relocated are non-absolute. Therefore, care should be taken in using a selective copy when executing the dump utility. All files requiring absolute addresses must be copied to tape first.

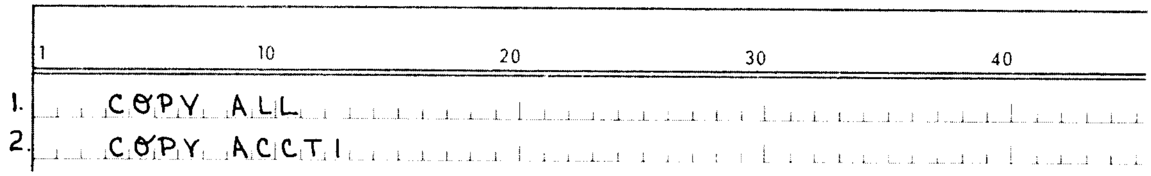
Each file to be copied is first scratched from the output disc (in case there is already a file of the same name on DISKOT). Then each file is allocated space on the output disc. The files on tape are processed sequentially.

The COPY ALL statement may be used with the INITIALIZE statement, but no other.

filename – the name of the file to be copied. This COPY filename statement is used if the user wishes to create a new volume consisting of selected files from TAPEIN. When the filename parameter is specified, the Restore routine scratches the entire output disc volume. TAPEIN is then searched (in a forward direction only) for the specified file; all other files are skipped. When found, the specified file is written to DISKOT, either with or without absolute allocation (as indicated for the Disc Dump routine).

One or more COPY filename data cards may be supplied in the job stream, but only one filename can be specified per data card. Filenames must be ordered as they appear on TAPEIN. The COPY filename statement may not be used with any other.

Examples:



```
 1        10        20        30        40
1.   CØPY ALL
2.   CØPY ACCT1
```

1. This statement causes all files put on tape by the prior execution of the Disc Dump routine to be copied to disc.

2. This statement causes only the file ACCT1 to be copied to disc.

2.6.2.3. DELETE Statement

The DELETE statement causes a file to be scratched from the output disc and allows for all other files on tape to be copied. All files encountered on tape before and after the filename file are copied to disc; the file to be deleted is skipped on tape.

Files to be deleted must be named on separate DELETE data cards, and they must be present in the job stream in the same order as they are filed on tape.

The DELETE statement may be used with the INITIALIZE option, but no other.

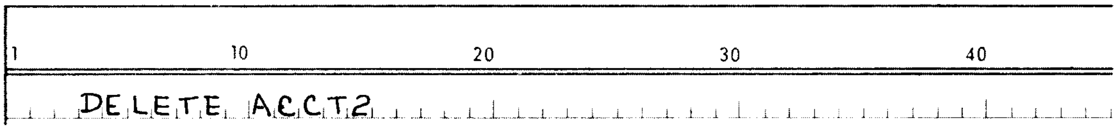
The format of the DELETE statement is:

OPERATION ƀ	OPERAND
DELETE	filename

POSITIONAL PARAMETER 1

filename – the name of the file to be deleted.

Example:



```
 1        10        20        30        40
   DELETE ACCT2
```

This statement functions to prevent ACCT2 from being copied from tape onto disc. All other files are copied to disc.

2.6.2.4. ADD Statement

The ADD statement allows the user to add a file or files from TAPEIN to a disc volume (all other files on TAPEIN are ignored). The file of the same name is scratched from DISKOT, and the file to be added is allocated space based upon the space available in the existent Format 4 and Format 5 records.

Files to be added must be named on separate ADD data cards, and they must be placed in the job stream in the same order as they are filed on tape.

The ADD statement may not be used with any other.

NOTE: When executing the Disc Dump routine, it is important to consider the future use of the ADD function. Since the ADD function works with an existent VTOC, requests for absolute allocation could be rejected. Therefore, it is advisable to ensure that possible ADD files not be given absolute status by the Disc Dump routine.

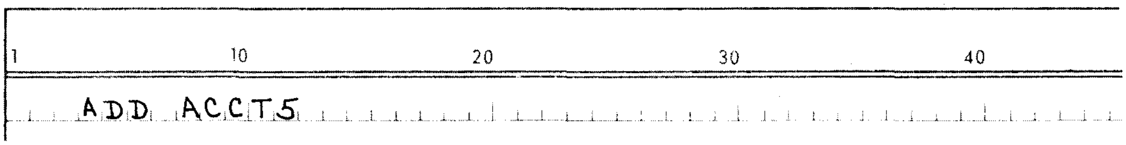
The format of the ADD statement is:

OPERATION ƀ	OPERAND
ADD	filename

POSITIONAL PARAMETER 1

filename – the name of the file to be added.

Example:



```
        ADD  ACCT5
```

This statement causes ACCT5 to be added to a disc volume from tape. All other files on tape are ignored.

2.6.2.5. Job Control Stream

Entry into the Tape-to-Disc Copy routine is accomplished by means of the job control stream. Sample sequences of such statements are given below. Any special parameters are entered into the routine by means of the statement cards, as described previously. For a full description of all job control statements, see *UNIVAC 9400 System Job Control Programmers Reference, UP-7793 (current version).*

Sample job control streams:

Example 1:

```
// JOB jobname
// DVC 3  // LFD PRNTR
// DVC 5
// VOL volume
// LFD TAPEIN
// DVC 20
// VOL volume
// LFD DISKOT
// EXEC UTTPDC00,LOAD$LIB,,REL
```

```
/$
  INITIALIZE
  COPY ALL
/*
/&
```

This example represents a request for INITIALIZE and COPY ALL (see 2.6.2.1 and 2.6.2.2). If both are used, the data cards must appear in the order shown (INITIALIZE followed by COPY ALL).

Example 2:

```
// JOB jobname
// DVC 3 // LFD PRNTR
// DVC 5
// VOL volume
// LFD TAPEIN
// DVC 20
// VOL volume
// LFD DISKOT
// EXEC UTTPDC00,LOAD$LIB,,REL
/$
   ADD file 5
   ADD file 6
/*
/&
```

In this example, assume that TAPEIN contains file 1, file 2, file 5, and file 6, and DISKOT had previously contained file 1, file 2, and file 3. File 5 and file 6 are copied from TAPEIN onto DISKOT. On completion, DISKOT contains file 1, file 2, file 3, file 5, and file 6.

Example 3:

```
// JOB jobname
// DVC 3 // LFD PRNTR
// DVC 5
// VOL volume
// LFD TAPEIN
// DVC 20

// VOL volume

// LFD DISKOT

// EXEC UTTPDC00,LOAD$LIB,,REL

/$

   COPY file 1

   COPY file 7

   COPY file 9

/*

/&
```

Example 3 is a sample of COPY filename. Whatever DISKOT contained formerly would be scratched. File 1, file 7 and file 9 would be copied to DISKOT. All other files on TAPEIN would be ignored.

Example 4:

```
// JOB jobname

// DVC 3 // LFD PRNTR

// DVC 5

// VOL volume

// LFD TAPEIN

// DVC 20

// VOL volume

// LFD DISKOT

// EXEC UTTPDC00,LOAD$LIB,,REL

/$

   INITIALIZE

   DELETE file 2

   DELETE file 4

/*

/&
```

In this example, assume that TAPEIN contains file 1, file 2, file 3, file 4, file 5, and file 6. File 2 and file 4 are deleted when the files are copied onto DISKOT. On completion, DISKOT contains file 1, file 3, file 5, and file 6.

NOTES:

- The utility control statements are data cards; therefore, they must be preceded by /$ and must be followed by /*.
- Logical file definitions (LFD) must be TAPEIN and DISKOT.
- The VOL statement is required for TAPEIN and DISKOT.

2.6.2.6. Printer Output

A printout of the VTOC of DISKOT is produced as it looks before processing has begun. This printout follows the main header. Following the VTOC printout is a subheader. The information printed below the subheader falls into three categories:

(1) Job control stream statements.

(2) Filenames of files on which action was initiated.

(3) One- or two-character code that pertains to the utility control statement or filename. On the printout, the character code is opposite the utility control statement or filename.

The character codes which pertain to utility control statements or filenames are given in Table 2–1. In the table, alphabetic codes refer to specific space management transient routines (A=allocate, O=obtain, S=scratch). Numeric codes indicate error conditions. One-character numeric codes refer to error conditions found during general processing.

CHARACTER CODE	ERROR CONDITIONS
A	Filename file has been allocated space on DISKOT.
A1,S2	Duplicate or invalid file identifier.
A2	No room in VTOC.
A3,S3	Input output error.
A4	Invalid physical unit block or volume serial number.
A5	Cannot allocate absolute request.
A6	Contiguous request cannot be assigned.
A7	More than 16 extents.
A8	No room on disc.
01,04	Volume serial number error.
02	File identifier not found or illegal identifier
1*	Error on the utility control statement.
2*	A second reel of the same volume was expected, and the READY (R) reply to console message UB05 was not given (see 2 6 2.7).
3*	This file was not copied to disc because an error was found during the attempt to copy the file from disc to tape.

*When UTTPDC00 terminates, a legend with an abbreviated explanation of these three codes is produced on the printer.

Table 2–1. Restore Routine, Error Conditions

Four messages are included in printer output:

RE-PREP DISKOT TO MATCH DISKIN OR DO NOT INITIALIZE

- In SVL, occurs when the address of Format 4 on DISKOT does not correspond to the address of Format 4 on TAPEIN.
- In Format 4, VTOC extent on DISKOT does not match VTOC extent from TAPEIN.

END OF TAPE. CARD IGNORED.

Last file on tape has been passed, and the next data card in the job control stream was not /*. Explanation is:

- Data cards are not sequenced in the same order as the files on TAPEIN.
- Data card filename does not match any filename on TAPEIN.

UTTPDC00 COMPLETE

Normal termination occurs.

UTTPDC00 ABNORMAL TERMINATION

Before terminating, a printout of the VTOC of DISKOT is produced as it looks when processing was ended. If the termination is normal, the printout is followed by the message UTTPDC00 COMPLETE; if the job step is cancelled due to unrecoverable error conditions, the printout is followed by the message UTTPDC00 ABNORMAL TERMINATION.

2.6.2.7. Console Messages

The following messages may be printed at the console during the running of the Tape-to-Disc Copy routine:

UB01 NO FCB

Explanation:

The file control block could not be found.

Action:

Check the control stream for all devices (PRNTR, TAPEIN, and DISKOT).

UB03 NO VOL1

Explanation:

Volume 1 label (SVL) could not be found on output disc.

Action:

Ensure that the disc has been properly initialized.

UB04 TRACK OVERRUN

Explanation:

The complete record cannot be written within track boundaries.

Action:

Unrecoverable.

UB05 DISC ERR filename

Explanation:

An unrecoverable disc error has occurred.

Action:

Unrecoverable.

UB06 TAPE FORMAT ERROR

Explanation:

An error exists in the format of input tape (including block serial number errors).

Action:

Unrecoverable. Tape-to-print of input tape suggested.

UB08 DISMOUNT TAPEIN + MOUNT NEXT REPLY (R) READY

Explanation:

This message is printed on the console when the input is continued on another reel.

Action:

Tape (second reel) should be mounted on same logical unit. The message is printed out a second time if R is not typed in. If R is not typed in on the second try, the filename is written to the printer with an incomplete volume error code and the job terminates.

UB09 END OF JOB

Explanation:

Normal job termination.

Action:

None.

UB10 NO START/END DATA

Explanation:

Either a /$ or /* card is missing.

Action:

If /$ is missing, the job must be restarted with the /$ included. (See sample control stream.)

UB11 INPUT INCOMPLETE JOB TERMINATED

Explanation:

The disc-to-tape copy routine (UTDCTP00) could not be completed and the information on tape is incomplete.

Action:

Rerun UTDCTP00.

UB13 DELETED: filename

Explanation:

The named file has been deleted from the output disc.

Action:

None.

UB14 VSN ERR

Explanation:

Volume serial number error has been detected.

Action:

Job terminates. Check // VOL card and VSN on output disc.

2.7. SYSTEM MODULE PATCH ROUTINE

The System Module Patch routine (UTPTCH00) copies a system tape into a maximum of four other tapes. While copying, names of specified modules can be changed, and patch corrections can be applied to load, object, and transient modules. Concurrently, the printer produces a listing of both the header blocks of all the modules as they appear on the new tapes and the contents of the utility control statements at the places on the new tapes where they have been applied. The routine operates under Job Control in the UNIVAC 9400 Operating System. A series of job control and utility control statements defines the particular aspects of the routine and directs its execution.

Utility control statements recognized by UTPTCH00 are NAME, PATCH, and ENTRY. All statements are freeform through column 71, but the operation code (NAME, PATCH, or ENTRY) must be separated from any following parameters by at least one space.

2.7.1. NAME Statement

The NAME statement identifies modules and their file type so that patch corrections (see 2.7.2) or transfer address changes (see 2.7.3) can be applied. If required, the NAME statement also allows the name or version number of a specified module to be changed.

The format of the NAME statement is:

OPERATION ƀ	OPERAND
NAME	module-name,t $\left[, \begin{Bmatrix} \text{new-module-name} \\ \text{VER=new-version-number} \end{Bmatrix} \right]$

POSITIONAL PARAMETER 1

module-name – specifies the present module name, which can contain a maximum of eight alphanumeric characters. If less than eight alphanumeric characters are given, module name is filled with either trailing zeros for load or transient modules or with trailing spaces for other module types.

POSITIONAL PARAMETER 2

t – specifies a single letter which indicates the type of module identified. The single letter codes are:

L = load module

O = object module

S = source module

C = copy module

P = proc module

T = transient routine

NOTE: Any routine in transient form can be specified.

POSITIONAL PARAMETER 3

new-module-name – specifies the new name of the module, if a name change is required.

VER=new-version-number – specifies a new version number of the module and must be four decimal digits.

if blank – module name and version number are not changed.

Examples:

```
  1        10                  20                  30                  40
1. NAME    ABCD,L
2. NAME    EFG,L,VER=0102
3. NAME    PQRS,S,XYZ
```

1. This statement names load module ABCD.

2. This statement names load module EFG and specifies a new version number (102).

3. The source module named PQRS is renamed XYZ.

2.7.2. PATCH Statement

The PATCH statement is used while copying a system tape to make editing corrections to object, load, and transient modules.

The format of the PATCH statement is:

OPERATION ƀ	OPERAND
PATCH	address[/esid], correction[/esid]

POSITIONAL PARAMETER 1

address – specifies in hexadecimal the address of the first byte where the PATCH correction is to be applied. For load and object modules, its value is obtained from the assembly listing, and, if linked to other modules, its value is adjusted by the value of the assigned address obtained from the linker map. For transient modules, its value is the relative address from the start of the routine.

/esid – indicates the external symbol identification obtained from the assembly listing of the control section containing the text to be modified. This parameter is required in object module patches and is specified as two hexadecimal digits. If omitted, esid is assumed to be zero.

POSITIONAL PARAMETER 2

correction – indicates the patch correction in the form C'nn...' for a character patch, X'nn...' or nn... for a hexadecimal patch. When indicating a patch in hexadecimal, an even number of hexadecimal digits must be specified.

/esid – indicates the external symbol identification obtained from the assembly listing of the control section containing the text to be modified. This parameter is specified as two hexadecimal digits. Address constants and external references must be followed by an esid number. Address constants must be either a fullword or a halfword. V-type (external address) constants cannot be patched.

Examples:

```
  1        10        20        30        40
1. NAME   EFG,L
   PATCH 5324/03,00002678/03
2. NAME   GEHI,Ø
   PATCH 522B/0E,C'H'
```

1. The hexadecimal value 00002678 is patched into the load module EFG. The patch begins at hexadecimal location 5324.

2. The one-byte character constant H is patched into the object module GEHI at location 522B.

NOTE: User comments can appear on PATCH statements after the first space which follows the correction data. These comments are ignored by the patch routine, but they appear on the printer listing.

2.7.3. ENTRY Statement

The ENTRY statement is used to change the transfer address of a load or object module.

The format of the ENTRY statement is:

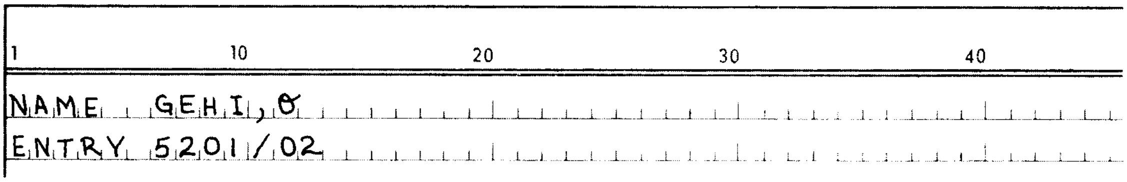
OPERATION ƀ	OPERAND
ENTRY	address[/esid]

POSITIONAL PARAMETER 1

address – specifies the new entry or transfer address, expressed as an even number of hexadecimal digits, by inserting leading zeros if necessary.

/esid – indicates the external symbol identification obtained from the assembly listing of the control section containing the text to be modified. This parameter is specified as two hexadecimal digits. If omitted, esid remains unchanged.

Example:



```
1        10         20         30         40
NAME  GEHI,0
ENTRY 5201/02
```

The entry address for the object module GEHI is now at hexadecimal location 5201.

2.7.4. LFD File Definitions

The logical filenames used by the UTPTCH00 routine and defined in the control stream at execution time are PRNTR for the printer, TAPEIN for the input tape, and TAPEOT01 for the output tape. If it is desired to copy more than one tape, the filenames which correspond to the additional tapes are TAPEOT02, TAPEOT03, and TAPEOT04.

2.7.5. Control Stream

Entry into the System Module Patch routine is accomplished by means of statements in a control stream. A sample sequence of such statements is given below. Any special parameters are entered into the routine by means of the utility control statements described in 2.7.1, 2.7.2, and 2.7.3. For a full description of the job control statements, see *UNIVAC 9400 System Job Control Programmers Reference, UP-7793 (current version).*

Sample control stream and UTPTCH00 statements:

```
// JOB jobname
// DVC 3
// LFD PRNTR
// DVC 5
// LFD SYSRES
// DVC 6
// LFD TAPEIN
// DVC 7
// LFD TAPEOT01
// OPTION NOVOL
// EXEC UTPTCH00,LOAD$LIB,,REL
/$
   NAME ABCD,L
   PATCH 6C34/56,89AB
   NAME EFG,L,VER=0102
   PATCH 5324/03,00002678/03
   NAME GEHI,O
   ENTRY 5201/02
   PATCH 522B/0E,C'H'
   PATCH 54E2/0E,X'89AB'
   NAME PQRS,5,XYZ
/*
/&
```

NOTE: NAME control cards must appear in the control stream in the same order as the corresponding modules appear on tape.

By inserting job control statements to set UPSI switch indicators, the patch routine can be modified as follows:

- To patch an OBJFIL tape, insert:

  // SET UPSI,1

- To omit replacing transient loaders before load modules, insert:

  // SET UPSI,X1

  where X indicates a bit that remains unchanged.

- To forward-space file past labels on input and output OBJFIL tapes, insert:

  // SET UPSI,1X1

  where X indicates a bit that remains unchanged.

2.7.6. Console Messages

Four error messages may be printed at the console, each of which is capable of terminating the running of the routine. A description of each of these messages follows:

UA02 UNDEF RDFCB

Explanation:

One of the necessary I/O devices required by the program has not been defined in an LFD statement.

Action:

Correct the job control stream and rerun.

UA03 TP BLCK NO ERR

Explanation:

The sequence check of block numbers from the input tape has found a block number which is not one greater than the previous block number.

Action:

Tape-to-print listing of the input tape should show where the block sequence number error occured.

UA04 MODULE FRMT ERR

Explanation:

The transfer record of a load or object module which is to be patched cannot be found.

Action:

The module to be patched is probably not in standard system format. Take a tape-to-print of input tape to verify.

UA05 NO TRNS LDRS

Explanation:

Transient tape loader routine (ASM00006), which is to be written to the tape before each load module, cannot be located.

Action:

If no prefix loader is required, SET UPSI,X1 and repeat. DAPS listing of input tape shows whether transient routine ASM00006 is available on the tape.

UNIVAC

SPERRY RAND