70/35-45-5

TAPE OPERATING SYSTEM (TOS)

UTILITY ROUTINES

RCA
Information
Systems

SPECTRA 70

70/35-45-55 TAPE OPERATING SYSTEM (TOS)

Utility Routines

RCA
Information
Systems

SPECTRA 70

70/35-45-55 TAPE OPERATING SYSTEM (TOS)

Utility Routines

70-35-302
November 1969

The information contained herein is subject to change without notice. Revisions may be issued to advise of such additions or corrections.

First Printing: December 1966
Reissued: May 1967
Edition: July 1967
Edition: March 1968
Edition: December 1968
Edition: July 1969

The July 1967 edition includes revision 1.

The March 1968 edition includes revisions 1 through 3.

The December 1968 edition includes revisions 1 through 6.

The July 1969 edition includes revisions 1 through 7.

The following revision has been issued to the July 1969 edition:
Revision 8: November 1969 (70-35-302-1)

Revision 8 includes all available information relating to software release version 016.

CONTENTS

CONTENTS
(Cont'd)

INTRODUCTION

◆ The TOS Utility System provides the TOS user with an integrated set of routines designed to relieve programming effort and to simplify his testing and production operations. In this regard, three major types of routines are provided: Peripheral Conversion, Diagnostics, and System Maintenance.

The peripheral conversion routines offer a ready means of converting data from one medium to another -- punched cards to magnetic tape, random access to random access, tape to printer, etc.

In the diagnostic area a variety of routines are supplied to assist the programmer in his program testing -- memory dumps, tape edits, and so forth. Also included in this package is a powerful diagnostic tool called the Automatic Integrated Debugging System (AIDS). AIDS gives TOS users a completely automatic testing capability not previously available for second-generation programming systems.

To facilitate library maintenance, routines are provided that can be used to create and maintain the following TOS libraries: the System Load Library (SLLT), the Call Library (CLT), and the Program Load Library (PLLT). The System Load Library is used to store control programs, language translators, and other RCA-supplied system routines (It may also contain the installations production programs). The Call Library contains assembly macros, FORTRAN macros, COBOL library statements, and object modules used by the language translators and the Linkage Editor to produce executable TOS programs. The Program Load Library contains those production programs loaded and executed by the Executive and Monitor systems.

The utility routines that comprise this system are described in Sections 2 through 7 of this manual, categorized as follows:

2. Peripheral Conversion
3. Peripheral Conversion-Random Access
4. Diagnostics
5. Diagnostics-Random Access
6. System Maintenance
7. System Maintenance-Random Access

In addition, a number of appendices are included. These sections can aid the reader in performing service operations on various library tapes, modifying peripheral conversion routines, determining memory requirements for individual routines, etc.

INTRODUCTION
(Cont'd)

The reader also is advised to consult the following publications related to the use and understanding of the utility programming system:

Spectra 70 Systems Standards, 70-00-610

TOS Control System, 70-00-609

TOS Operators' Guide, 70-35-403

TOS Sort/Merge System, 70-35-303

Spectra 70 Random Access Techniques 79-05-003

1. CONSIDERATIONS FOR USE

OPERATING PROCEDURES AND CONVENTIONS

Equipment Configurations

◆ The required equipment section for each routine described in this manual includes only the minimum equipment complement to execute the routine. It does not include the system resident device (SYSRES), which is always required, or an alternate program library if one is used.

The optional equipment section includes additional or alternate devices that can be used with the routine. If a magnetic tape alternate is used for parameter input, a card-to-tape facility must be available at the installation; if magnetic tape is used for printer output, a tape-to-printer facility must be available.

Loading of Utility Routines

All utility routines except Snapshot and the self-loading routines (Self-Loading Memory Print, Self-Loading Tape Edit, and Self-Loading Random Access Edit) are loaded by the Executive or Monitor as described in the Operators' Guide.

The routine name required in the load statement appears in parenthesis following the title of the routine. For example, the Card to Tape (CDTP) routine can be called into memory by the following console message:

```
EΔLODΔCDTP.
```

Device Assignments

◆ Each routine includes a section that lists the symbolic names of the devices used, the device types, and their uses. Device assignments are given for operating under both Executive and/or Monitor, depending on the nature of the routine.

Executive Control Operation

Device assignments for routines that run under Executive control are made using run-time parameter cards (//ΔASSGN), or by means of the console typewriter in response to Executive assign device messages. These procedures are described in detail in the Operators' Guide.

Monitor Control Operation

When running under Monitor control, device assignments are also made using //ΔASSGN cards or the console typewriter, with the exception of the parameter input device.

The parameter input device for Monitor operation is always SYSIPT, which may be a card reader or magnetic tape. As this device is assigned by the device mnemonic used in the executive load Monitor request, no parameter or console message is required.

MEMORY REQUIREMENTS

Memory is normally assigned to a program at load time based on the memory size requirements appearing in its program descriptor block. This size is generally sufficient for both the program and the input/output areas.

If the allocated input/output areas for a particular program are not large enough for the current application, the size of these areas may be increased. This can be accomplished at load time by using the mmmmmm option of the EΔLOD console request. Or, the Linkage Editor routine may be used with additional memory allocated through the use of the PROG parameter card.

The memory requirements for the utility routines supplied for TOS are listed in Appendix D.

LABEL CONVENTIONS

General

Most utility routines that write to or read from magnetic tape are designed to process tapes recorded in the following standard Spectra label* format:

$$\text{VOL1} \quad \text{HDR1} \quad \text{TM} \quad \text{data} \quad \text{TM} \quad \left\{\begin{matrix}\text{EOV}\\ \text{EOF}\end{matrix}\right\} \quad \text{TM} \quad \text{TM}$$

Operating procedures for these routines are greatly simplified, therefore, when the format described above is used as the installation standard. Furthermore, it is recommended that the Tape Volume Initializer routine be used to prepare all tapes with VOL and HDR labels before they are released for operations use.

Other label conventions can, of course, be used. However, in this case own-coding is often required to generate and process the labels. Often, operators may be required to "force" purge-date and other label checking functions that have been designed into the utility routines to provide data protection facilities for the installation.

Refer to the label processing section of Appendix A for own-coding considerations if nonstandard label processing is desired.

Peripheral Conversion Routines

◆ The peripheral conversion routines in Section 2 are preset to process standard labels. These routines therefore require that information used to generate and check labels be supplied by VOL and TPLAB run-time parameter cards. (See Section 2 of the Operators' Guide for the format of these cards.)

The programmer must know the name used in the DTF statement when the routine was assembled. This name is required as the "filename" entry in the VOL card.

---

*For a description of these labels refer to the Systems Standards Reference Manual, 70-00-610.

Peripheral Conversion Routines *(Cont'd)*

Table 1-1. DTF Names Used for Peripheral Conversion Routines

Type of File	Name
Magnetic tape input.	ITUTP2F
Magnetic tape output.	ITUTP3F

For seven-level input tapes, labels must be recorded in even-parity, translate-on mode (48, 88, or C8); output labels are always generated in this manner.

Peripheral Conversion Routines - Random Access

◆ The random access conversion routines in Section 3 are also preset to process standard labels.

For magnetic tape input and output files, the conventions outlined above apply. For random access devices, label information must be supplied by a VDC run-time parameter card. (See Section 2 of the Operators' Guide for the format of this card.)

As with the VOL card, it is necessary for the programmer to know the name used in the DTF statement for the routine. This name is then used as the "filename" entry in the VDC card.

Table 1-2. DTF Names Used for Random Access Peripheral Conversion Routines

Type of File	Name
Magnetic tape input.	ITUTP2F
Magnetic tape output.	ITUTP3F
Random Access input.	ITUTP4F
Random Access output.	ITUTP5F

RANDOM ACCESS CONVENTIONS

◆ Before a random access volume can be processed by one of the peripheral conversion routines, the volume must first be initialized and at least one file allocated. These operations are performed by the RAINIT and RAALLR utility routines described in Sections 3 and 7, respectively.

It is also necessary that the serial number of any volume to be processed appears in the on-line catalog maintained by the Executive. The entering of serial numbers in this catalog is effected by the E∆OLC console routine. See Section 3, Executive Console Routines, in the Operators' Guide.

All random-access peripheral conversion routines also require a Volume Displacement Card (VDC) at run time to identify the serial number of the volume to be processed. The format of this parameter is described in Section 2 of the Operators' Guide. The VDC "filename" entry for each routine is listed in the Considerations for Use section of each routine.

RANDOM ACCESS CONVENTIONS (Cont'd)

In order to generate a random access record that contains a separate Key field, the Utility-Modifier and Field Select parameters must be supplied.

When using any of the random-access peripheral conversion routines, the first track of each file is unavailable for data storage. This track is automatically reserved by these routines for storing user header and trailer labels, whether or not labels are used for the file.

PAPER TAPE CONVENTIONS

◆ Paper tape may be substituted for a "card" input device or a "card" output device in the following peripheral conversion routines:

Card to Tape (CDTP)

Selective Card to Printer and/or Punch (CDPR)

Selective Tape to Printer and/or Punch (TPPR)

Card to Random Access (CDRA)

Random Access to Printer and/or Punch (RAPR)

Input and Output Standards

The following standards apply whenever paper tape is used as the input or output medium for the routines mentioned above.

Input

◆ Gapped or gapless tape may be processed.

The input code may be 5-, 6-, 7-, or 8-level. (See Tape Translation section.)

No checking is made for labels. If this is desired, user own-coding must be employed.

Records may be fixed-length (blocked or unblocked), or of undefined length.

The maximum size of an input block (or record) is 4,096 characters.

The last record must contain /* in its first two positions to denote the end of the input file.

Output

◆ Gapped output is always produced.

The output code may be 5-, 6-, 7-, or 8-level. (See Paper Tape Translation section.)

Labels are not generated. If this is desired, user own-coding must be employed.

Records may be fixed-length (blocked or unblocked), or of undefined length.

The maximum size of an output block (or record) is 4,096 characters.

The last record punched is a /* record, which denotes the end of the input file.

Paper Tape Translation

◆ Translation is not required for eight-level input tapes and eight-level output tapes. It is also not required when paper tape is being copied, processing of data is not specified, and input and output tapes are the same level.

For all other conditions, the translation of input or output characters into their EBCDIC equivalents must be accommodated by the hardware or the user's program.

Routine Parameters

◆ Except for the Utility Modifier parameter, all routine parameters are the same. In respect to the Utility Modifier parameter, the F, A, and B entries must be specified as follows. (All other entries in this parameter remain unchanged.)

Routine	Entry	Meaning
CDTP CDPR TPPR CDRA RAPR	, Fx	Input record format: x=F fixed-length (preset). =U undefined. Note: Preset value is F for CDTP, CDPR, and CDRA; and U for TPPR and RAPR. Variable-length paper tape, magnetic tape, or random access records must be described as undefined.
CDTP CDPR TPPR CDRA RAPR	, A=(g) or , A=(n, m) or , A=(K=x, D=y)	Input format: Undefined records: g = maximum size of input blocks. Fixed-length records: n = record size m = block size Fixed-length with keys (RA only): x = size of Key field y = size of Data field Note: Preset values are 80, 80 for CDTR, CDPR, and CDRA. Preset value is 1000 for TPPR and RAPR. The maximum record or block sizes are: 4,096 (magnetic tape and paper tape), 3600 (disc), 3,300 (drum), and 2,048 (mass storage).

Routine Parameters *(Cont'd)*

Routine	Entry	Meaning
CDTP CDPR TPPR CDRA RAPR	, B=(h) or , B=(a, b) or , B=(K=x, D=y)	Output format: Undefined records: h = maximum size of output blocks. Fixed-length records: a = record size b = block size Fixed-length with keys (RA only): x = size of Key field y = size of Data field Note: For printing, the preset value is 132; for punching, the preset values are 80, 80; for magnetic tape and random access, the preset value is 1000. The maximum record or block sizes are: 4,096 (magnetic tape and paper tape), 3,600 (disc), 3,000 (drum), and 2,048 (mass storage).

PACKING AND UNPACKING OF DECIMAL FIELDS

When the Field-Select parameter is used with peripheral conversion routines to pack or unpack decimal fields, care must be exercised when stating the sizes of the input and output fields. The following review of the format of packed and unpacked fields is given here to help the reader calculate the correct values for the "n" and "m" entries in the field-select card.

Unpacked Decimal Fields

◆ The rightmost byte of a decimal field contains the sign of the field in its high-order four bits. All other bytes in the field contain 1 bits in their four high-order positions. (These high-order bits are called the zone position of the byte; the four low-order bits are called the numeric portion of the byte.) Thus, a seven-byte decimal field, in unpacked format, appears as shown below:

Z	N	Z	N	Z	N	Z	N	Z	N	Z	N	S	N

where:

Z = zone portion

N = numeric portion

S = sign

Packed Decimal Fields

◆ In packed format, each byte contains two decimal digits except for the least significant byte, which contains one digit and a four-bit sign. For example:

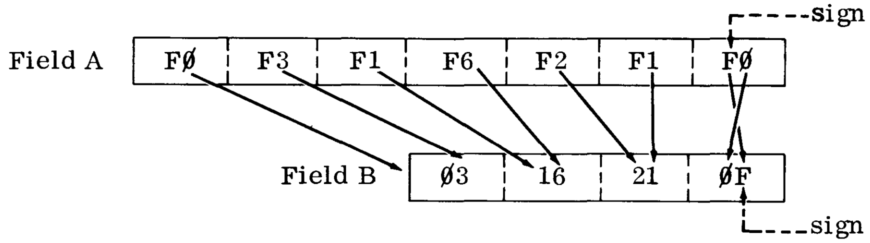
N	N	N	N	N	N	N	S

where:

N = numeric digit

S = sign

The following illustration shows how a seven-byte input field would be packed into a four-byte output field if (P, 7, 4) were specified in the Field-Select parameter:



The above process would be reversed if Field B were the input field and (U, 4, 7) had been specified.

2. PERIPHERAL CONVERSION

TAPE VOLUME INITIALIZER (TPINIT)

General Description

◆ The Tape Volume Initializer routine prepares reels of magnetic tape for label processing by writing up to eight volume labels, a dummy HDR1 label,* and a tape mark on each tape initialized. All labels are written as 80-character blocks.

Up to 16 tape reels (UTV001 through UTV016) may be initialized in 1 cycle of this routine. Additional cycles may be executed if more than one reel is to be initialized and sufficient tape drives are not available.

Preset Functions

None.

Optional Functions

Initialize up to 16 tapes with 1 to 8 volume labels, a dummy header label, and a tape mark.

Input

◆ Input to this routine can consist of a single control parameter to generate single, standard volume labels; or it may consist of a control parameter, a set of volume label image parameters, and an end volume parameter when user-supplied volume labels are desired.

Output

◆ Output of this routine consists of up to 16 magnetic tapes per cycle, with each tape containing up to 8 volume labels, a dummy header label, and a tape mark. In addition, a log of the volume labels written to the output tapes is typed on the console typewriter.

Equipment Configuration

Required

◆ Processor (65K)

Console typewriter.

Card reader, or Videoscan document reader with card read feature.

Magnetic tape device.

Optional

◆ Up to 15 additional magnetic tape output devices may be used.

---

*The dummy header label contains HDR1 in positions 1-4 and hexadecimal spaces in all other positions.

Routine Parameters - General

◆ To generate standard volume labels, only a Utility Control parameter is required; to create user-supplied or multiple volume labels, the following are required:

A Utility Control parameter.

A set of Volume Label Image parameters (one to eight) for each output tape.

An End Volume Label Image parameter for each set of volume labels.

An End parameter.

Routine Parameters - Detailed (Standard Label Generation)

Utility Control Parameter

◆ *Format:*

ΔUΔTnn,Nnn,SERIAL = (nnnnnn),CODE = (nnnnnnnnnn),P,REWIND

Entry	Explanation
ΔUΔ	Parameter identifier.
Tnn	nn = number of tape devices to be used during initialization (01-16).
,Nnn	nn = total number of reels to be initialized (01-99).
,SERIAL = (nnnnnn)	nnnnnn = six numeric digits representing the volume serial number assigned to the first reel to be labeled. This number is automatically incremented by 1 for each volume label written to a succeeding tape.
,CODE = (nnnnnnnnnn)	nnnnnnnnnn = 10 alphanumeric characters that identify the owner of the volume.
,P	(Optional) When used, the security field in each volume label contains the value 1. If omitted, the security value will be 0.
,REWIND	(Optional) When this entry is used, all output reels are rewound to BOT after label initialization, and the routine terminates. If omitted, the routine requests that a new reel be mounted on each output device after a complete cycle has been executed. The next set of labels will then be written to UTV001, UTV002, etc. This procedure is repeated until the reels specified by Nnn have been initialized.

Utility Control Parameter (Cont'd)

Note:

Entries following ΔUΔ may appear in any order.

Examples:

ΔUΔT03,N03,SERIAL = (000365),CODE = (RCA-EDP857),P,REWIND

ΔUΔT05,N12,SERIAL = (000001),CODE = (CREDITΔΔΔΔ)

Routine Parameters - Detailed (Nonstandard Label Generation)

Utility Control Parameter

◆ *Format:*

ΔUΔCARD,Tnn,Nnn,REWIND

Entry	Explanation
ΔUΔ	Parameter identifier.
CARD	Indicates that user-supplied labels are provided.
,Tnn	nn = number of tape stations to be used during initialization (01-16).
,Nnn	nn = total number of reels to be initialized (01-99).
,REWIND	(Optional) When this entry is used, all output reels are rewound to BOT after label initialization, and the routine terminates. If omitted, the routine assumes that a new reel has been mounted on each output tape device, and the next set of labels are written to UTV001, UTV002, etc., until the operator terminates the job.

Examples:

ΔUΔCARD,T02,N02

ΔUΔCARD,T05,N10

Volume Label Parameters

◆ *Format:*

Card Columns	Content	Meaning
1-3	VOL	Label identifier.
4	1	Volume label sequence number.
5-10	nnnnnn	Volume serial number (six digits) which uniquely identifies the tape volume.
11	n	Volume security indicator; n = 0 or 1. 0 - volume has no security and may be used. 1 - volume has security and may not be used.
12-41		Reserved. Leave blank.
42-51	identifier	10 alphanumeric characters that identify the owner of the volume.
52-80		Reserved. Leave blank.

The first volume label for an output reel must adhere to the format shown above; succeeding volume labels for the same reel have the following format:

Card Columns	Contents
1-3	VOL
4	2-8
5-80	Undefined. May be assigned by the programmer.

End Volume Set Parameter

◆ *Format:*

ΔENDV

An ENDV parameter must be used to indicate the end of each set of Volume Label Image card(s). See example on page 2-6.

END Parameter

◆ *Format:*

ΔEND

This parameter is mandatory and must be used to indicate the end of parameter input when user-supplied volume labels are supplied.

Device Assignments

◆ *Under Executive Control*

SDN	Device Type	Remarks
UTV001	Magnetic tape.	First reel to be initialized.
UTV002	Magnetic tape.	Second reel to be initialized.
•		
•		
UTV016	Magnetic tape.	Last reel to be initialized.
UTVPRM	Card reader.	Parameter input.

Under Monitor Control

SDN	Device Type	Remarks
UTV001	Magnetic tape.	First reel to be initialized.
UTV002	Magnetic tape.	Second reel to be initialized.
•		
•		
UTV016	Magnetic tape.	Last reel to be initialized.
SYSIPT	Card reader.	Parameter input.

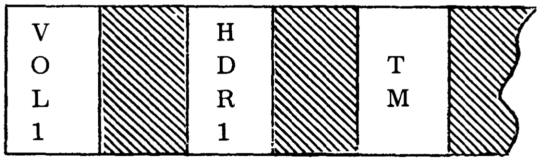
Parameter Examples

◆ 1. *Standard Volume Labels*

A. Five magnetic tape reels are to be initialized in one cycle of this routine. All tapes are to be rewound to BOT after being labeled. The volume serial number assigned to the first reel will be 000002; all reels will have a security field value of 1 and an owner-identifier code of RCA-EDP857.

```
ΔUΔT05,N05,SERIAL = (000002),CODE = (RCA-EDP857),P,REWIND
```

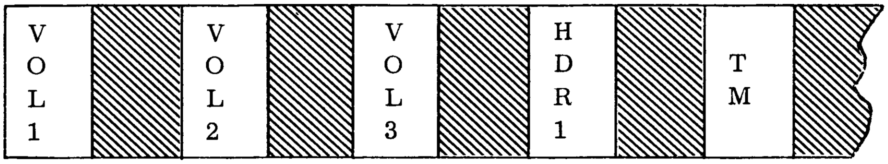
Output Tapes



B. Six magnetic tape reels are to be initialized in two cycles of this routine (three reels per cycle). All tapes are to be rewound and unloaded. The volume serial number assigned to the first reel is 000001. Security fields contain 0 and the owner-identifier code is ACCOUNTING.

Parameter Examples *(Cont'd)*

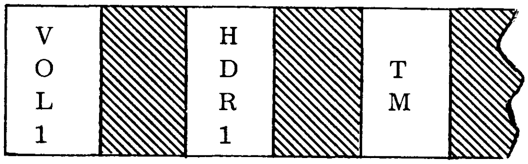
UTV003:



Parameter Examples
(Cont'd)

```
ΔUΔT03,N06,SERIAL=(000001),CODE=(ACCOUNTING)
```

Output Tapes



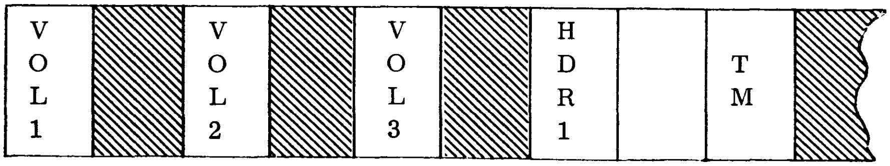
2. *User-Supplied Volume Labels*

Three magnetic tapes are to be initialized in one cycle of this routine. All tapes are to be rewound and unloaded.

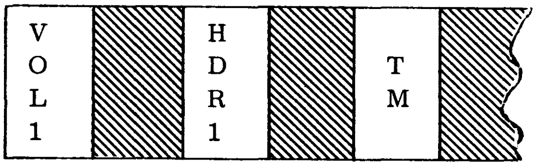
```
ΔUΔCARD,T03,N03
VOL1 ...
VOL2 ...       label set for tape UTV001
VOL3 ...
ΔENDV
VOL1 ...       label set for tape UTV002
ΔENDV
VOL1 ...
VOL2 ...
VOL3 ...       label set for tape UTV003
ΔENDV
ΔEND
```

Output Tapes:

UTV001:



UTV002:



CARD TO TAPE (CDTP)

General Description

◆ The Card to Tape routine transcribes 80-column card records or paper tape to magnetic tape in standard Spectra 70 format. Input cards are punched in EBCDIC format, with the final card containing /* in the first two columns to signify the end of the file. The generated output file contains standard Spectra 70 labels and may be single or multivolume.

Preset Functions

This routine is preset for the following functions:

To copy 80-character records (EBCDIC format) to magnetic tape in unblocked format.

To position output tapes at BOT at the start and end of transcription.

To alternate tapes for multivolume output and when two output devices are available.

To write standard header and trailer labels* on the output tapes, with a double tape mark terminating each reel.

To type out record and block counts at the end of the routine.

To provide a console typewriter listing (log) of the input parameters.

Optional Functions

The following optional functions are provided and may be specified by the use of parameters:

Blocking of output records up to a maximum of 4,096 characters per block.

Field selecting, packing, and unpacking of input fields.

Positioning and disposition of the output tapes.

Sequence checking of the input file.

Suppressing the logging of input parameters.

Input

◆ Input to the Card to Tape routine consists of a card file to be transcribed to magnetic tape. The last card of this file must contain /* in the first two columns. If a nonblank is in Column 80, the /* is treated as data.

When optional functions are desired, the parameters that specify these options must be entered before the first input record is read.

---

*Label information is provided by VOL and TPLAB run time parameter cards. For processing of user labels, see Modification of Peripheral Routines, Appendix A.

Output

◆ Output consists of the contents of the card input file transcribed to tape in the format specified by the parameters. The output file may be unlabeled or may contain standard Spectra 70 labels; it may be single or multivolume.

Record and block counts are typed out at the end of this routine. A log of input parameters may also be provided.

Equipment Configuration

Required

◆ Processor (65K).

Console typewriter.

Card reader, or Videoscan document reader with card read feature.

Magnetic tape device.

Optional

◆ An additional magnetic tape device can be assigned for use as an alternate output. A paper tape reader can be used instead of the card reader. Refer to page 1-4 for paper tape considerations.

Routine Parameters - General

◆ The following three parameter cards are used with this routine:

Utility-Modifier Parameter

This parameter specifies blocking of output records; field selection; sequence checking of the input cards; positioning and disposition of output tapes; logging of input parameters; and creation of unlabeled output tapes.

Field-Select Parameter(s)

This parameter specifies field selection, packing, and unpacking of fields in the input records.

This parameter must be used when the field select operation has been specified in the utility-modifier parameter.

DNOJS Parameter

When the routine is running under Monitor, this parameter informs Monitor that the data input is not on SYSIPT.

END Parameter

This parameter signifies the end of parameter input when a utility-modifier parameter has been used.

Routine Parameters - Detailed

Utility-Modifier Parameter

◆ *Format:*

$$\Delta U \Delta Tx,FF,A=(80,80),B=(a,b),W=\begin{pmatrix}X,o\\X00,onn\end{pmatrix},L=(X,x),I1,Ox,Q=(x,y),Zn$$

Entry	Meaning
ΔUΔ	Parameter identifier.
Tx	Function: x = C copy cards to tape in unblocked format (preset). = R copy cards to tape in blocked format. = F field select; copy cards to tape in unblocked format. = RF field select; copy cards to tape in blocked format.
FF	Input record format: Fixed-length records (preset).
A = (80,80)	Input format: Unblocked 80-character records (preset).
B = (a,b)	Output format: a = output record size. b = output block size (multiple of record size). Note: Preset values are 80,80.
W = (X, o) or W = (X00, onn)	Initial positioning of output tape: o = R rewind (preset). = N do not rewind. nn = number of tape marks to be unwound after tape has been positioned (01-99).
L = (X, x)	Output tape labeling: x = X labels supplied by TPLAB card or by user own-coding (preset). = N unlabeled output with beginning tape mark. = T unlabeled output without beginning tape mark.
I1	Input mode: EBCDIC card input (preset).

Utility-Modifier Parameter (Cont'd)

Entry	Meaning
Ox	Disposition of output tape: x = R rewind (preset). = U rewind and unload. = N do not rewind but leave positioned after double tape mark.
Q = (x,y)	Input sequence check: x = first column in input card to be sequence checked. (Column 1 = 1.) y = length of field to be checked (maximum of 10 characters). Note: Preset function does not provide for sequence checking.
Zx	Log routine parameters: x = L log (preset). = N do not log.

Notes:

1. All entries are optional and may appear in any order. When an entry is omitted, the preset value is assumed.
2. All entries, except the first, must be preceded by a comma.
3. Multiple parameter cards may be used; it is not required that all entries appear on a single card.
4. Refer to page 1-5 for parameter requirements with paper tape input.

Examples:

```
ΛUΛTR,B=(80,400)

ΛUΛL=(X,N)

ΔUΛTF,W=(X00,N02),ON

ΛUΛTRF,B=(44,220),OU,Q=(1,6),ZN
```

Field-Select Parameter

◆ *Format:*

ΛFSΛr,s,t/r,s,t/......./r,s,t

Entry	Meaning
ΛFSΛ	Parameter identifier.
r,s,t	r = starting position (relative to one) of the input field to be selected. s = size of input field. t = starting position (relative to one) of the field in the output record. Note: Commas must be used to separate entries, and a slash (/) to separate one set of entries from the next. See examples.
(U,n,m)	If the input field is to be unpacked when moved to the output record, replace s in above parameter with (U,n,m): U = identifies unpack operation. n = size of input field in bytes. m = size of output field in bytes. Note: r and t remain the same.
(P,n,m)	If the input field is to be packed when moved to output record, replace s in above parameter with (P,n,m): P = identifies pack operation. n = size of input field in bytes. m = size of output field in bytes. Note: r and t remain the same.

Notes:

1. When a field-select parameter is used, the utility-modifier parameter also must be used.
2. When using field selection, all positions in the output record not filled with input data will be space-filled.
3. More than one field-select card may be used; it is unnecessary to fill a card completely before starting another. All information for a particular field, however, must appear on the same card.

Examples:

ΛFSΛ1,6,14/7,10,36/17,1,1

ΛFSΛ17,30,41

ΛFSΛ1,(P,5,3),14/7,10,36/17,(U,2,3),1

ΛFSΛ18,(P,30,16),41

DNOJS Parameter

◆ *Format:*

ΛDNOJS

This parameter informs Monitor that the data input for the routine is not on SYSIPT. The routine will then request the assignment of PRIPT.

Parameter Examples
(Cont'd)

2. To copy a card file to tape, five cards per block.

```
∧U∧TR,FF,A = (80,80),B = (80,400),OR

∧END

(data cards)

/*
```

3. To copy a card file to tape, 5 records per block, selecting only the first 36 positions of input records to be transcribed to tape.

```
∧U∧TRF,FF,A = (80,80),B = (36,180),OR

∧FS∧1,36,1

∧END

(data cards)

/*
```

END Parameter

◆ *Format:*

Δ

This parameter signifies the end of parameter input and must be included whenever a utility-modifier parameter has been used.

◆ *Under Executive Control*

SDN	Device Type	Remarks
PRIPT	Card reader or paper tape reader.	Input device.
PROPT1	Magnetic tape.	Primary output device.
PROPT2	Magnetic tape.	Alternate output device.
PRPRM	Card reader, magnetic tape, or paper tape reader.	Parameter input device.

◆ *Under Monitor Control*

SDN	Device Type	Remarks
SYSIPT	Card reader or magnetic tape.	Parameter and card data or card image input device.
PRIPT	Paper tape reader or card reader. *	Paper tape or card data input device (if not on SYSIPT).
PROPT1	Magnetic tape.	Primary output device.
PROPT2	Magnetic tape.	Alternate output device.

*Refer to page 1-4 for paper tape input considerations.

Parameter Examples

◆ 1. To copy a card file to tape utilizing the standard preset functions:

(data cards)

/*

Note: Although routine parameters are not required, VOL and TPLAB cards must be provided for label generation.

SELECTIVE CARD TO PRINTER AND/OR PUNCH (CDPR)

General Description

◆ The Selective Card to Printer and/or Punch routine transcribes 80-column card records or paper tape to punched cards or paper tape and/or to the printer.

Card files are punched in EBCDIC and the final card contains /* in the first two columns to signify the end of the file.

Printed output may be in Character (EBCDIC graphics) mode or Hexadecimal (two digits per character) mode; the print format may be List or Display.

List Format

Output is restricted to one print line (132 or 160 print positions) of data per input record transcribed.

This format may be specified with or without field selection. When the field-select operation is not specified, data is printed in the Character mode with no intervening spaces on the print line, nor spacing between lines. When the field-select operation is used, all selected data is printed in the mode (Character or Hexadecimal) and the print positions specified by the field-select parameters.

Headings are printed only when specified.

Display Format

The complete 80-character input record is printed.

A scale line is printed at the top and bottom of each page.

The first print line for a record contains the block number, record number within the block, block size, and record size in the first 24 positions. Record data begins in position 27 of each print line.

No field-select parameters are permitted; however, Character or Hexadecimal mode may be specified in the utility-modifier parameter. When Character mode is specified, the print line is edited to space between each set of 10 data characters. When Hexadecimal mode is specified, the print line is edited to space between each set of four print characters (two bytes).

Headings are printed only when specified.

Refer to page 2-21 for sample Display formats.

General Description *(Cont'd)*

Preset Functions

This routine is preset for the following functions:

To select all cards of the input file and to print them on a 132-character print line, in Display format and Character mode, single spaced, and with page numbers. (Page advance and page numbering assume that the printer carriage tape contains punches in channel 12.)

To type out record counts and the number of pages printed at the end of the routine.

To provide a console typewriter listing (log) of the input parameters.

Optional Functions

The following optional functions may be specified by the use of parameters:

To print-only or punch-only all records of the input file.

To print-only or punch-only selected records of the input file.

To specify Hexadecimal print mode or EBCDIC punch mode.

To print in List format.

To double or triple space; to suppress page numbering.

To sequence check the input; to sequence number the output.

To field select, pack, unpack, or convert to Hexadecimal mode any or all input fields.

To print header lines on each page.

To suppress the logging of input parameters.

Input

◆ Input to this routine consists of a card file to be transcribed to punched cards and/or printed. The last card of this file must contain /* in the first two columns. If a nonblank character is in column 80, the /* is treated as data.

When optional functions are desired, the parameters that specify these options must be entered before the first input record is read.

Output

◆ Output of this routine consists of the contents of the input card file transcribed to punched cards and/or printed in the format specified by the parameters. The last card of an output card file contains /* in the first two columns.

Equipment Configuration

Required

◆ Processor (65K).

Console typewriter.

Card reader, or Videoscan document reader with card read feature.

Card punch.

Printer.

Optional

◆ A paper tape reader may be used instead of the card reader. Refer to page 1-4 for paper tape considerations.

Routine Parameters - General

◆ The following six parameter cards are used with this routine:

Utility-Modifier Parameter

This parameter specifies a print and/or punch operation, field-selection, output format, output mode, print spacing, page numbering, input sequence checking, output sequence numbering, and logging of input parameters.

Field-Select Parameter(s)

This parameter specifies field-selection, packing, unpacking, and hexadecimal conversion of fields in the input records.

Page Heading Parameters

These parameters provide the text for page headings on the output listing.

PRINT Parameter

This parameter specifies the character that must appear in the first position of an input record to be selected for printing. Only records containing the designated select character are printed. (Multiple select characters may be supplied.)

PUNCH Parameter

This parameter specifies the character that must appear in the first position of an input record to be selected for punching. Only records containing the designated select character are punched. (Multiple select characters may be supplied.)

DNOJS Parameter

When the routine is running under Monitor, this parameter informs Monitor that the data input is not on SYSIPT.

PTOPT Parameter

When running under Monitor, this parameter informs the routine that paper tape output is to be produced.

END Parameter

This parameter signifies the end of parameter input whenever a utility-modifier parameter has been used.

Routine Parameters - Detailed

Utility-Modifier Parameter

◆ *Formats:*

Print and Punch

ΔUΔTx,FF,A = (80,80),B = (p),I1,OC,Q = (x,y),N = (c,d),Sn,Px,Zx

Print-only

ΔUΔTx,FF,A = (80,80),B = (p),I1,Ox,Q = (x,y),Sn,Px,Zx

Punch-only

ΔUΔTx,FF,A = (80,80),B = (80,80),I1,O1,Q = (x,y),N = (c,d),Zx

Entry	Meaning
ΔUΔ	Parameter identifier.
Tx	Function: x = D print-only in Display format (preset). = B punch and print in List format. = BF punch and print in List format with field-selection. = C punch-only. = F punch-only with field-selection. = L print-only in List format. = LF print-only in List format with field-selection.
FF	Input record format: Fixed-length records (preset).
A = (80,80)	Input format: Unblocked, 80-character records (preset).
B = (p) or B = (80,80)	Output format: p = 132 or 160 (printer line size; preset to 132). (80,80) = punch-only function.
I1	Input mode: EBCDIC mode.
Ox	Output mode: x = C Character mode for print or print and punch function (preset). = 1 EBCDIC mode for punch-only function. = X Hexadecimal mode for print-only function (Display format, only).

Utility-Modifier Parameter (Cont'd)

Entry	Meaning
Q = (x,y)	Input sequence check: x = first column in input field to be sequence checked. (Column 1 = 1.) y = length of field to be checked (maximum of 10 characters). <u>Note</u>: Preset function does not provide for sequence checking.
N = (c,d)	Output sequence numbering: c = first column of sequence field in output card. d = length of field to be numbered (maximum of 10 characters). <u>Note</u>: Preset function does not provide for sequence numbering.
Sn	Spacing: n = 1 single spacing (preset). = 2 double spacing. = 3 triple spacing.
Px	Page numbering: x = Y number pages (preset). = N do not number pages.
Zx	Log routine parameters: x = L log (preset). = N do not log.

Notes:

1. All entries are optional and may appear in any order. If an entry is omitted, the preset value is assumed.
2. All entries, except the first, must be preceded by a comma.
3. Multiple parameter cards may be used; it is not required that all entries appear on a single card.
4. Refer to page 1-5 for parameter requirements with paper tape input or output.

Examples:

```
ΛUΛTB,B=(160),Q=(10,8),N=(1,8),ZN

ΛUΛTBF,B=(160),OC,Q=(1,5),N=(1,9)

ΛUΛTC,B=(80,80),O1,Q=(5,6)

ΛUΛTD,OX,ZN

ΛUΛTLF,S2,PN
```

Field-Select Parameter

Format:

ΔFSΔr,s,t/r,s,t/...../r,s,t

Entry	Meaning
ΔFSΔ	Parameter identifier.
r,s,t	r = starting position (relative to one) of the input field to be selected. s = size of input field. t = starting position (relative to one) of the field in the output record. Note: Commas must be used to separate entries, and a slash (/) to separate one set of entries from the next. See examples.
(U,n,m)	If the input field is to be unpacked when moved to the output record, replace s in above parameter with (U,n,m): U = identifies unpack operation. n = size of input field in bytes. m = size of output field in bytes. Note: r and t remain the same.
(P,n,m)	If the output field is to be packed when moved to the output record, replace s in above parameter with (P,n,m): P = identifies pack operation. n = size of input field in bytes. m = size of output field in bytes. Note: r and t remain the same.
(X,n)	If each character in the input field selected is to be converted to its hexadecimal equivalent (two characters), replace s in the above parameter with (X,n): X = hexadecimal operation. n = size of input field. (Note: The size of the output field is assumed to be 2 x n.) Note: r and t remain the same.

Notes:

1. When a field-select parameter is used, the utility-modifier parameter also must be used.
2. When using field selection, all positions in the output record not filled with input data will be space-filled.
3. More than one field-select card may be used; it is unnecessary to fill a card completely before starting another. All information for a particular field, however, must appear on the same card.

Examples:

ΔFSΔ1,15,5/16,15,20

ΔFSΔ3,(U,6,11),15/22,40,33

ΔFSΔ63,(X,17),47/1,46,1

ΔFSΔ5,(X,6),2

Page Heading Parameters

◆ *Format:*

ΔHnΔtext

Entry	Meaning
ΔH1Δtext	Text for positions 1-76 of print header lines.
ΔH2Δtext	Text for positions 77-142 of print header lines.
ΔH3Δtext	Text for positions 143-160 of print header lines.

Note:

Any or all parameters may be supplied.

PRINT Parameter

◆ *Format:*

ΔPRINTx......x//Δ

Entry	Meaning
ΔPRINT	Parameter identifier.
x......x	Select character(s): x = any character that must appear as the first character in an input record to be printed. Note: Column 77 is the last column in which a select character may be specified. The Select character does not get printed. Ⓐ
//Δ	Parameter termination sequence.

Examples:

To print only records containing an A in the first position:

ΔPRINTA//Δ

To print only records containing an A,5,7, or Y in the first position:

ΔPRINTA57Y//Δ

PUNCH Parameter

◆ *Format:*

ΔPUNCHx......x//Δ

Entry	Meaning
ΔPUNCH	Parameter identifier.
x......x	Select character(s): n = any character that must appear as the first character in an input record to be printed. Note: Column 77 is the last column in which a select character may be specified. The Select character does not get punched. Ⓐ
//Δ	Parameter termination sequence.

Device Assignments *(Cont'd)*

Under Monitor Control

SDN	Device Type	Remarks
SYSIPT	Card reader or magnetic tape.	Parameter and card data or card image input device.
PRIPT	Paper tape reader or card reader.*	Paper tape or card data input device (if not in SYSIPT).
SYSOPT	Card punch or magnetic tape.	Output device.
PROPT	Paper tape punch. *	Output device.
SYSLST	Printer or magnetic tape.	Output device.

*Refer to page 1-4 for paper tape input or output considerations.

Parameter Examples

◆ 1. To transcribe an input card file to punched cards and to print in List format using the preset functions.

```
(input card file)
      /*
```

2. To print all input cards with K,P, or 3 in the first position using Display format and Hexadecimal mode.

```
ΔU ΔTD,B= (160),OX,S2,PY
Δ PRINTKP3//Δ
ΔEND
(input card file)
/*
```

3. To (a) punch only input records containing an A,1 or Z as their first character, and (b) print only input records containing a 6,B, or Z as their first character. (All records will be field selected as specified by the field select parameter.)

```
ΔUΔTBF,N= (76,5),S2,PY
ΔFSΔ5,10,1/16,(X,3),12/76,5,20
ΔPUNCHA1Z//Δ
ΔPRINT6BZ//Δ
ΔEND
(card input file)
/*
```

PUNCH Parameter (Cont'd)

Examples:

To punch only records containing an A in the first position.

Δ PUNCHA//Δ

To punch only records containing an A,5,7,or Y in the first position.

Δ PUNCHA57Y//Δ

DNOJS Parameter

◆ *Format:*

Δ DNOJS

This parameter informs Monitor that the data input for the routine is not on SYSIPT. The routine will then request the assignment of PRIPT.

PTOPT Parameter

◆ *Format:*

ΔPTOPT

This parameter is used under Monitor to inform the routine that paper tape output is to be produced. The routine will then request the assignment of PROPT.

END Parameter

◆ *Format:*

Δ END

This parameter signifies the end of parameter input and must be included.

Ⓓ

Device Assignments

Under Executive Control

SDN	Device Type	Remarks
PRIPT	Card reader or paper tape reader.*	Data input device.
PROPT	Card punch or paper tape punch. *	Output device.
PRLST	Printer.	Output device.
PRPRM	Card reader, paper tape reader, or magnetic tape.	Parameter input device.

Sample Print Formats

◆ *List Format*

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......XXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......XXXXXXXXXX
```

Note: When field selection is used, the edited print line is constructed as specified by the field-select parameters.

Display Format

Character Mode

```
Scale
     -> B#  R#  BS  RS  1          10     ......         90
Line

        1   1   80  80  XXXXXXXXXX XX......XXXXXXXXXX
                        XXXXXXXXXX XX......XXXXXXXXXX

        2   1   80  80  XXXXXXXXXX XX......XXXXXXXXXX
                        XXXXXXXXXX XX......XXXXXXXXXX
```

Hexadecimal Mode

```
Scale
     -> B#  R#  BS  RS      2      4      6......   40
Line

        1   1   80  80  XXXX  XXXX  XXXX......XXXX
        2   1   80  80  XXXX  XXXX  XXXX......XXXX
```

Scale Line Key:

B# = input block number.

R# = record number within the input block.

BS = block size.

RS = record size.

TAPE TO TAPE (TPTP)

General Description

◆ The Tape to Tape routine transcribes data from one magnetic tape to another. The input and output tape blocks can range in size from 12 to 4,096 characters and can contain fixed-length records, variable-length records, or records of undefined size. Except for records of undefined size, records can be blocked or unblocked.

In the case of fixed-length records, input fields can be field-selected, packed, or unpacked during the copying process.

Tape volumes may be labeled or unlabeled, single or multivolume.

Preset Functions

This routine is preset for the following functions:

To copy a magnetic tape file of undefined records (up to 1,000 characters) to the output tape.

To rewind tapes to BOT at the start and end of transcription.

To check standard header and trailer labels* on input tapes.

To write standard header and trailer labels* on output tapes.

To alternate tapes for multivolume output when two tape devices are available.

To type out record and block counts at the end of the routine.

To provide a console typewriter listing (log) of the input parameters.

Optional Functions

The following optional functions may be specified by the use of parameters:

Reblocking of fixed-length record input.

Fixed or variable-length record processing.

Field-selecting, packing, or unpacking fields in fixed-length records.

Processing of unlabeled, single-volume input files.

Positioning and disposition of input and output tapes.

Suppressing the logging of input parameters.

---

*Label information is provided by VOL and TPLAB run time parameter cards. For processing of user labels, see Modification of Peripheral Routines, Appendix A.

Input

◆ Input to this routine consists of a labeled (single or multivolume) or unlabeled (single volume) tape file containing fixed-length, variable-length, or undefined records.

Fixed- or variable-length records may be unblocked or blocked. If blocked, the last block may be short. Undefined records must be unblocked.

When optional functions are desired, the parameters that select these options must be entered from the parameter input device.

Output

◆ Output of this routine consists of a labeled (single or multivolume) or unlabeled (single volume) tape file. Fixed- or variable-length records may be blocked, unblocked, or reblocked; records of undefined length must always be unblocked.

Record and block counts are typed out at the end of this routine; a log of input parameters may also be provided.

Equipment Configuration

Required

◆ Processor (65K).

Console typewriter.

Magnetic tape devices (two required).

Card reader, or Videoscan document reader with card read feature.

Optional

◆ Additional magnetic tape devices may be assigned for multivolume input or output.

Routine Parameters - General

◆ The following three parameters are used with this routine:

Utility-Modifier Parameter

This parameter specifies blocking and reblocking of input records, record format, block and record sizes, and field selection; positioning and disposition of the input and output tapes, logging of parameters, and generation of unlabeled output volumes.

Field-Select Parameter(s)

This parameter specifies field selection, packing, and unpacking of fields in the input records. This parameter must be used when the field-select operation has been specified in the utility-modifier parameter.

END Parameter

This parameter signifies the end of all parameter input.

Routine Parameters - Detailed

Utility-Modifier Parameter

◆ *Format:*

$$\Delta U \Delta Tx,Fx,A=\left(\begin{matrix}g\\ n,m\end{matrix}\right),B=\left(\begin{matrix}h\\ a,b\end{matrix}\right),W=\left(\begin{matrix}i,o\\ ixx,oxx\end{matrix}\right),L=(i,o),Ix,Ox,Zx$$

Entry	Meaning
ΔUΔ	Parameter identifier.
Tx	Function: x = C copy input tape file to an output tape (preset). = R reblock fixed-length record input. = F field select fixed-length record input. = RF reblock and field select fixed-length record input. Note: Whenever the number of records in an input block differs from the number of records in the output block, this is considered reblocking. In the case of variable-length records, it is unnecessary to specify reblocking. The routine will automatically reblock (if input and output block sizes differ) based on the contents of the block length and record length fields at the beginning of the block.
Fx	Input record format: x = U undefined records (preset). = F fixed-length records. = V variable-length records.
A = (g) or A = (n,m)	Input format: g = maximum size of variable-length or undefined records (preset = 1,000). n = size of fixed-length records. m = size of input blocks. Note: Maximum input block size is 4,096 characters.
B = (h) or B = (a,b)	Output format: h = maximum size of variable-length or undefined records (preset = 1,000). a = size of fixed-length records. b = size of output blocks. Note: Maximum output block size is 4,096 characters.

Utility Modifier Parameter (Cont'd)

Entry	Meaning
W = (i, o) or W = (ixx, oxx)	Initial positioning of tapes: i = R rewind input tape (preset). = N do not rewind input tape. o = R rewind output tape (preset). = N do not rewind output tape. xx = number of tape marks to be unwound after the tape has been positioned (01-99).
L = (i, o)	Tape labeling: i = X check label against TPLAB parameter (preset). = N unlabeled input tape. o = X label supplied by TPLAB parameter (preset). = N unlabeled output tape with beginning tape mark. = T unlabeled output tape without beginning tape mark.
Ix	Disposition of input tape: x = R rewind (preset). = U rewind and unload. = N do not rewind. = M multivolume input; rewind and unload.
Ox	Disposition of output tape: x = R rewind (preset). = U rewind and unload = N do not rewind but leave positioned after double tape marks.
Zx	Log routine parameters: x = L log (preset). = N do not log.

Notes:

1. All entries are optional and can appear in any order. If an entry is omitted, the preset value is assumed.
2. All entries, except the first, must be preceded by a comma.
3. Multiple parameter cards may be used; it is not required that all entries appear on a single card.

Examples:

ΔUΔTC,FU,A = (1030),B = (1030)

ΔUΔTRF,FF,A = (80,80),B = (80,400),IU,OR

ΔUΔTC,FF,A = (80,400),B = (80,400),IM

ΔUΔTF,FV,A = (556),B = (556),L = (X,N),ZN

Field-Select Parameter

◆ *Format:*

ΔFSΔr,s,t/r,s,t/....../r,s,t

Entry	Meaning
ΔFSΔ	Parameter identifier.
r,s,t	r = starting position (relative to one) of the input field to be selected. s = size of input field. t = starting position (relative to one) of the field in the output record. Note: Commas must be used to separate entries, and a slash (/) to separate one set of entries from the next. See examples.
(U,n,m)	If the input field is to be unpacked when moved to the output record, then replace s in above parameter with (U,n,m): U = identifies unpack operation. n = size of input field in bytes. m = size of output field in bytes. Note: r and t remain the same.
(P,n,m)	If the input field is to be packed when moved to output records, replace s in above parameter with (P,n,m): P = identifies pack operation. n = size of input field in bytes. m = size of output field in bytes. Note: r and t remain the same.

Notes:

1. When a field-select parameter is used, the utility-modifier parameter also must be used.
2. When using field selection, all positions in the output record not filled with input data will be space-filled.
3. More than one field-select card can be used; it is unnecessary to fill a card completely before starting another. All information for a particular field, however, must appear on the same card.

Examples:

ΔFSΔ1,6,14/7,10,36/17,1,1

ΔFSΔ16,30,41

ΔFSΔ1,(P,5,3),14/17,10,36/17,(U,2,3),1

ΔFSΔ18,(P,30,16),41

Parameter Examples *(Cont'd)*

3. To copy variable-length, blocked input records with a maximum size of 800 characters.

```
ΔUΔTF,FV,A = (800),B = (800)
ΔEND
```

END Parameter

◆ *Format:*

ΔEND

This parameter signifies the end of parameter input and must be included when a utility-modifier parameter has been used.

Device Assignments

Under Executive Control

SDN	Device Type	Remarks
PRIPT1	Magnetic tape.	Primary input device.
PRIPT2	Magnetic tape.	Alternate input device.
PROPT1	Magnetic tape.	Primary output device.
PROPT2	Magnetic tape.	Alternate output device.
PRPRM	Magnetic tape, paper tape reader, or card reader.	Parameter input device.

Under Monitor Control

SDN	Device Type	Remarks
PRIPT1	Magnetic tape.	Primary input device.
PRIPT2	Magnetic tape.	Alternate input device.
PROPT1	Magnetic tape.	Primary output device.
PROPT2	Magnetic tape.	Alternate output device.
SYSIPT	Card reader or magnetic tape.	Parameter input device.

Parameter Examples

◆ 1. To copy undefined tape records to an output tape using the preset functions:

No routine parameters required. (VOL and TPLAB cards, however, must be provided for label checking and generation.)

2. To reblock fixed-length input records from 5 to 10 records per block and field-select the identification field from position 96 to 1.

```
ΔUΔTRF,A=(100,500),B=(100,1000)
ΔFSΔ1,95,6/96,5,1
ΔEND
```

SELECTIVE TAPE TO PRINTER AND/OR PUNCH (TPPR)

General Description

◆ The Selective Tape to Printer and/or Punch routine transcribes data from magnetic tape to punched cards and/or to the printer.

The input volume may be labeled or unlabeled. If labels are used, a multivolume input file may be processed.

Card output files are punched in EBCDIC with the final card containing /* in the first two columns to signify the end of file.

Printed output may be in Character (EBCDIC graphics) mode or Hexadecimal.(two digits per character) mode; the print format may be List or Display. Refer to page 2-37 for sample Print formats.

List Format

Output is restricted to one print line (132 or 160 print positions) of data per input record transcribed.

This format may be specified with or without field selection. When the field-select operation is not specified, data is printed in the Character mode with no intervening spaces on the print line. When the field-select operation is specified, all selected data is printed in the mode (Character or Hexadecimal) and the print positions specified by the field-select parameters.

Headings are printed only when specified.

Display Format

The complete input record is printed, regardless of its length and the number of print lines required.

A scale line is printed at the top and bottom of each page.

The first print line for a record contains the block number, record number within the block, block size, and record size in the first 30 positions. Record data begins in position 31 of each print line.

No field-select parameters are permitted; however, Character or Hexadecimal mode may be specified in the utility-modifier parameter. When Character mode is specified, the print line is edited to space between each set of 10 data characters. When Hexadecimal mode is specified, the print line is edited to space between each set of four print characters (two bytes).

Headings are printed only when specified.

General Description *(Cont'd)*

Preset Functions

This routine is preset for the following functions:

To process undefined input tape records of up to 1,000 characters.

To select and print all input records on a 132-character print line; in Display format and Character mode; single spaced, with page numbers.

To rewind the input tape to BOT at the start and end of transcription.

To check standard header and trailer labels* on input tapes.

To type out record counts, block counts, and number of pages printed at the end of the routine.

To provide a console typewriter listing (log) of the input parameters.

Optional Functions

The following optional functions may be specified by the use of parameters:

Processing of a single, unlabeled input volume.

Punching only all input records.

Printing only or punching only selected records of the input file.

Hexadecimal print mode or EBCDIC punch mode.

Printing in List format.

Double or triple spacing; suppression of page numbers.

Sequence numbering of the output cards.

Field-selecting, packing, unpacking, and conversion to Hexadecimal mode specific fields of fixed-length records.

Positioning and disposition of the input tape.

Providing the text for print header lines.

Positioning the input file to the first record to be printed or punched.

Suppressing the logging of input parameters.

---

*Standard label information is provided by VOL and TPLAB run time parameter cards. For processing of user labels, see Modification of Peripheral Routines, Appendix A.

Input

◆ Input to this routine consists of a labeled or unlabeled tape file containing fixed-length, variable-length, or undefined records to be transcribed to the printer and/or the card punch.

Fixed- or variable-length records may be unblocked or blocked. Undefined records must be unblocked.

When optional functions are desired, the parameters that specify these options must be entered from the parameter input device.

Output

◆ Output consists of the contents of the tape file transcribed to punched cards and/or printed in the format specified by the parameters. The last card punched contains /* in the first two columns.

Equipment Configuration

Required

◆ Processor (65K)

Console typewriter.

Card punch.

Magnetic tape device.

Card reader, or Videoscan document reader with card read feature.

Printer.

Optional

◆ An additional magnetic tape device can be used for multivolume input. The paper tape punch can be used instead of the card punch. Refer to page 1-4 for paper tape considerations.

Routine Parameters - General

◆ The following six parameters are used with this routine:

Utility Modifier Parameter

This parameter specifies print and/or punch operation, field selection, input format, output format, and output mode; positioning and disposition of the input tape; print spacing, page numbering; sequence numbering of the output; and logging of parameters.

Field Select Parameter(s)

This parameter specifies field selection, packing, unpacking, and Hexadecimal conversion of fields in the input records.

This parameter must be used when the field-select operation has been specified in the utility-modifier parameter.

Page Heading Parameter(s)

This parameter provides the text for page headings on the output listing.

Routine Parameters - General *(Cont'd)*

PRINT Parameter

This parameter specifies the character that must appear in the first position of an input record to be selected for printing. Only records containing the designated select character are printed. (Multiple select characters may be supplied.)

PUNCH Parameter

This parameter specifies the character that must appear in the first position of an input record to be selected for punching. Only records containing the designated select character are punched. (Multiple select characters may be supplied.)

PTOPT Parameter

When running under Monitor, this parameter informs the routine that paper tape output is to be produced.

END Parameter

This parameter signifies the end of parameter input whenever a utility-modifier parameter has been used.

Routine Parameters - Detailed

Utility-Modifier Parameter

◆ *Format:*

Print and Punch

$$\Delta \mathrm{U}\Delta \mathrm{Tx,Fx,A=}\begin{pmatrix} g \\ n,m \end{pmatrix}\mathrm{,B=(p),W=}\begin{pmatrix} i,X \\ ixx,X00 \end{pmatrix}\mathrm{,L=(i,X),Ix,Ox,Rn,N=(c,d),Sn,Px,Zx}$$

Print-only

$$\Delta \mathrm{U}\Delta \mathrm{Tx,Fx,A=}\begin{pmatrix} g \\ n,m \end{pmatrix}\mathrm{,B=(p),W=}\begin{pmatrix} i,X \\ ixx,X00 \end{pmatrix}\mathrm{,L=(i,X),Ix,Ox,Rn,Sn,Px,Zx}$$

Punch-only

$$\Delta \mathrm{U}\Delta \mathrm{Tx,Fx,A=}\begin{pmatrix} g \\ n,m \end{pmatrix}\mathrm{,B=(80,80),W=}\begin{pmatrix} i,X \\ ixx,X00 \end{pmatrix}\mathrm{,L=(i,X),Ix,O1,Rn,N=(c,d),Zx}$$

Entry	Meaning
ΔUΔ	Parameter identifier.
Tx	Function: x = D print-only in Display format (preset). = B punch and print in List format. = BF punch and print in List format with field-select. = C punch-only. = F punch-only with field-select. = L print-only in List format. = LF print-only in List format with field-select. = MB punch and print Monitor input in List format. = MC punch-only Monitor input. = ML print-only Monitor input in List format.

Utility-Modifier Parameter (Cont'd)

Entry	Meaning
Fx	Input record format: x = U undefined records (preset). = V variable-length records. = F fixed-length records.
A = (g) or A = (n,m)	Input format: g = maximum size of variable-length or undefined records (preset = 1,000). n = size of fixed-length records. m = size of fixed-length record blocks. Note: Maximum block size is 4,096 characters.
B = (p) or B = (80,80)	Output format: p = 132- or 160-position print line (preset to 132). (80,80) = punch-only function specified.
W = (i,X) or W = (ixx,X00)	Initial positioning of the input tape: i = R rewind (preset). = N do not rewind. xx = number of tape marks to be unwound after the tape has been positioned (01-99).
L = (i,X)	Input tape label checking: i = X check label against TPLAB parameter (preset). = N unlabeled input tape.
Ix	Disposition of input tape: x = R rewind (preset). = U rewind and unload. = N do not rewind. = M multivolume input; rewind and unload.
Ox	Output mode: x = C Character mode for print or print and punch functions (preset). = 1 EBCDIC mode for punch-only function. = X Hexadecimal mode for printing in Display format.
Rn	Record bypass (optional): First logical record of input file to be printed and/or punched, where n may range from 1 to 99999. Note: All records preceding record n will be bypassed.

Utility-Modifier Parameter (Cont'd)

Entry	Meaning
N = (c,d)	Output sequence numbering: c = first column of sequence field in output card. d = length of field to be numbered (maximum of 10 characters). <u>Note</u>: Preset function does not provide for sequence numbering.
Sn	Spacing: n = 1 single spacing (preset). = 2 double spacing. = 3 triple spacing. = 4 input record contains write control byte as <u>first</u> character.
Px	Page numbering: x = Y number pages (preset). = N do not number pages.
Zx	Log routine parameters: x = L log (preset). = N do not log.

Notes:

1. All entries are optional and can appear in any order. If an entry is omitted, the preset value is assumed.
2. All entries, except the first, must be preceded by a comma.
3. Multiple parameter cards may be used; it is not required that all entries appear on a single card.
4. Refer to page 1-5 for parameter requirements with paper tape output.

Examples:

```
ΔUΔTBF,FF,A=(100,1000),B=(160),R10,N=(3,4),ZN

ΔUΔTC,FV,A=(130),B=(80,80),IU

ΔUΔTD,FU,A=(650),B=(132),OX,S2,PN

ΔUΔTL,FV,A=(200),B=(160)
```

Field-Select Parameter

◆ *Format:*

ΛFSΛr,s,t/r,s,t....../r,s,t

Entry	Meaning
ΛFS Λ	Parameter identifier.
r,s,t	r = starting position (relative to one) of the input field to be selected. s = size of input field. t = starting position (relative to one) of the field in the output record. Note: Commas must be used to separate entries, and a slash (/) to separate one set of entries from the next. See examples.
(U,n,m)	If the input field is to be unpacked when moved to the output record, replace s in above parameter with (U,n,m): U = identifies unpack operation. n = size of input field in bytes. m = size of output field in bytes. Note: r and t remain the same.
(P,n,m)	If the input field is to be packed when moved to output record, replace s in above parameter with (P,n,m): P = identifies pack operation. n = size of input field in bytes. m = size of output field in bytes. Note: r and t remain the same.
(X,n)	If each character of the input field is to be converted from EBCDIC to its hexadecimal equivalent (two characters), replace s in above parameter with (X,n); X = hexadecimal operation. n = size of input field (Note: the size of the output field is assumed to be 2 x n). Note: r and t remain the same.

Notes:

1. When a field-select parameter is used, the utility-modifier parameter also must be used.
2. When using field selection, all positions in the output record not filled with input data will be space-filled.
3. More than one field-select card may be used; it is unnecessary to fill a card completely before starting another. All information for a particular field, however, must appear on the same card.

Examples:

ΛFSΔ15,32,40

ΛFSΔ1,(P,8,5),15/6,10,20/15,(U,3,5),3

ΛFSΛ5,(X,6),2

Page Heading Parameters

◆ *Format:*

ΔHnΔtext

Entry	Meaning
ΔH1Δtext	Text for positions 1-76 of print header lines.
ΔH2Δtext	Text for positions 77-142 of print header lines.
ΔH3Δtext	Text for positions 143-160 of print header lines.

Note:

Any or all parameters may be supplied.

PRINT Parameter

◆ *Format:*

ΔPRINTx......x//Δ

Entry	Meaning
ΔPRINT	Parameter identifier.
x......x	Select character(s): x = any character that must appear as the first character in an input record to be printed. Note: Column 77 is the last column in which a select character may be specified. The Select character does not get printed. Ⓐ
//Δ	Parameter termination sequence.

Examples:

To print only records containing an A in the first position:

ΔPRINTA//Δ

To print only records containing an A,5,7, or Y in the first position:

ΔPRINTA57Y//Δ

Device Assignments

◆ *Under Executive Control*

SDN	Device Type	Remarks
PRIPT1	Magnetic tape.	Primary input device.
PRIPT2	Magnetic tape.	Alternate input device.
PROPT	Card punch or paper tape punch. *	Output device.
PRLST	Printer.	Output device.
PRPRM	Card reader, paper tape reader, or magnetic tape.	Parameter input device.

Under Monitor Control

SDN	Device Type	Remarks
PRIPT1	Magnetic tape.	Primary input device.
PRIPT2	Magnetic tape.	Alternate input device.
SYSOPT	Card punch or magnetic tape.	Output device.
PROPT	Paper tape punch. *	Output device.
SYSLST	Printer or magnetic tape.	Output device.
SYSIPT	Card reader or magnetic tape.	Parameter input device.

*Refer to page 1-4 for paper tape output consideration.

Parameter Examples

◆ 1. To print selected blocked, variable-length records (maximum size of 122 characters) in List format and Character mode.

```
ΔUΔTL,FV,A=(122),B=(160)
/ΔPRINTQRS123//Δ
ΔEND
```

PUNCH Parameter

◆ *Format:*

Δ PUNCHx......x//Δ

Entry	Meaning
ΔPUNCH	Parameter identifier.
x......x	Select character(s): n = any character that must appear as the first character in an input record to be punched. Note: Column 77 is the last column in which a select character may be specified. Ⓐ The Select character does not get punched.
//Δ	Parameter termination sequence.

Examples:

To punch only records containing an A in the first position.

ΔPUNCHA//Δ

To punch only records containing an A,5,7, or Y in the first position:

Δ PUNCHA57Y//Δ

PTOPT Parameter

◆ *Format:*

ΔPTOPT

This parameter is used under Monitor to inform the routine that paper tape output is to be produced. The routine will then request the assignment of PROPT.

END Parameter

◆ *Format:*

ΔEND

Ⓓ This parameter signifies the end of parameter input.

Parameter Examples
(Cont'd)

2. To punch and list selected fixed-length records, using field selection and page headings.

```
ΔUΔTBF,FF,A=(100,100),B=(160)

ΔFSΔ99,(X,2),1/1,98,5

ΔPRINT/24FH//Δ

ΛPUNCHKG34//Δ

ΔH2ΔACCOUNTINGΔLISTINGΔDEPT-XY5

ΔEND
```

3. To print and/or punch a Monitor SYSLST/SYSOPT tape, bypassing the label check.

```
ΔUΔTMB,FV,W=(R01,X00),L=(N,X),S4,PN
ΔEND
```

Sample Print Formats

◆ *List Format*

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......XXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......XXXXXXXXXX
```

Note:

The edited print line is constructed as specified by the field-select parameters.

Display Format

```
                     Character Mode (132 print positions)

Scale
     →B#   R#   BS   RS   1            10      ......          90
Line

      1    1   100  100   XXXXXXXXXX  XX......XXXXXXXXXX
                          XXXXXXXXXX  XX......XXXXXXXXXX

      2    1   100  100   XXXXXXXXXX  XX......XXXXXXXXXX
                          XXXXXXXXXX  XX......XXXXXXXXXX

                    Hexadecimal Mode (132 print positions)

Scale
     →B#   R#   BS   RS       2       4       6......    40
Line

      1    1   160   80   XXXX  XXXX  XXXX......XXXX
           1   160   80   XXXX  XXXX  XXXX......XXXX
```

Scale Line Key: B# = input block number.
R# = record number within the input block.
BS = block size.
RS = record size.

TAPE DUPLICATE (DUP)

General Description

◆ The Tape Duplicate routine makes one or more copies of a tape from BT to a double tape mark, or to the special EOV record if the tape is an RCA Master tape. The input tape may be seven or nine level, labeled or unlabeled, and contain interspersed tape marks. If the input is seven level and contains labels, both the labels and data must be in the same recording mode.

Preset Functions

The routine is preset to make one copy of a tape from BT to a double tape mark and display the number of bytes, blocks, and tape marks copied. Block size is preset to a maximum of 1,044 bytes.

Optional Functions

The routine provides the following optional functions:

1. Make more than one copy of the input.
2. Make copies to two or three tapes simultaneously.
3. Utilize up to six tape drives when tape swapping is selected.
4. Copy the special EOV record which appears after the double tape mark on RCA Master tapes. The routine will type an error message if the number of bytes read from the input does not agree with the number contained in the EOV record.
5. Read each output tape in reverse to determine if the number of bytes written to the tape agrees with the number of bytes read from the input.

Input

◆ Input to the routine consists of a seven- or nine-level magnetic tape and a parameter entered from the console typewriter or card reader.

Output

◆ Output of the routine is one or more copies of the input tape. The copies may be made to seven- or nine-level tape or a combination of these. Console typewriter messages indicate the number of bytes, blocks, and tape marks read from the input and any byte count discrepancies found on output tapes.

Equipment Configuration

Required

◆ Processor (65K)

Console typewriter

Magnetic tape devices (two required)

Optional

◆ Additional magnetic tape devices (up to six) may be used as output devices.

Seven-level tapes may be substituted for nine-level tapes.

Routine Parameter - General

◆ Only one parameter is used with this routine. It is entered using the console typewriter when running under the Executive and the card reader when running under Monitor.

Routine Parameter - Detailed

◆ *Format:*

ΔDUPΔa,bbb,c,Y,Y,N

Entry	Meaning
ΔDUPΔ	Parameter Identifier.
a	Number of output tapes to be created per pass (1-3).
,bbb	Total number of copies to be made (001-999).
,c	Tape swapping, where: c - S use tape swapping - N do not use tape swapping
,Y	Optional. This entry indicates that the input is an RCA Master tape and that the EOV record is to be copied to the output. Note: When this entry is not used and the next entry is used, the comma must appear.
,Y	Optional. This entry indicates that the byte count is to be checked on each output tape. Note: When this entry is not used and the next entry is used, the comma must appear.
,N	Optional. This entry indicates that the output tape is not to be purged. If not used, the VOL label on the output tape (if any) will be preserved if the input contains at least one VOL label.

Note:

1. When the parameter is entered from the console typewriter the leading space is not used.

Parameters Examples

◆ 1. Copy a Tape using Preset Functions

Under the Executive, preset functions are obtained by typing in nΔ and pressing EOT in reply to the message n DUP 0401A ENTER PARAM.

Under the Monitor, preset functions are obtained by not having a routine parameter on SYSIPT.

2. Make three copies of an RCA Master tape without tape swapping and with the EOV and byte count check options.

Δ3,003,N,Y,Y

Parameters Examples
(Cont'd)

3. Make eight copies of a tape using tape swapping (two copies per pass) and the byte count check option.

   Δ2,008,S,,Y

4. Make two copies of a tape with no tape swapping and only one output tape drive available.

   Δ1,002,N

Considerations For Use

◆ 1. Output tapes are not purged by this routine.

2. Since the byte count check is a read reverse operation the routine cannot check byte count on seven-level tapes. When this is specified for a seven-level tape, only the block count is checked.

3. Labels on seven-level tape must be recorded in the same mode as the data.

Device Assignments

◆ Under Monitor or Executive:

SDN	Device Type	Remarks
SYSIPT	Card reader or magnetic tape.	Parameter input under Monitor.
TAPEIN	Magnetic tape.	Input device.
TAPE01	Magnetic tape.	First output tape.
TAPE02	Magnetic tape.	Second output tape.
TAPE03	Magnetic tape.	Third output tape.
TAPE04	Magnetic tape.	Alternate for first output when tape swapping is used.
TAPE05	Magnetic tape.	Alternate for second output when tape swapping is used.
TAPE06	Magnetic tape.	Alternate for third output when tape swapping is used.

3. PERIPHERAL CONVERSION-RANDOM ACCESS

RANDOM ACCESS VOLUME INITIALIZER (RAINIT)

General Description

◆ The Random Access Volume Initializer routine prepares and formats random access volumes for use with the Spectra 70 TOS programming system. (A volume is defined here as being one 70/564 Disc Unit, one 70/565 Drum Unit, or a 70/568 magazine.)

Preset Functions

This routine formats a random access volume in the following manner:

1. A service analysis is performed by writing to and reading from each track. (If a defective track is detected, an alternate track in the volume is assigned.)
2. A Home Address record and a Track Descriptor record are created and written at the beginning of each track.
3. For disc and drum, two special IPL preventative coding blocks are created and written as records 1 and 2 on track 0, cylinder 0. There are no IPL blocks for the mass storage magazine.

   (The coding in these two blocks will generate an error typeout should an attempt be made to load the volume using the Initial Program Loader.)
4. For disc and drum, a standard Volume label is created and written as record number 3 on track 0, cylinder 0. For mass storage, the standard Volume label is written as record 1 of track 0, cylinder 0.
5. A dummy Volume Table of Contents (VTOC) is created for the volume. (This table is subsequently used to contain a directory of all files stored in the volume, the boundaries for each file, the alternate track area and the areas available for data within the volume.)

   For disc and drum, the VTOC can be placed anywhere on the volume. For the mass storage magazine, the VTOC always occupies cylinder 0, tracks 0-7.

Optional Functions

None.

Input

◆ Input to this routine consists of a random access volume to be initialized and a programmer-prepared Volume parameter which describes how the volume is to be formatted. If desired, multiple volumes can be initialized.

Output

◆ Output of this routine is a random access volume (or volumes) initialized as directed by the input parameters.

Equipment Configuration

Required

◆ Processor (65K)

Console typewriter

Card reader, or Videoscan document reader with card read feature

Random access device (70/564 Disc Unit, 70/565 Drum Unit, or 70/568 Magazine)

Optional

◆ Additional random access devices may be used as input to this routine.

Routine Parameters - General

◆ *VOLIN Parameter*

The VOLIN parameter provides all the information pertinent to preparing an input volume for initialization. One parameter must be supplied for each volume initialized.

END Parameter

This parameter denotes the end of initialization and must immediately follow the last VOLIN parameter.

Routine Parameters - Detailed

Volume Parameter for Disc and Drum

◆ *Format:*

ΔVOLIN,dd,ssssss,aaa,b,c,ddd,e,nnnn,,ONA=xxxxxxxxxx

Entry	Meaning
ΔVOLIN	Parameter identifier.
,dd	Type of device being initialized: dd = two-character mnemonic assigned to the random access device at system generation time.
,ssssss	Serial number for the volume (six characters). This value must be right-justified and zero-filled.
,aaa	Optional. Cylinder number of where VTOC table is to be placed (0-255). See note 2.
,b	Optional. Track number of left-hand end of VTOC (0-9). See note 2.
,c	Optional. Track number of right-hand end of VTOC (0-9). See note 2.

Volume Parameter for Disc and Drum (Cont'd)

Entry	**Meaning**
,ddd	Optional. Cylinder number of right-hand end of alternate track area (0-255). See note 3.
,e	Optional. Track number of the right-hand end of the alternate track area (0-9). See note 3.
,nnnn	Optional. Number of alternate tracks available (0-9999). See note 3.
,	This entry is not applicable for disc and drum, but the comma must appear.
,ONA=xxxxxxxxxx	Optional. Owner-identifier code for the volume, where xxxxxxxxxx is a 10-character alphanumeric field.

Notes:

1. When an entry is omitted, the absence of that entry must be indicated by a comma.

2. If the cylinder number entry for the VTOC is omitted, the routine assigns the VTOC to cylinder 0.

   If the track number entries for the VTOC are omitted, the routine assigns track 0.

3. If the cylinder number entry of the alternate track area is omitted, cylinder 0 is assumed.

   If the right-hand end track number of the alternate track area is omitted, track 0 is assumed.

   If the number of alternate tracks entry (nnnn) is omitted, the routine assumes that no alternate tracks are available.

Examples:

```
ΔVOLIN,C0,000017,,,,127,7,6,,ONA=PAYROLLΔ14

ΔVOLIN,A0,000001,0,1,9,202,9,0010,,ONA=XYZΔΔΔΔΔΔΔ
```

Volume Parameter for Mass Storage

◆ *Format:*

ΔVOLIN,dd,ssssss,,,07,dddd,e,nnnn,m,ONA=xxxxxxxxxx

Entry	Meaning
ΔVOLIN	Parameter identifier.
,dd	Type of device being initialized: dd = two-character mnemonic assigned to the random access device at system generation time.
,ssssss	Serial number for the volume (one to six characters). This value is right-justified and zero-filled if less than six characters.
,,,07	Required. Track number of right-hand end of VTOC.
,dddd	Required. Cylinder number of right-hand end of alternate track area (1-4095).
,e	Required. Track number of right-hand end of alternate track area.
,nnnn	Required. Number of alternate tracks available (0-9999).
,m	Required. Magazine number (0-7).
,ONA-xxxxxxxxxx	Optional. Owner-identification code for the volume, where xxxxxxxxxx is a 10-character alphanumeric field.

Note:

All entries except the last one are required for a mass storage magazine. If the last entry is omitted, however, the comma must appear.

Example:

ΔVOLIN,E0,000777,,,07,4095,7,384,4,ONA=MAGΔDIRECT

END Parameter

◆ This parameter is mandatory and must appear after the final Volume parameter to signify the end of parameter input.

Format:

ΔEND

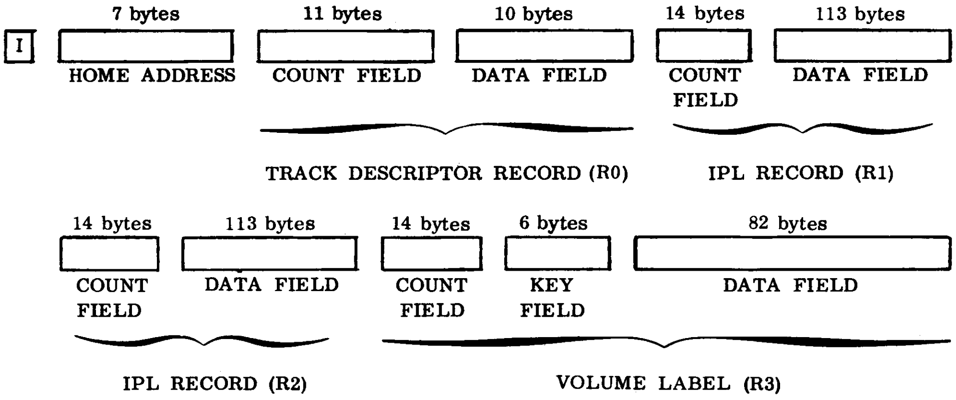
Considerations for Use

◆ As cylinder 0 is reserved in the system for volume label information, the most efficient placement for the VTOC table is also in cylinder 0. The VTOC is always on cylinder 0 for mass storage.

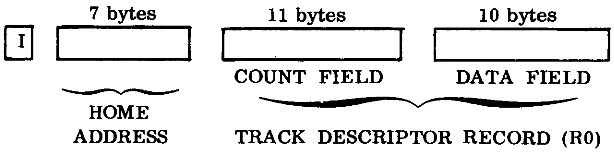
Each random access volume used in the system must be assigned a unique serial number as these numbers are used for device assignment purposes by the Executive.

Track Initialization for Disc and Drum

◆ This routine records the following information on track 0 of cylinder 0:



All other tracks in the volume are formatted as follows:



Track Initialization for Disc and Drum (Cont'd)

The contents of the volume serial number, data file directory, and owner-identification code fields are determined from the Volume parameter at initialization time.

VTOC Table Initialization

◆ The Volume Table of Contents table may contain four types of entries: formats 1, 3, 4, and 5.

Format 1 - This entry consists of a file label containing creation and expiration dates, extent addresses, and other indicative information about the file to which it refers.

Format 3 - This entry is used as a continuation of format 1 when additional file information is necessary.

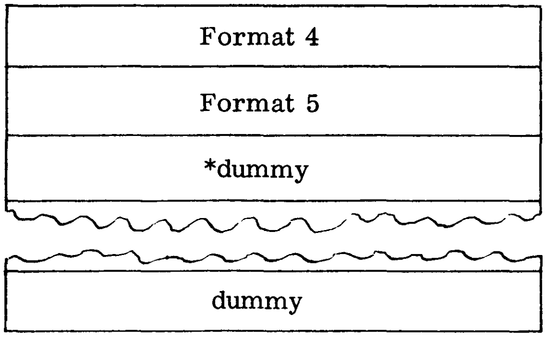
Format 4 - This entry is always the first record in the VTOC table. It describes the size and limits of the VTOC, and gives the location of the alternate track area for the volume.

Format 5 - This entry is always the second record in the VTOC table. It contains the addresses of available extents on the volume.

The RAINIT routine creates a format 4, a format 5, and dummy format 1 entries at initialization time. The Allocator creates format 1 and format 3 entries and updates the format 4 and format 5 entries as required.

After initialization, the VTOC table appears as follows:

VTOC Table



*Dummy format 1 records (filled with binary zeros)

Track Initialization for Mass Storage

◆ For a mass storage volume this routine records a home address and a track descriptor record (as in disc and drum) on each track of the volume. There are no IPL blocks, however, and the standard volume label is R1 on track 0 of cylinder 0.

Track Initialization for Mass Storage (Cont'd)

In addition, the location and format of the VTOC on a mass storage magazine are fixed, and the VTOC will always appear as follows after initialization:

		Cylinder Zero
	Track 0	Volume Label
VTOC	Track 1	Format 4
	Track 2	Format 5
	Track 3	dummy Format 1
	Track 4	dummy Format 1
	Track 5	dummy Format 1
	Track 6	dummy Format 1
	Track 7	dummy Format 1

As illustrated, there may only be five format 1 labels on a mass storage magazine.

Inability to Record Home Address or Track Descriptor Record

◆ If a track descriptor record or home address block cannot be recorded for disc or drum the user is so notified and the volume is deleted from the system.

If a track descriptor record or home address block cannot be recorded for mass storage, the user is notified, but must allow the routine to continue to initialize the remainder of the magazine. The volume label and VTOC will be recorded. The user must replace the cards that were indicated in the error typeouts. These cards may then be initialized by the 70/568 Service Program, thus eliminating the need to reinitialize the entire volume.

Device Assignments

◆ Under Executive Control:

SDN	Device type	Remarks
CDRR01	Card reader.	Parameter input.

Note:

As the type of random access input device is defined in the Volume parameter, it is not necessary to make this device assignment at load time.

Track Initialization for Disc and Drum (Cont'd)

Home Address Block

A standard Home Address block is created for each track.

The flag byte in this block indicates the condition of the track. If bit 0 of this byte is set to 0, the track is a good track; if set to 1 the track is defective. If bit 1 of this byte is set to 0, the track is not an alternate track; if set to 1 the track is an alternate track.

Track Descriptor Record (R0)

A standard Count field is recorded, followed by a 10-byte data field which has the following format:

DATA FIELD

CC	HH	RN	BR	RO	CC CC
2	2	1	2	1	2

CC HH = cylinder and head number.
RN = record number (set to 0).
BR = bytes remaining on track.
RO = record overflow indicator (set to 0).

IPL BLOCKS (R1 and R2)

Not applicable; for system use only.

Volume Label (R3)

The Volume label consists of a standard Count field, a Key field which contains the constant VOL1, and an 80-character block formatted as follows:

Bytes	**Content**
1-4	VOL1
5-10	Volume serial number.
11	Volume security indicator (0).
12-21	Data file directory. (The first five bytes contain the left-hand end address of the VTOC table (CCHHR); the last five bytes are blank.)
22-41	Reserved for future use.
42-51	Owner-identification code.
52-80	Reserved for future use.

CARD TO RANDOM ACCESS (CDRA)

General Description

◆ The Card to Random Access routine transcribes 80-column card records or paper tape to a random access file. Input cards are punched in EBCDIC format, with the final card containing /* in the first two columns to signify the end of the file. The generated output may be single or multivolume.

Preset Functions

This routine is preset for the following functions when the output device is a disc unit:

To copy 80-character records (EBCDIC format) to a random access volume in unblocked format without keys.

To accept multivolume output, provided all volumes are on-line.

To type out record and block counts at the end of the routine.

To provide a console typewriter listing (log) of the input parameters.

To perform a write-disc check.

Optional Functions

The following optional functions are provided and may be specified by the use of parameters:

Selection of drum or mass storage as the output device.

Blocking of output records up to a maximum of 3,600 characters per block for disc, 3,000 for drum, and 2,048 for mass storage.

Field selection, packing, and unpacking of input fields.

Providing keys for output records.

Sequence checking of the input file.

Suppressing the logging of input parameters.

Input

◆ Input to the Card to Random Access routine consists of a card file to be transcribed to a random access file. The last card of the card file must contain /* in the first two columns. If a nonblank character is in column 80, the /* is treated as data.

When optional functions are desired, the parameters that specify these options must be entered before the first input record is read.

Output

◆ Output of this routine consists of the contents of the card input file transcribed to a random access file in the format specified by the parameters. The output may be single or multivolume.

Record and block counts are typed out at the end of this routine. A log of input parameters may also be provided.

Equipment Configuration

Required

◆ Processor (65K).

Console typewriter.

Card reader, or Videoscan document reader with card read feature.

Random access device.

Optional

◆ Additional random access devices may be assigned for multivolume output.

A paper tape reader may be used instead of the card reader. Refer to page 1-4 for paper tape considerations.

Routine Parameters - General

◆ The following three parameters are used with this routine:

Utility-Modifier Parameter

This parameter specifies blocking of output records; field selection; creation of keys; sequence checking of the input cards; logging of input parameters; and the type of output random access device used.

Field-Select Parameter

This parameter specifies field selection, location of key fields, packing, and unpacking of the fields in the input records.

This parameter must be used when the field select operation has been specified in the Utility-Modifier parameter.

DNOJS Parameter

When the routine is running under Monitor, this parameter informs Monitor that the data input is not on SYSIPT.

END Parameter

This parameter signifies the end of parameter input.

Routine Parameters - Detailed

Utility-Modifier Parameter

◆ *Format:*

$$\Delta U\Delta Tx,FF,A=(80,80),B=\left\{\begin{array}{l}(a,b)\\(K=x,D=y)\end{array}\right\},Ox,I1,Q=(x,y),Zx,D=(X,o)$$

Entry	**Meaning**
ΔUΔ	Parameter identifier.
Tx	Function: x = C copy cards to random access in unblocked format (preset). = R copy cards to random access in blocked format. = F field select; copy cards to random access in unblocked format. = RF field select; copy cards to random access in blocked format. Note: Whenever the number of records in the input block differs from the number of records in the output block, this is considered reblocking.
,FF	Input record format: Fixed-length, 80-character records (preset).
,A = (80,80)	Input format: Unblocked, 80-character records (preset).
,B = (a,b) or ,B = (K = x,D = y)	Output format: Fixed-length without keys: a = size of record. b = size of block. Note: Preset values are (80,80). Fixed-length with keys: x = size of key field. y = size of data field.

Utility-Modifier Parameter (Cont'd)

Entry	Meaning
,Ox	Perform disc-write check: x = Y yes (preset). = N no.
,I1	Input mode: EBCDIC card input (preset).
,Q=(x,y)	Input Sequence Check: x = first column in input card to be sequence checked (column 1 = 1). y = length of field to be checked (maximum of 10 characters). Note: Preset function does not provide for sequence checking.
,Zx	Log routine parameters: x = L log (preset). = N do not log.
,D=(X,o)	Output random access device: o = 1 disc (preset). = 2 drum. = 3 mass storage.

Notes:

1. All entries are optional and may appear in any order. When an entry is omitted, the preset value is assumed.
2. All entries, except the first, must be preceded by a comma.
3. Multiple parameter cards may be used; it is not required that all entries appear on a single card.
4. Refer to page 1-5 for parameter requirements with paper tape input.

Examples:

ΔUΔTF,FF,B=(K=10,D=70),Q=(1,10),D=(X,2)

ΔUΔTR,FF,B=(80,400)

Field Select Parameter

◆ *Format:*

ΔFSΔr,s,t/r,s,t/...../r,s,t

Entry	Meaning
ΔFSΔ	Parameter identifier.
r,s,t	r = starting position (relative to one) of the input field to be selected. s = size of the input field. t = starting position (relative to one) of the field in the output record.
r,s,(K,t)	If the input field is to be moved to the Key of the output record, replace entry t in above parameter with (K,t): K = identifies Key. t = starting position (relative to one) of the Key to be moved. Note: r and s remain the same.
(U,n,m)	If the input field is to be unpacked when moved to the output record, replace entry s in the above parameter with (U,n,m): U = identifies unpack operation. n = size of input field in bytes. m = size of output field in bytes. Note: r and t remain the same.
(P,n,m)	If the input field is to be packed when moved to the output record, replace entry s in above parameter with (P,n,m): P = identifies pack operation. n = size of input field in bytes. m = size of output field in bytes. Note: r and t remain the same.

Notes:

1. When a Field Select parameter is used, the Utility-Modifier parameter also must be used.
2. When using field selection, all positions in the output record not filled with input data will be space-filled.
3. More than one field-select card may be used; it is unnecessary to fill a card completely before starting another. All information for a particular field, however, must appear on the same card.

Examples:

ΔFSΔ1,10,(K,1)/11,70,1

ΔFSΔ1,(P,10,6),1/11,70,7

DNOJS Parameter

◆ *Format:*

ΔDNOJS

This parameter informs Monitor that the data input for the routine is not on SYSIPT. The routine will then request the assignment of PRIPT.

END Parameter

◆ *Format:*

Δ END

This parameter signifies the end of parameter input and must be included whenever a Utility-Modifier parameter has been used.

Considerations for Use

◆ 1. A Volume Displacement Card (VDC) parameter must be supplied at run time. See TOS Operator's Guide, Section 2, for format.

The filename entry for this parameter must be ITUTP5F.

The size of the extent matrix generated in this routine allows for 1 volume with 7 extents (104 bytes). If a larger extent matrix is desired, its size can be specified in the VDC parameter.

2. If the serial number of the volume to be processed does not appear in the On-Line Catalog, the EΔOLC console routine must be run first.

3. All random access output volumes must be initialized by the RAINIT routine and allocated by the RAALLR routine.

4. This routine does not transcribe data on the first track of the file area. (This track is reserved in the system for user header and trailer information.)

Device Assignments

◆ *Under Executive Control:*

SDN	Device Type	Remarks
PRIPT	Card reader or paper tape reader. *	Input data.
PRPRM	Card reader, magnetic tape, or paper tape reader.	Parameter input.

Under Monitor Control:

SDN	Device Type	Remarks
SYSIPT	Card reader or magnetic tape.	Parameter and card data or card image input.

*Refer to page 1-4 for paper tape input considerations.

CARD TO RANDOM ACCESS/MASS STORAGE (CDRAM)

General Description

◆ This routine performs the same functions as the CDRA routine, and uses the same parameter formats and the same typeouts. Nevertheless, due to the nature of the Model 70/568 Mass Storage Unit, this routine (CDRAM) has been enhanced to include Dynamic Alternate Track Assignment, a feature of FCP, which should increase the usefulness of this peripheral routine for 70/568 users.

This enhancement increases the memory requirements of the routine by approximately 4,000 bytes. (See Appendix D, Memory Requirements.)

It is not required that the CDRAM version be used with the 70/568, nor that the CDRA version be used with disc or drum. It is recommended, however, that CDRAM be used with 70/568.

The routine's preset option expects the output device to be a 70/564 disc. Ⓐ

Device Assignments *(Cont'd)*

SDN	Device Type	Remarks
PRIPT	Paper tape reader or card reader. *	Paper tape or card data input (if not on SYSIPT).

Parameter Examples

◆ 1. To copy a card file to disc utilizing the standard preset functions:

```
//ΔVDCΔITUTP5F,,file identification,serial number

//ΔEND

(data cards)

/*
```

Note:

Although routine parameters are not required, the VDC card must be provided for label checking.

2. To copy a card file to disc with card columns 1-5 becoming the Key field and column 6-80 becoming the Data field of the output record:

```
//ΔVDCΔITUTP5F,,file identification,serial number

//ΔEND

ΔUΔTF,FF,A=(80,80),B=(K=5,D=75)

ΔFSΔ1,5,(K,1)/6,75,1

ΔEND

(data cards)

/*
```

3. To copy a card file to drum without Keys, five cards per block, sequence checking the first two card columns:

```
//ΔVDCΔITUTP5F,,file identification,serial number

//ΔEND

ΔUΔTR,FF,A=(80,80),B=(80,400),Q=(1,2),D=(X,2)

ΔEND

(data cards)

/*
```

TAPE TO RANDOM ACCESS (TPRA)

General Description

◆ The Tape to Random Access routine transcribes data from magnetic tape to a random access file.

Input tape blocks can range in size from 12 to 4,096 characters and may contain fixed-length records, variable-length records, or records of undefined size. Except for records of undefined size, records may be blocked or unblocked.

Output records may be blocked to a maximum size of 3,600 characters for disc, 3,000 for drum, and 2,048 for mass storage; and may contain fixed-length records, variable-length records, or records of undefined size. Except for records of undefined size, records may be blocked or unblocked. Keys may be created for fixed-length output only.

The generated output file may be single or multivolume, provided all volumes are on-line.

Preset Functions

This routine is preset for the following functions when the output device is a disc unit:

To copy a magnetic tape file of undefined records (up to 1,000 characters to a random access volume without Keys.

To rewind the input tape to BOT at the start and end of transcription.

To check standard header and trailer labels on the input tape.

To accept multivolume output, provided all volumes are on-line.

To type out record and block counts at the end of the routine.

To provide a console typewriter listing (log) of the input parameters.

To perform a write-disc check.

Optional Functions

The following optional functions may be specified by the use of parameters:

Selection of drum or mass storage as the output device.

Reblocking of fixed-length or variable-length record input.

General Description *(Cont'd)*

Fixed or variable-length record processing.

Field selection, packing, and unpacking of input fields in fixed-length records.

Processing of unlabeled, single-volume input files.

Positioning and disposition of input tape.

Providing key fields for output records.

Blocking of output records to a maximum of 3,600 characters per block for disc, 3,000 for drum, and 2,048 for mass storage.

Suppressing the logging of input parameters.

Input

◆ Input to this routine consists of a labeled (single or multivolume) or unlabeled (single volume) tape file containing fixed-length, variable-length, or undefined records.

Fixed- or variable-length records may be blocked or unblocked. If blocked, the last block may be short. Undefined records must be unblocked.

When optional functions are desired, the parameters that select these options must be entered from the parameter input device.

Output

◆ Output of this routine consists of the contents of the tape input file transcribed to a random access file in the format specified by the parameters. Output may be single or multivolume. If multivolume output is desired, all volumes must be on-line.

Record and block counts are typed out at the end of this routine, a log of input parameters may also be provided.

Equipment Configuration

Required

◆ Processor (65K).

Console typewriter.

Card reader, or Videoscan document reader with card read feature.

Magnetic tape device.

Random access device.

Optional

◆ Additional magnetic tape devices may be assigned for multivolume input.

Additional random access devices may be assigned for multivolume output.

Routine Parameters - General

◆ The following three parameters are used with this routine:

Utility-Modifier Parameter

This parameter specifies blocking and reblocking of input records; record format; block and record sizes; field selection; positioning and disposition of input tape; creation of keys; logging of parameters; and the type of output random access device used.

Field-Select Parameter

This parameter specifies field selection, location of key field, packing, and unpacking of fields in the input records.

This parameter must be used when the field-select operation has been specified in the Utility-Modifier parameter.

END Parameter

This parameter signifies the end of parameter input.

Routine Parameters - Detailed

Utility-Modifier Parameter

◆ *Format*

$$\Delta U \Delta Tx,Fx,A = \left\{ \begin{matrix} (n,m) \\ (g) \end{matrix} \right\} ,B = \left\{ \begin{matrix} (a,b) \\ (K=x,D=y) \\ (n) \end{matrix} \right\}$$

$$,Ox,Ix,W = \left\{ \begin{matrix} (i,X) \\ (ixx,X00) \end{matrix} \right\} ,L=(i,X),Zx,D=(X,o)$$

Entry	Meaning
ΔUΔ	Parameter identifier.
Tx	Function: x = C copy input tape file to random access (preset). = R reblock fixed-length input tape to random access. = F field select; copy tape to random access. = RF reblock and field select fixed-length tape input to random access. Note: Whenever the number of records in input block differs from the number of records in output block, this is considered reblocking.
,Fx	Input record format: x = U undefined records (preset). = F fixed-length records. = V variable-length records.

Utility-Modifier Parameter (Cont'd)

Entry	Meaning
,A = (n,m) or ,A = (g)	Input format: Fixed-length records: n = size of record. m = size of block. Variable or undefined records: g = maximum size of block (preset to 1,000). Note: Maximum input block size is 4,096.
,B = (a,b) or ,B = (K = x,D = y) or ,B = (n)	Output format: Fixed-length without keys: a = size of record. b = size of block. Fixed-length with keys: x = size of key field. y = size of data field. Variable or undefined: n = maximum size of block (preset to 1,000).
,Ox	Perform write-disc check: x = Y yes (preset). = N no.
,Ix	Disposition of input tape on termination: x = R rewind (preset). = U rewind and unload. = N do not rewind. = M multivolume input; rewind and unload.
,W = (i,X) or ,W = (ixx,X00)	Initial positioning of input tape: i = R rewind input tape (preset). = N do not rewind input tape. xx = number of tape marks to be unwound after the tape has been positioned (01-99).
,L = (i,X)	Tape labeling: i = X check label against TPLAB parameter (preset). = N unlabeled input tape.
,Zx	Log routine parameters: x = L log (preset). = N do not log.
,D = (X,o)	Output random access device: o = 1 disc (preset). = 2 drum. = 3 mass storage.

Utility-Modifier Parameter (Cont'd)

Notes:

1. All entries are optional and may appear in any order. If an entry is omitted, the preset value is assumed.
2. All entries, except the first, must be preceded by a comma.
3. Multiple parameter cards may be used; it is not required that all entries appear on a single card.

Examples:

ΔUΔTRF,FF,A = (50,300),B = (K = 10,D = 40),D = (X,2)

ΔUΔTF,FF,A = (100,100),B = (100,100),L = (N,X)

Field-Select Parameter

◆ *Format:*

ΔFSΔr,s,t/r,s,t/...../r,s,t

Entry	**Meaning**
ΔFSΔ	Parameter identifier.
r,s,t	r = starting position (relative to one) of the <u>input</u> field to be selected. s = size of the input field. t = starting position (relative to one) of the field in the <u>output</u> record.
r,s,(K,t)	If the input field is to be moved to the Key of the output record, replace entry t in above parameter with (K,t): K = identifies Key. t = starting position (relative to one) in the Key field. <u>Note:</u> r and s remain the same.
(U,n,m)	If the input field is to be unpacked when moved to the output record, replace entry s in the above parameter with (U,n,m): U = identifies unpack operation. n = size of input field in bytes. m = size of output field in bytes. <u>Note:</u> r and t remain the same.
(P,n,m)	If the input field is to be packed when moved to the output record, replace entry s in above parameter with (P,n,m): P = identifies pack operation. n = size of input field in bytes. m = size of output field in bytes. <u>Note:</u> r and t remain the same.

Field-Select Parameter (Cont'd)

Notes:

1. When a Field Select parameter is used, the Utility-Modifier parameter must also be used.
2. When using field selection, all positions in the output record not filled with input data will be space-filled.
3. More than one field-select card may be used; it is unnecessary to fill a card completely before starting another. All information for a particular field, however, must appear on the same card.

Examples:

ΔFSΔ11,40,1/1,10,(K,1)

ΔFSΔ1,(U,2,3),30

END Parameter

◆ *Format:*

ΔEND

This parameter signifies the end of parameter input and must be included whenever a Utility-Modifier parameter has been used.

Considerations for Use

◆ 1. A Volume Displacement Card (VDC) parameter must be supplied at run time. See TOS Operator's Guide, section 2, for format.

The filename entry for this parameter must be ITUTP5F.

The size of the extent matrix generated in this routine allows for 1 volume with 7 extents (104 bytes). If a larger extent matrix is desired, its size can be specified in the VDC parameter.

2. If the serial number of the volume to be processed does not appear in the On-Line Catalog, the EΔOLC console routine must be run first.
3. All random access output volumes must be initialized by the RAINIT routine and allocated by the RAALLR routine.
4. This routine does not transcribe data to the first track of the file area. (This track is reserved in the system for user header and trailer information.)
5. When standard tape labels are to be checked, VOL and TPLAB cards must be supplied. See TOS Operator's Guide, section 2, for format.

The filename entry in the VOL card for the input file must be ITUTP2F.

Device Assignments

◆ *Under Executive Control*

SDN	Device Type	Remarks
PRIPT1	Magnetic tape.	Primary input device.
PRIPT2	Magnetic tape.	Alternate input device.

TAPE TO RANDOM ACCESS/MASS STORAGE (TPRAM)

General Description

◆ This routine performs the same functions as the TPRA routine, and uses the same parameter formats and the same typeouts. Nevertheless, due to the nature of the Model 70/568 Mass Storage Unit, this routine (TPRAM) has been enhanced to include Dynamic Alternate Track Assignment , a feature of FCP, which should increase the usefulness of this peripheral routine for 70/568 users.

This enhancement increases the memory requirements of the routine by approximately 4,000 bytes. (See Appendix D, Memory Requirements.)

It is not required that the TPRAM version be used with the 70/568, nor that the TPRA version be used with disc or drum. It is recommended, however, that TPRAM be used with 70/568.

Device Assignments *(Cont'd)*

SDN	Device Type	Remarks
PRPRM	Card reader, magnetic tape, or paper tape reader.	Parameter input device.

Under Monitor Control

SDN	Device Type	Remarks
PRIPT1	Magnetic tape.	Primary input device.
PRIPT2	Magnetic tape.	Alternate input device.
SYSIPT	Card reader or magnetic tape.	Parameter input device.

Note:

As the random access device is defined in the VDC card, it is not necessary to make this assignment at load time.

Parameter Examples

◆ 1. To copy undefined tape records to a random access volume on disc using the preset functions:

Routine parameters are not required. Nevertheless, the following cards must be supplied for tape label checking and to specify the serial number of the random access volume:

```
//ΔVOLΔsysxxx,ITUTP2F
//ΔTPLABΔ....
//ΔVDCΔITUTP5F,,file identification, serial number
//ΔEND
```

2. To copy a blocked tape file to disc in unblocked format. Positions 96-100 of each input record contain data for the Key field, and positions 1-95 contain the Data field portion:

```
{//ΔVOLΔsysxxx,ITUTP2F }  include if label checking
{//ΔTPLABΔ....         }  is desired
//ΔVDCΔITUTP5F,,file identification, serial number
//ΔENDΔ
ΔUΔTRF,FF,A=(100,400),B=(K=5,D=95)
ΔFSΔ96,5,(K,1)/1,95,1
ΔEND
```

3. To copy an unlabeled, unblocked tape file to drum without keys:

```
//ΔVDCΔITUTP5F,,file identification, serial number
//ΔENDΔ
ΔUΔTC,FF,A=(100,100),B=(100,100),L=(N,X),D=(X,2)
ΔEND
```

RANDOM ACCESS TO RANDOM ACCESS (RARA)

General Description

◆ The Random Access to Random Access routine transcribes data from one random access file to another. The input and output records may be fixed-length (with or without keys), variable-length, or of undefined size. They may be blocked to a maximum size of 3,600 characters for disc, 3,000 for drum, and 2,048 for mass storage.

In the case of fixed-length records, input fields may be field-selected, packed, or unpacked during the copying process.

Multivolume input and output are allowed, provided all volumes are on-line.

Preset Functions

This routine is preset for the following functions when the input and output devices are <u>disc</u> units:

To copy an input file containing undefined records (up to 1,000 characters) to the output file. If input records contain Key fields, the key data is placed at the beginning of the output Data field. A separate output Key field is <u>not</u> generated.

To check standard random access labels.

To accept multivolume input and output, provided all volumes to be processed are on-line.

To perform a write-disc check.

To type out record and block counts at the end of the routine.

To provide a console typewriter listing (log) of the input parameters.

Optional Functions

The following optional functions may be specified by the use of parameters:

Selection of drum or mass storage as the input and/or output devices.

Reblocking of fixed-length record input.

Fixed-or variable-length record processing.

Copying fixed-length records with or without keys to the output file.

Suppressing the transfer of key fields to the output file.

Field selection, packing, and unpacking of input fields in fixed-length records.

Blocking of output records to a maximum of 3,600 characters per block for disc, 3,000 for drum, and 2,048 for mass storage.

Suppressing the logging of input parameters.

Input

◆ Input to this routine consists of a random access file containing fixed-length records (with or without keys), variable-length records, or records of undefined size.

Records with keys may not be blocked. Other fixed-length and variable-length records may be blocked or unblocked. If blocked, the last block may be short. Undefined records must be unblocked.

When optional functions are desired, the parameters that select these options must be entered from the parameter input device.

Output

◆ Output of this routine consists of data transcribed from one random access file to another random access file in the format specified by the parameters. Output may be single or multivolume. If multivolume input or output is desired, all volumes must be on-line.

Record and block counts are typed out at the end of this routine; a log of input parameters also can be provided.

Equipment Configuration

Required

◆ Processor (65K)

Console typewriter

Card reader or Videoscan document reader with card read feature

Random access device

Optional

◆ Additional random access devices may be assigned for multivolume input or output.

Routine Parameters - General

◆ The following three parameters are used with this routine:

Utility-Modifier Parameter

This parameter specifies blocking and reblocking of input records; record format; block and record sizes; field selection; processing of keys; logging of parameters; and the type of input and output random access devices.

Field-Select Parameter

This parameter specifies field selection, packing, and unpacking of fields in the input record. It can also be used to specify that input record Key fields be field selected into the output record, or that input record Key fields be eliminated from the output record.

This parameter must be used when the field select operation has been specified in the Utility Modifier parameter.

END Parameter

This parameter signifies the end of parameter input.

Routine Parameters - Detailed

Utility-Modifier Parameter

◆ *Format:*

$$\Delta U \Delta Tx, Fx, A = \left\{\begin{matrix}(n,m)\\(K=x,D=y)\\(g)\end{matrix}\right\}, B = \left\{\begin{matrix}(a,b)\\(K=x,D=y)\\(h)\end{matrix}\right\}, Ox, Zx, D=(i,o)$$

Entry	**Meaning**
ΔUΔ	Parameter identifier.
Tx	Function: x = C copy input random access file to output random access file (preset). = R reblock fixed-length record input. = RF reblock and field-select fixed-length record input. Note: Whenever the number of records in the input block differs from the number of records in the output block, this is considered reblocking.
, Fx	Input record format: x = U undefined records (preset). = F fixed-length records. = V variable-length records.
, A = (n, m) or , A = (K=x, D=y) or , A = (g)	Input format: Fixed-length without keys: n = size of record. m = size of block. Fixed-length with keys: x = size of key field. y = size of data field. Variable or undefined: g = maximum size of block (preset to 1,000).
, B = (a, b) or , B = (K=x, D=y) or , B = (h)	Output format: Fixed-length without keys: a = size of record. b = size of block. Fixed-length with keys: x = size of key field. y = size of data field. Variable or undefined: h = maximum size of block (preset to 1,000)

Utility-Modifier Parameter (Cont'd)

Entry	Meaning
,Ox	Perform write-disc check: x = Y yes (preset). = N no.
,Zx	Log routine parameters: x = L log (preset). = N do not log.
,D=(i,o)	Input/output random access device: i = 1 disc input (preset). = 2 drum input. = 3 mass storage input. o = 1 disc output (preset). = 2 drum output. = 3 mass storage output.

Notes:

1. All entries are optional and may appear in any order. When an entry is omitted, the preset value is assumed.
2. All entries, except the first, must be preceded by a comma.
3. Multiple parameter cards may be used; it is not required that all entries appear on a single card.

Examples:

```
ΔUΔTF,FF,A=(K=10,D=40),B=(K=10,D=40),D=(1,3)
ΔUΔTR,FF,A=(50,50),B=(50,300)
```

Field-Select Parameter

◆ *Format:*

ΔFSΔr,s,t/r,s,t/..../r,s,t

Entry	Meaning
ΔFSΔ	Parameter identifier.
r,s,t	r = starting position (relative to one) of the input field to be selected. s = size of the input field. t = starting position (relative to one) of the field in the output record.
(K,r),s,t	If the key of the input record is to be moved to the output record, replace entry r in the above parameter with (K,r): K = identifies Key. r = starting position in the input Key field. Note: s and t remain the same.
r,s,(K,t)	If the input field is to be moved to the Key of the output record, replace entry t in the above parameter with (K,t): K = identifies Key. t = starting position in the output record key. Note: r and s remain the same.
(U,n,m)	If the input field is to be unpacked when moved to the output record, replace entry s in the above parameter with (U,n,m): U = identifies unpack operation. n = size of input field in bytes. m = size of output field in bytes. Note: r and t remain the same.
(P,n,m)	If the input field is to be packed when moved to the output record, replace entry s in above parameter with (P,n,m): P = identifies pack operation. n = size of input field in bytes. m = size of output field in bytes. Note: r and t remain the same.

Notes:

1. When a Field Select parameter is used, the Utility-Modifier parameter also must be used.

Field-Select Parameter (Cont'd)

2. When using field selection, all positions in the output record not filled with input data will be space-filled.
3. More than one field select card may be used; it is unnecessary to fill a card completely before starting another. All information for a particular field, however, must appear on the same card.

Examples:

ΔFSΔ(K,1),10,(K,1)/1,40,1

ΔFSΔ46,5,1/1,45,6

END Parameter

◆ *Format:*

ΔEND

This parameter signifies the end of parameter input and must be included whenever a Utility Modifier parameter has been used.

Considerations for Use

◆ 1. A VDC parameter must be supplied for both input and output volumes at run time. See TOS Operator's Guide, section 2, for format.

The filename entry for this parameter for the input file is ITUTP4F.

The filename entry for this parameter for the output file is ITUTP5F.

The size of the extent matrix generated in this routine allows for 1 volume with 7 extents (104 bytes). If a larger extent matrix is desired, its size can be specified in the VDC parameter.

2. If the serial numbers of the volumes to be processed do not appear in the On-Line Catalog, the EΔOLC console routine must be run first.
3. All random access input and output volumes must be initialized by the RAINIT routine and allocated by the RAALLR routine.
4. This routine does not transcribe from or to the first track of the file area. (This track is reserved in the system for user header and trailer information.)

Device Assignments

◆ *Under Executive Control*

SDN	Device Type	Remarks
PRPRM	Card reader, magnetic tape, or paper tape reader.	Parameter input device.

Under Monitor Control

SDN	Device Type	Remarks
SYSIPT	Card reader or magnetic tape.	Parameter input device.

Note:

As the random access devices are defined by VDC cards, it is not necessary to make these assignments at load time.

RANDOM ACCESS TO RANDOM ACCESS/MASS STORAGE (RARAM)

General Description

◆ This routine performs the same functions as the RARA routine, and uses the same parameter formats and the same typeouts. Nevertheless, due to the nature of the Model 70/568 Mass Storage Unit, this routine (RARAM) has been enhanced to include Dynamic Alternate Track Assignment, a feature of FCP, which should increase the usefulness of this peripheral routine for 70/568 users.

This enhancement increases the memory requirements of the routine by approximately 4,000 bytes. (See Appendix D, Memory Requirements.)

It is not required that the RARAM version be used with the 70/568, nor that the RARA version be used with disc or drum. It is recommended, however, that RARAM be used with 70/568.

Parameter Examples

◆ 1. To copy a disc file of undefined records to another disc file:

   Routine parameters are not required. Nevertheless, the following cards must be supplied to specify the serial numbers of the random access volumes:

```
//ΔVDCΔITUTP4F,,file identification,serial number
//ΔVDCΔITUTP5F,,file identification, serial number
//ΔEND
```

2. To reblock a disc file of fixed-length input from 10 to 5 records per block to a drum:

```
//ΔVDCΔITUTP4F,,file identification,serial number
//ΔVDCΔITUTP5F,,file identification,serial number
//ΔEND
ΔUΔTR,FF,A=(50,500),B=(50,250),D=(1,2)
ΔEND
```

3. To copy a blocked disc file to another disc file in unblocked format with positions 45-50 of the input record becoming the Key field of the new file.

```
//ΔVDCΔITUTP4F,,file identification,serial number
//ΔVDCΔITUTP5F,,file identification,serial number
//ΔEND
ΔUΔTRF,FF,A=(50,500),B=(K=5,D=45)
ΔFSΔ46,5,(K,1)/1,45,1
ΔEND
```

4. To copy unblocked records with keys from one disc file to another. Records on the receiving file are to be blocked five records per block, with the original key placed in positions 46-50:

```
//ΔVDCΔITUTP4F,,file identification, serial number
//ΔVDCΔITUTP5F,,file identification, serial number
//ΔEND
ΔUΔTRF,FF,A=(K=5,D=45),B=(50,500)
ΔFSΔ(K,1),5,46/1,45,1
ΔEND
```

RANDOM ACCESS TO TAPE (RATP)

General Description

◆ The Random Access to Tape routine transcribes data from a random access file to a magnetic tape.

Input random access records may be blocked to a maximum size of 3,600 characters for disc, 3,000 for drum, or 2,048 for mass storage. They may contain fixed-length records, variable-length records, or records of undefined size. Fixed-length records may be blocked or unblocked, with or without Key fields.

Output tape blocks can range in size from 12 to 4,096 characters, and may contain fixed-length records, variable-length records, or records of undefined size. In the case of fixed-length records, records may be blocked or unblocked. Tape volumes may be labeled or unlabeled, single or multivolume.

Preset Functions

This routine is preset for the following functions when the input device is a disc unit:

To copy an input file containing undefined records (up to a maximum size of 1,000 characters) to an output magnetic tape. If input records contain Key fields, the key data is placed at the beginning of the output data field. A separate key field is not generated.

To rewind the output tape to BOT before the copying process; to rewind and unload the output tape at the end of the copying process.

To write standard header and trailer labels on the output tapes.

To alternate tapes for multivolume output when two tape devices are available.

To accept multivolume input, provided all volumes are on-line.

To type out record and block counts at the end of the routine.

To provide a console typewriter listing (log) of the input parameters.

Optional Functions

The following optional functions may be specified by the use of parameters:

Selection of drum or mass storage as the input device.

Reblocking of fixed-length or variable-length record input.

Copying of fixed-length records (with or without keys) to the output tape.

Field-selection, packing, or unpacking of fields in fixed-length records.

Positioning and disposition of output tape.

Suppressing the logging of input parameters.

Suppressing the transfer of key fields to the output tape volume.

Accepting blocked, fixed-length, or variable-length records up to a maximum of 3,600 characters per block for disc, 3,000 for drum, or 2,048 for mass storage.

Input

◆ Input to this routine consists of a random access file containing fixed-length records (with or without keys), variable-length records, or undefined records.

Fixed-length or variable-length records may be unblocked or blocked. If blocked, the last block may be short. Undefined records must be unblocked.

When optional functions are desired, the parameters that select these options must be entered from the parameter input device.

Output

◆ Output of this routine consists of a labeled (single or multivolume) or unlabeled (single volume) tape file. Fixed-length or variable-length records may be blocked, unblocked, or reblocked; records of undefined length must always be unblocked.

Record and block counts are typed out at the end of this routine; a log of input parameters may also be provided.

Equipment Configuration

Required

◆ Processor (65K)

Console typewriter

Card reader, or Videoscan document reader with card read feature

Magnetic tape device

Random access device

Optional

◆ Additional magnetic tape devices may be assigned for multivolume output.

Additional random access devices may be assigned for multivolume input.

Routine Parameters - General

◆ The following three parameters are used with this routine:

Utility-Modifier Parameter

This parameter specifies blocking and reblocking of input records, record format; block and record sizes; field selection; positioning and disposition of output tapes; logging of parameters; generation of unlabeled output volumes; and the type of input random access device used.

Routine Parameters - General
(Cont'd)

Field-Select Parameter

This parameter specifies field selection, packing, and unpacking of fields in the input record. It also can be used to move the Key field into the output record.

This parameter must be used when the field-select operation has been specified by a Utility-Modifier parameter.

END Parameter

This parameter signifies the end of parameter input.

Routine Parameters - Detailed

Utility-Modifier Parameter

◆ *Format:*

$$\Delta U \Delta Tx, Fx, A = \left\{ \begin{matrix} (n,m) \\ (K=x, D=y) \\ (g) \end{matrix} \right\}, B = \left\{ \begin{matrix} (a,b) \\ (h) \end{matrix} \right\}, Ox, W = \left\{ \begin{matrix} (X,o) \\ (X00, oxx) \end{matrix} \right\},$$

$$L=(X,o), Zx, D=(i,X)$$

Entry	Meaning
ΔUΔ	Parameter identifier.
Tx	Function: x = C copy input file to output tape (preset). = R reblock fixed-length or variable-length input records. = F field select fixed-length record input. = RF reblock and field select fixed-length record input. Note: Whenever the number of records in the input block differs from the number of records in the output block, this is considered reblocking.
, Fx	Input record format: x = U undefined (preset). = F fixed-length. = V variable-length.

Utility-Modifier Parameter (Cont'd)

Entry	Meaning
,A=(n,m) or ,A=(K=x,D=y) or ,A=(g)	Input format: Fixed-length without keys: n = size of record. m = size of block. Fixed-length with keys: x = size of key field. y = size of data field. Variable or undefined: g = maximum size of block (preset to 1,000).
,B=(a,b) or ,B=(h)	Output format: Fixed-length records: a = size of record. b = size of block. Variable or undefined records: h = maximum size of block (preset to 1,000). Note: Maximum size of output blocks is 4,096.
,Ox	Disposition of output tape on termination: x = R rewind. = U rewind and unload (preset). = N do not rewind but leave positioned after double tape marks.
,W=(X,o) or ,W=(X00,oxx)	Initial positioning of output tape: o = R rewind (preset). = N do not rewind. xx = number of tape marks to be unwound after the tape has been positioned (01-99).
,L=(X,o)	Tape labeling: o = X label supplied by TPLAB parameter (preset). = N unlabeled output tape with leading TM. = T unlabeled output tape without leading TM.
,Zx	Log routine parameters: x = L log (preset). = N do not log.
,D=(i,X)	Input random access device. i = 1 disc (preset). = 2 drum. = 3 mass storage.

Utility-Modifier Parameter (Cont'd)

Notes:

1. All entries are optional and may appear in any order. If any entry is omitted, the preset value is assumed.
2. All entries, except the first, must be preceded by a comma.
3. Multiple parameter cards may be used; it is not required that all entries appear on a single card.

Examples:

```
ΔUΔTC,FU,A=(1030),B=(1030)L=(X,X)
ΔUΔTRF,FF,A=(K=10,D=50),B=(60,300),ON,ZN
ΔUΔTF,FF,A=(K=5,D=45),B=(50,50),L=(X,T),D=(2,X)
```

Field-Select Parameter

◆ *Format:*

```
ΔFSΔr,s,t/r,s,t/...../r,s,t
```

Entry	Meaning
ΔFSΔ	Parameter identifier.
r,s,t	r = starting position (relative to one) of the <u>input</u> field to be selected. s = size of the input field. t = starting position (relative to one) of the field in the <u>output</u> record.
(K,r),s,t	If the Key field is to be moved into the output record, replace r in above parameter with (K,r): K = identifies Key. r = starting position (relative to one) of the Key to be moved. <u>Note:</u> s and t remain the same.
(U,n,m)	If the input field is to be unpacked when moved to the output record, replace entry s in the above parameter with (U,n,m): U = identifies unpack operation. n = size of input field in bytes. m = size of output field in bytes: <u>Note:</u> r and t remain the same.
(P,n,m)	If the input field is to be packed when moved to the output record, replace entry s in above parameter with (P,n,m): P = identifies pack operation. n = size of input field in bytes. m = size of output field in bytes. <u>Note:</u> r and t remain the same.

Field-Select Parameter (Cont'd)

Notes:

1. When a Field Select parameter is used, the Utility-Modifier parameter must also be used.
2. When using field selection, all positions in the output record not filled with input data will be space-filled.
3. More than one field-select card may be used; it is unnecessary to fill a card completely before starting another. All information for a particular field, however, must appear on the same card.

Examples:

ΔFSΔ(K,1),10,1/1,50,11

ΔFSΔ46,(U,5,9),1/1,45,10

ΔFSΔ18,(P,30,16),41

END Parameter

◆ *Format:*

ΔEND

This parameter signifies the end of parameter input and must be included whenever a Utility-Modifier parameter has been used.

Considerations for Use

◆ 1. A VDC parameter must be supplied at runtime. See TOS Operator's Guide, section 2, for format.

The filename entry for this parameter must be ITUTP4F.

The size of the extent matrix generated in this routine allows for 1 volume with 7 extents (104 bytes). If a larger extent matrix is desired, its size can be specified in the VDC parameter.

2. If the serial number of the volume to be processed does not appear in the On-Line Catalog, the EΔOLC console routine must be run first.
3. All random access input volumes must be initialized by the RAINIT routine and allocated by the RAALLR routine.
4. This routine does not transcribe to the output tape the first track of the file area. (This track is reserved in the system for user header and trailer label information.)
5. When standard tape labels are to be generated, VOL and TPLAB cards must be supplied. See TOS Operator's Guide, section 2, for format.

The filename entry in the VOL card for the output file must be ITUTP3F.

Device Assignments

◆ *Under Executive Control*

SDN	Device Type	Remarks
PROPT1	Magnetic tape.	Primary output.
PROPT2	Magnetic tape.	Alternate output.
PRPRM	Card reader, magnetic tape, or paper tape reader.	Parameter input.

Under Monitor Control

SDN	Device Type	Remarks
PROPT1	Magnetic tape.	Primary output.
PROPT2	Magnetic tape.	Alternate output.
SYSIPT	Card reader or magnetic tape.	Parameter input.

Note:

As the random access device is defined in the VDC card, it is not necessary to make this assignment at load time.

Parameter Examples

◆ 1. To copy undefined records from disc to an output tape utilizing the preset functions:

Routine parameters are not required. Nevertheless, the following cards must be supplied for tape label generation and to specify the serial number of the random access volume:

```
//ΔVOLΔsysxxx,ITUTP3F
//ΔTPLABΔ....
//ΔVDCΔITUTP4F,,file identification, serial number
//ΔENDΔ
```

2. To copy fixed-length disc records with keys to magnetic tape, blocking them five to the block:

```
{//ΔVOLΔsysxxx,ITUTP3F}  include if label generation is
{//ΔTPLABΔ            }  desired.
//ΔVDCΔITUTP4F,,file identification, serial number
//ΔEND
ΔUΔTRF,FF,A=(K=10,D=90),B=(100,500)
ΔFSΔ(K,1),10,1/1,90,11
ΔEND
```

RANDOM ACCESS TO PRINTER AND/OR PUNCH (RAPR)

General Description

◆ The Random Access to Printer and/or Punch routine transcribes data from a random access file to punched cards or paper tape and/or to the printer.

Card output files are punched in EBCDIC with the final card containing /* in the first two columns to signify the end of file.

Printed output may be in Character mode (EBCDIC graphics) or Hexadecimal mode (two digits per character); the print format may be List or Display.

List Format

Output is restricted to one print line (132 or 160 print positions) of data per input record transcribed.

This format may be specified with or without field selection. When the field-select operation is not specified, data is printed in the Character mode and is not spaced between print positions. When the field-select operation is specified, all selected data is printed in the mode (Character or Hexadecimal) and the print positions specified by the Field-Select parameters.

Headings are printed only when specified.

Display Format

The complete input record is printed, regardless of its length and the number of print lines required.

A scale line is printed at the top and bottom of each page.

The first print line for a record contains the block number, record number within the block, block size, and record size in the first 30 positions. Record data begins in position 31 of each print line.

No Field-Select parameters are permitted; however, Character or Hexadecimal mode may be specified in the Utility-Modifier parameter. When Character mode is specified, the print line is edited to space between each set of 10 data characters. When Hexadecimal mode is specified, the print line is edited to space between each set of four print characters (two bytes).

General Description *(Cont'd)*

Preset Functions

This routine is preset for the following functions when the input device is a disc unit:

To process undefined random access input records of up to 1,000 characters.

To select all input records (including keys, if present) and print them on a 132-character print line, in Display format and Character mode, single-spaced, with page numbers.

To type out record counts, block counts, and number of pages printed at the end of the routine.

To provide a console typewriter listing (log) of the input parameters.

To accept multivolume input, provided all volumes are on-line.

Optional Functions

The following optional functions may be specified by the use of parameters:

Selection of drum or mass storage as the input device.

Punching only of all input records.

Printing and punching of all or selected input records.

Printing only or punching only selected records of the input file.

Hexadecimal print mode or EBCDIC punch mode.

Printing in List format; double or triple spacing; suppression of page numbers.

Sequence numbering of output cards.

Field-selection for fixed-length records.

Packing, unpacking, and conversion to Hexadecimal mode specific fields of field-length records.

Printing and punching of Key fields of fixed-length records.

Providing the text for print header lines.

Suppressing the logging of input parameters.

Input

◆ Input to this routine consists of a random access file (single or multivolume) containing fixed-length, variable-length, or undefined records to be transcribed to the printer and/or the card punch.

Fixed- or variable-length records may be blocked or unblocked; if blocked, the last block may be short. Fixed-length unblocked records may have Keys. Undefined records must be unblocked.

When optional functions are desired, the parameters that specify these options must be entered from the parameter input device.

Output

◆ Output consists of the contents of a random access volume transcribed to punched cards and/or printed in the format specified by the parameters. The last card punched contains /* in the first two columns.

Equipment Configuration

Required

◆ Processor (65K).

Console typewriter.

Card punch.

Card reader, or Videoscan document reader with card read feature.

Printer.

Random access device.

Optional

◆ Additional random access devices may be used for multivolume input.

The paper tape punch may be used instead of the card punch.

Routine Parameters - General

◆ The following six parameters are used with this routine.

Utility-Modifier Parameter

This parameter specifies the input device; print and/or punch operation; field selection; input format; output format; output mode; line spacing; page numbering; sequence numbering of the output; and logging of parameters.

Field-Select Parameter

This parameter specifies field selection, packing, unpacking, and Hexadecimal conversion of fields in the input record. It also indicates when the Key field is to be processed.

This parameter must be used when the field-select operation has been specified in the Utility-Modifier parameter.

Routine Parameters - General *(Cont'd)*

Page Heading Parameter

This parameter provides the text for page headings on the output listing.

PRINT Parameter

This parameter is used when only selected records are to be printed. The parameter specifies the character that must appear in the first position of the input record (data field portion) to be selected for printing. Only records containing the designated select character are printed. (Multiple select characters may be supplied.)

PUNCH Parameter

This parameter is used when only selected records are to be punched. The parameter specifies the character that must appear in the first position of the input record (data field portion) to be selected for printing. Only records containing the designated select character are printed. (Multiple select characters may be supplied.)

PTOPT Parameter

When running under Monitor, this parameter informs the routine that paper tape output is to be produced.

END Parameter

This parameter signifies the end of parameter input.

Routine Parameters - Detailed

Utility-Modifier Parameter

◆ *Formats:*

Print and Punch

$$\Delta U \Delta Tx,Fx,A = \left\{\begin{matrix}(n,m)\\(K=x,D=y)\\(g)\end{matrix}\right\} ,B=(p),Ox,Rn,N=(c,d),Sn,Px,Zx,D=(i,X)$$

Print-only

$$\Delta U \Delta Tx,Fx,A = \left\{\begin{matrix}(n,m)\\(K=x,D=y)\\(g)\end{matrix}\right\} ,B=(p),Ox,Rn,Sn,Px,Zx,D=(i,X)$$

Punch-only

$$\Delta U \Delta Tx,Fx,A = \left\{\begin{matrix}(n,m)\\(K=x,D=y)\\(g)\end{matrix}\right\} ,B=(80,80),Ox,Rn,N=(c,d),Zx,D=(i,X)$$

Utility-Modifier Parameter (Cont'd)

Entry	Meaning
ΔUΔ	Parameter identifier.
Tx	Function: x = B punch and print in List format. = BF punch and print in List format with field-select. = C punch only. = F punch only with field select. = D print only in Display format (preset). = L print only in List format. = LF print only in List format with field select. = MB punch and print Monitor input in List format. = MC punch-only Monitor input. = ML print-only Monitor input in List format.
,Fx	Input record format: x = U undefined (preset). = V variable-length. = F fixed-length.
,A = (n,m) or ,A = (K = x,D = y) or ,A = (g)	Input format: Fixed-length without keys: n = size of record. m = size of block. Fixed-length with keys: x = size of key field. y = size of data field. Variable or undefined: g = maximum size of block (preset to 1,000).
,B = (p) or ,B = (80,80)	Output format: (p) = 132 or 160 (print line size). (80,80) = punch-only function specified.
,Ox	Output mode: x = C Character mode for print or print and punch functions (preset). = 1 EBCDIC mode for punch-only function. = X Hexadecimal mode for printing in Display format.
,Rn	Record bypass (optional): n = first logical record of input file to be printed and/or punched (1-99999) (preset to 1). Note: All records preceding record n will be bypassed.

Utility-Modifier Parameter (Cont'd)

Entry	Meaning
,N = (c,d)	Output sequence numbering: c = first column of sequence field in output card. d = length of field (maximum of 10 characters).
,Sn	Spacing: n = 1 single spacing (preset). = 2 double spacing. = 3 triple spacing. = 4 input record contains write control byte as first character.
,Px	Page numbering: x = Y number pages (preset). = N do not number pages.
,Zx	Log routine parameters: x = L log (preset). = N do not log.
,D = (i,X)	Input random access device. i = 1 disc (preset). = 2 drum. = 3 mass storage.

Notes:

1. All entries are optional and may appear in any order. When an entry is omitted, the preset value is assumed.
2. All entries, except the first, must be preceded by a comma.
3. Multiple parameter cards may be used; it is not required that all entries appear on a single card.
4. Refer to page 1-5 for parameter requirements with paper tape output.

Examples:

```
ΔUΔTBF,FF,A=(K=10,D=40),B=(132),R10,S2,D=(3,X)

ΔUΔTL,FV,A=(200),B=(160)

ΔUΔTF,FF,A=(K=5,D=70),B=(80,80),O1,N=(1,5)
```

Field-Select Parameter

◆ *Format:*

ΔFSΔr,s,t/r,s,t/...../r,s,t

Entry	Meaning
ΔFSΔ	Parameter identifier.
r,s,t	r = starting position (relative to one) of the input field to be selected. s = size of input field. t = starting position (relative to one) of the field in the output record.
(K,r),s,t	If the Key of the input record is to be moved to the output record, replace entry r in above parameter with (K,r): K = identifies Key. r = starting position (relative to one) of the field in the Key to be moved. Note: s and t remain the same.
(U,n,m)	If the input field is to be unpacked when moved to the output record, replace entry s in the above parameter with (U,n,m): U = identifies unpack operation. n = size of input field in bytes. m = size of output field in bytes. Note: r and t remain the same.
(P,n,m)	If the input field is to be packed when moved to the output record, replace entry s in above parameter with (P,n,m): P = identifies pack operation. n = size of input field in bytes. m = size of output field in bytes. Note: r and t remain the same.
(X,n)	If each character of the input field is to be converted from EBCDIC to its hexadecimal equivalent (two characters), replace s in above parameter with (X,n): X = Hexadecimal operation. n = size of input field. (Note: the size of the output field is assumed to be 2 times n.) Note: r and t remain the same.

Field-Select Parameter (Cont'd)

Notes:

1. When a Field Select parameter is used, the Utility-Modifier parameter also must be used.
2. When using field selection, all positions in the output record not filled with input data will be space-filled.
3. More than one field-select card may be used; it is unnecessary to fill a card completely before starting another. All information for a particular field, however, must appear on the same card.

Examples:

ΔFSΔ(K,1),10,71/1,70,1

ΔFSΔ1,(X,5),150

ΔFSΔ(K,1),(X,10),1/1,40,20

Page Heading Parameter

◆ *Format:*

ΔHnΔtext

Entry	Meaning
ΔH1Δtext	Text for positions 1-76 of print header line.
ΔH2Δtext	Text for positions 77-142 of print header lines.
ΔH3Δtext	Text for positions 143-160 of print header lines.

Note:

Any or all parameters may be supplied.

PRINT Parameter

◆ *Format:*

ΔPRINTx.......x//Δ

Entry	Meaning
ΔPRINT	Parameter identifier.
x.....x	Select character(s): x = any character that must appear as the first character in an input record to be printed. Note: Column 77 is the last column in which a select character may be specified.
//Δ	Parameter termination sequence.

Examples:

To print only records containing an A in the first position of the Data field.

ΔPRINTA//Δ

To print only records containing an A,5,7, or Y in the first position of the Data field.

ΔPRINTA57Y//Δ

Device Assignments *(Cont'd)*

Under Monitor Control

SDN	Device Type	Remarks
SYSOPT	Card punch or magnetic tape.	Output device.
SYSLST	Printer or magnetic tape.	Output device.
SYSIPT	Card reader or magnetic tape.	Parameter input device.
PROPT	Paper tape punch.*	Output device.

*Refer to page 1-4 for paper tape output considerations.

Note:

As the random access device is defined in the VDC cards at run time, it is not necessary to make this assignment at load time.

PUNCH Parameter

◆ *Format:*

ΔPUNCHx.....x//Δ

Entry	Meaning
ΔPUNCH	Parameter identifier.
x.....x	Select character(s): x = any character that must appear as the first character in an input record to be punched. <u>Note:</u> Column 77 is the last column in which a select character may be specified.
//Δ	Parameter termination sequence.

Examples:

To punch only records containing an A in the first position of the Data field:

ΔPUNCHA//Δ

To punch only records containing an A,5,7, or Y in the first position of the Data field:

ΔPUNCHA57Y//Δ

PTOPT Parameter

◆ *Format:*

ΔPTOPT

This parameter is used under Monitor to inform the routine that paper tape output is to be produced. The routine will then request the assignment of PROPT.

END Parameter

◆ *Format:*

ΔEND

This parameter signifies the end of parameter input and is included whenever a Utility-Modifier parameter has been used.

Device Assignments

◆ *Under Executive Control*

SDN	Device Type	Remarks
PROPT	Card punch or paper tape punch.*	Output device.
PRLST	Printer.	Output device.
PRPRM	Card reader, magnetic tape, or paper tape reader.	Parameter input device.

Considerations for Use

◆

1. A VDC parameter must be supplied at runtime. See TOS Operators' Guide, Section 2, for format.

   The filename entry for this parameter is ITUTP4F.

   The size of the extent matrix generated in this routine allows for 1 volume with 7 extents (104 bytes). If a larger extent matrix is desired, its size can be specified in the VDC parameter.

2. If the serial number of the volume to be processed does not appear in the On-Line Catalog, the EΔOLC console routine must be run first.

3. All random access input volumes must be initialized by the RAINIT routine and allocated by the RAALLR routine.

4. This routine does not transcribe from the input file the first track of the file area. (This track is reserved in the system for user header and trailer label information.)

Parameter Examples

◆

1. To print a file of disc records, with Keys, in List format and Character mode:

```
//ΔVDCΔITUTP4F,,file identification, serial number

//ΔENDΔ

ΔUΔTLF,FF,A= (K=10,D=90),B=(132)

ΔFSΔ(K,1),10,1/1,90,11

ΔEND
```

2. To punch and print selected fixed-length blocked records from drum using page headings and field selection; and numbering the punched card output:

```
//ΔVDCΔITUTP4F,,file identification, serial number

//ΔEND

ΔUΔTBF,FF,A=(100,300),B=(160),N=(1,5)

ΔFSΔ99,(X,2),1/1,95,5

ΔPRINT24FH//Δ

ΔPUNCH13EG//Δ

ΔH2Δ.......

ΔEND
```

Parameter Examples
(Cont'd)

3. To print a Monitor SYSLST file on disc.

```
//ΔVDCΔITUP4F,,SYSOPT,serial number
//ΔEND
ΔUΔTMLΔFV,S4,PN
ΔEND
```

4. DIAGNOSTICS

EXECUTIVE DUMP PRINT (DUMPRT)

General Description

◆ The Executive Dump Print routine produces an edited listing of memory dumps recorded on magnetic tape by the Executive Dump console routine.

Any or all memory dumps on the input tape may be selected for editing.

Preset Functions

Program dump information is edited and printed in the following order:

1. A dump header line which contains the name of the dumped program, the date and time that the dump was taken, and the contents of the P1 program counter.
2. The 16 general registers used by the P1 processing state, edited in full-word hexadecimal format.
3. The floating-point registers, edited in hexadecimal mantissa and exponent format.
4. The program table entry of the dumped program.
5. The device list entries (if any) of the dumped program.
6. The Executive storage area.
7. The program area.
8. The run time parameters (if any) provided for the program.
9. The Executive tables.

Optional Functions

None.

Input

◆ Input to this routine consists of a magnetic tape that contains memory dumps taken by the Executive Dump console routine. Tape print directions are entered from the console typewriter.

A tape mark precedes each program dump on the input tape; a double tape mark appears after the last program dump.

Output

◆ Output consists of an edited program dump(s) displayed on the on-line printer.

Equipment Configuration

Required

◆ Processor (65K)

Console typewriter

Magnetic tape device

Optional

◆ Additional magnetic tape devices can be used as input to this routine.

Routine Parameters

◆ None.

Device Assignments

SDN	Device Type	Remarks
DUMPMT	Magnetic tape	Input tape containing Executive memory dumps.

SELF-LOADING MEMORY PRINT

General Description

◆ The Self-Loading Memory Print routine is an emergency testing aid that provides a listing of all, or part, of main memory and the contents of scratch-pad memory. This routine is used when a program has terminated in an abnormal or unexpected manner and the standard memory print (Dump and Terminate console request) cannot be used.

This routine contains its own bootstrap, loader, and device control. As a result, it is not dependent on any other programming system and is loaded without the Executive or Monitor.

Preset Functions

Scratch-pad memory is displayed in hexadecimal with all registers labeled according to their function. The contents of main memory are printed in 16-byte groups of hexadecimal characters with the corresponding EBCDIC graphic appearing above the associated hexadecimal code. Three 16-byte groups appear on each print line.

Each group within a line is preceded by the hexadecimal address of the first byte in that group. To conserve printing, the routine displays only the first of duplicate print lines on the output listing; this condition is indicated by an asterisk immediately following the first address on the line that is printed.

Parameters specifying the area of memory to be printed are furnished by means of the console typewriter. (If desired, this routine can be used to print only the contents of scratch-pad memory. This is effected by assigning the same hexadecimal address to the left-hand end and right-hand end of the area to be printed.)

Optional Functions

None.

Input

◆ 1. The contents of scratch-pad memory.

2. The contents of the designated main memory area.

3. Parameters specifying the area of memory to be printed and the output device (printer or magnetic tape).

Output

◆ The output of this routine is a printer listing showing the contents of scratch-pad memory and the areas of main memory specified by the parameters.

If desired, the output can be written to magnetic tape instead of the printer. (If the tape is seven-level, output is written at 800 bpi, odd parity, with pack/unpack.) The output tape is left positioned to provide for stacking of dumps. At end-of-job the tape is rewound.

Equipment Configuration

Required

◆ Processor (65K)

Console typewriter

Card reader, or Videoscan document reader with punched card read feature

Printer

Optional

◆ A magnetic tape device can be substituted for the printer.

Routine Parameters - General

◆ Parameter data is supplied to this routine by means of the console typewriter. Initially, the operator types in the following information:

1. the limits of the memory area to be printed.
2. the channel and unit number of the output device.
3. the type of output device (either printer or magnetic tape).

After the contents of the area specified have been printed, the operator can obtain additional prints by simply typing in the left-hand end and right-hand end addresses of another memory area.

When no further dump operations are required, the operator types END to terminate the routine.

Routine Parameters - Detailed

◆ The Self-Loading Memory Print routine requires that all parameter information be submitted through the console typewriter. After the second program card has been read, the message FLOAT is displayed. The operator then keys in the float address for the routine.

The print routine may be floated in memory as low as $00080_{(16)}$, or loaded at the end of memory, based on the following core sizes:

65K - 0F610

131K - 1F610

262K - 3F610

524K - 7F610

Routine Parameters - Detailed *(Cont'd)*

After the routine has been loaded, the memory area to be displayed is inserted from the console typewriter in the following format:

lllllrrrrr∧cuu∧d

where: lllll = hexadecimal address of the left-hand end of the memory area to be printed.

rrrrr = hexadecimal address of the right-hand end of the memory area to be printed.

cuu = channel and unit number of the output device.

d = output device type. If nine-level tape, enter T; if seven-level, enter S; if printer, leave blank.

After the dump of the area defined above has been completed, the routine provides the operator with the facility to obtain additional dumps. To do this the operator types in the limits of the next memory area in the following format:

lllllrrrrr

When no further dump operations are desired, the operator terminates the routine by typing END.

Device Assignments

◆ Not applicable.

Related Programming Systems

◆ If the output of this routine is transcribed to magnetic tape, the preedit option of the Tape Edit routine is used to print the output tape.

SNAPSHOT

General Description

◆ The Snapshot routine is a program testing aid that provides an edited listing of the general registers of the P1 state, the floating-point registers, and selected areas of main memory.

To obtain snapshots during program execution, the programmer must provide SNAPS macro call lines in his assembly source deck at those points where snapshot dumps are required.

If desired, any or all snapshot points can be inhibited at run time.

Preset Functions

Not applicable.

Optional Functions

Memory areas can be edited into the following formats:

Format H - Memory is printed in hexadecimal. Each print line contains 40 bytes separated into groups of 1, 2, or 4 bytes. Each line is preceded by the memory location of its first byte.

Format G - Memory is printed in EBCDIC graphics. Each line contains 56 bytes separated into groups of 1, 2, or 4 bytes. Each line is preceded by the memory location of its first byte.

Format C - Memory is printed in both EBCDIC graphics and hexadecimal. The first line contains 40 bytes of graphics separated into groups of 1, 2, or 4 bytes. The following line contains the same 40 bytes printed in hexadecimal. Each hexadecimal line is preceded by the memory location of its first byte.

Format M - Memory is printed in half-word hexadecimal, 40 bytes to the line, with the corresponding instruction mnemonics printed immediately below. Each hexadecimal line is preceded by the memory location of its first byte.

Format S - Each word of memory is converted to floating point format. The 24-bit mantissa and the 7-bit exponent are displayed in hexadecimal.

Six words appear on a line and each line is preceded by the memory location of its first byte.

Format F - Each double-word of memory is converted to floating-point format. The 56-bit mantissa and the 7-bit exponent are displayed in hexadecimal.

Four double-words appear on a line, with each line preceded by the memory location of its first byte.

An asterisk (*) between the location counter and the displayed contents of memory is used to indicate that one or more lines were not printed because their contents were identical to the line on which the asterisk appears.

General Description *(Cont'd)*

Input

◆ 1. The contents of the P1 state general registers and the floating-point registers.

2. The contents of the specified memory area.

3. SNAPS parameter call lines describing the limits of memory to be printed, the print format, and the grouping factor.

Output

◆ The output of this routine consists of an edited listing of the general registers of the P1 state, the floating-point registers, and the contents of the designated memory area displayed on the printer, or optionally written to tape. If the output is transcribed to magnetic tape, the Tape Edit routine (preedit option) is used to print the output tape.

Equipment Configuration

Required

◆ Processor (65K)

Console typewriter

Card reader, or Videoscan document reader with card read feature
Printer

Optional

◆ A magnetic tape device can be substituted for the printer.

Routine Parameters - General

◆ Parameter information is supplied to this routine by SNAPS macro call lines inserted into the assembly source program. These parameters specify the output format, the memory area to be edited, and the grouping factor. (This information can also be furnished to the Snapshot routine through the console typewriter if a 2 indicator is coded into the SNAPS macro call line.)

During program testing, snap points can be inhibited by a console typewriter request or by a card parameter. However, if the program is running under Monitor control, snap points cannot be inhibited.

A multiple parameter listing feature is also provided whereby multiple areas and formats may be displayed for any snap point.

Routine Parameters - Detailed

◆ Three variations of the SNAPS macro call line are provided: independent, delayed, and multiple listing. For the independent format, the programmer provides the memory limits of the area to be printed and the print format to be used. In the delayed format, the memory area and print format information are supplied at program execution time. The multiple listing format is used in conjunction with a parameter list, providing the ability to display more than one memory area at a single snapshot point.

SNAPS Macro (Independent) Call Line

◆ *Format:*

nnnnnnnnΔSNAPSΔn,1,lhe,rhe,f,n

Card Columns	Content	Meaning
1-8		May contain symbolic tag or be left blank.
9		Not used; leave blank.
10-14	SNAPS	Macro identifier.
15		Not used; leave blank.
16	n	Hexadecimal character (0-F) which the programmer assigns to identify this snapshot point. (At execution time this identifier can be used with the INHIBIT parameter to suppress this snapshot.)
17-18	,1	Identifies the independent format.
19....	,lhe	LHE address of the memory area to be printed, expressed symbolically. Relative addressing is permissible, provided the tag and appendage do not exceed eight characters.
	,rhe	RHE address of the memory area to be printed, expressed symbolically. Relative addressing is permissible, provided the tag and appendage do not exceed eight characters.
	,f	Specifies the format in which memory is to be printed; where: f=H hexadecimal. =G EBCDIC graphics. =C hexadecimal with graphic equivalents. =M half-word hexadecimal with corresponding instruction mnemonics. =S single-word floating-point hexadecimal. =F double-word floating-point hexadecimal.
	,n	Specifies the format grouping factor; where: n=1 one byte per print group. =2 two bytes per print group. =4 four bytes per print group.

SNAPS Macro (Independent) Call Line (Cont'd)

Note:

If the LHE and RHE addresses are the same or the LHE address is greater than the RHE address, only the general registers and floating-point registers will be displayed.

Example:

Figure 4-1 shows the SNAPS parameter card used to print an 800-character area defined as WK in the source program. Printing is to be in EBCDIC format, one byte per print group; the identifier assigned to this operation is A.

Card Columns	1-9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Content		S	N	A	P	S		A	,	1	,	W	K	,	W	K	+	7	9	9	,	G	,	1

Figure 4-1.

SNAPS Macro (Delayed) Call Line

◆ *Format:*

nnnnnnnnΔSNAPSΔn,2

Card Columns	Content	Meaning
1-8		May contain symbolic tag or be left blank.
9		Not used; leave blank.
10-14	SNAPS	Macro identifier.
16	n	Hexadecimal character 0-F which the programmer assigns to identify this snapshot point. (At execution time this identifier can be used with the INHIBIT parameter to suppress this snapshot.)
17-18	,2	Identifies the delayed format.

SNAPS Macro (Delayed) Call Line (Cont'd)

Note:

At program execution time the console operator provides memory limits and print format by typing in the following information:

llllllrrrrrrfn

where:

llllll = LHE address (hexadecimal) of area to be displayed.

rrrrrr = RHE address (hexadecimal) of area to be displayed.

f = print format (H,G,C,M,S, or F)

n = grouping factor (1,2 or 4).

SNAPS Macro (Multiple Listing) Call Line

◆ *Format:*

nnnnnnnnΔSNAPSΔn,c,a

Card Columns	Content	Meaning
1-8		May contain symbolic tag or be left blank.
9		Not used; leave blank.
10-14	SNAPS	Macro identifier.
15		Not used; leave blank.
16	n	Hexadecimal character (0-F) which the programmer assigns to identify this snapshot point. (At execution time this identifier can be used with the INHIBIT parameter to suppress this snapshot.)
17-18	,c	Number of entries contained in parameter list; where: c = 8 one entry c = C five entries = 9 two entries = D six entries = A three entries = E seven entries = B four entries = F eight entries See note 1 below.
19....	,a	Symbolic address of parameter list, where a is the tag used for the first entry in the list. See note 1 below.

SNAPS Macro (Multiple Listing) Call Line (Cont'd)

Notes:

1. The snapshot parameter list may contain up to eight entries, each of which designate a particular memory area to be displayed. Each entry is eight bytes, defined as follows:

```
Tag    Op   Operand
----   --   -------
name   DC   AL4(lhe)  )
       DC   X'fn'     >   first area to be displayed.
       DC   AL3(rhe)  )
       DC   AL4(lhe)  )
       DC   X'fn'     >   second area to be displayed.
       DC   AL3(rhe)  )
               .
               .
               .
```

The lhe and rhe addresses are symbolic tags, with or without relative appendages; 'fn' is the print format, where:

f = 8 hexadecimal
    7 EBCDIC graphics
    3 hexadecimal and graphics
    4 half-word hexadecimal with instruction mnemonics
    2 single-word floating point hexadecimal
    6 double-word floating-point hexadecimal

n = 1 one byte per print group
  = 2 two bytes per print group
  = 4 four bytes per print group

2. The same parameter list may be used by more than one snapshot call line. For example:

```
TESTA   SNAPS  0,9,CHECK
TESTB   SNAPS  A,9,CHECK
CHECK   DC     AL4(WORK)
        DC     X'32'
        DC     AL3(WORK+22)
        DC     AL4(TOTALS)
        DC     X'84'
        DC     AL3(TOTALS+417)
```

INHIBIT Parameter

◆ This parameter is optional and can be supplied at program execution time to inhibit snapshot points assembled into the source program.

Only one inhibit parameter may be given. The opportunity to do so occurs when the first SNAP point is encountered at program execution time.

Format:

ΔINHIBITΔs,s , ... , s

Entry	Meaning
ΔINHIBITΔ	Parameter identifier.
s	Snapshot point (0-F) to be inhibited. From 1 to 16 points may be specified, with each identifier separated from the preceding one by a comma.

Notes:

1. This parameter cannot be used when the program is executed under Monitor control.
2. This parameter may be entered by a console type-in or from a parameter card read from the card reader. If entered by the typewriter, the first nine characters (ΔINHIBITΔ) must not be typed.

Examples:

ΔINHIBITΔ2

ΔINHIBITΔ3,A,C

ΔINHIBITΔ0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Device Assignments

◆ Under Executive Control

SDN	Device Type	Remarks
SNPRDR	Card reader.	For INHIBIT parameters.
SNAPOP	Printer or magnetic tape.	Output device. If magnetic tape, preedit option of Tape Edit routine must be used to print the output tape.

Under Monitor Control

SDN	Device Type	Remarks
SYSLST	Printer or magnetic tape.	Output device.

SELF-LOADING TAPE EDIT

General Description

◆ The Self-Loading Tape Edit routine is an emergency testing aid that displays the contents of a magnetic tape reel in hexadecimal format, EBCDIC graphic format, or both. The format in which the data is to be displayed is determined from an input parameter message entered by way of the console typewriter.

This routine contains its own bootstrap, loader, and device control. As a result, it is not dependent on any other programming system and is loaded without the Executive or Monitor.

Preset Functions

This routine provides an edit of the data on a magnetic tape reel until a double tape mark is recognized.

Optional Functions

The input tape may be edited (in one of three print formats) from its current position or from BOT, based on the edit option specified in the input parameter. Multiple edits can be taken within the same run.

Input

◆ 1. A reel of magnetic tape that terminates with a double tape mark.

2. Parameters entered from the console typewriter that specify the type of output device and the edit option desired.

Output

◆ Output consists of a printer listing showing the contents of the input tape edited according to the option specified. Or, this information may be transcribed to another tape for later printing.

Equipment Configuration

Required

◆ Processor (65K)

Console typewriter

Printer

Card reader, or Videoscan document reader with card read feature

Magnetic tape device

Optional

◆ A magnetic tape can be substituted for the printer as the output device.

Routine Parameters - General

◆ Two parameters are required: an output parameter, which provides the type of output device to be used; and an input parameter, which specifies the editing option to be exercised.

Both parameters are entered from the console typewriter.

Routine Parameters - Detailed

◆ After the routine has been loaded, the following two parameters are entered from the console typewriter.

Output Parameter

◆ *Format:*

cuuΔd

where: cuu = channel and unit number of the output device.

d = device type. If tape, enter T; if printer, leave blank.

Input Parameter

◆ *Format:*

cuuΔnΔss

where: cuu = channel and unit number of the input device.

n = edit option to be exercised (decimal 1-6).

ss = seven-level magnetic tape control information. (If nine-level input, these positions are not used.)

Option Number	Meaning
1	Rewind tape to BOT; print in graphic format to double tape mark.
2	Rewind tape to BOT; print in hexadecimal format to double tape mark.
3	Rewind tape to BOT; print in hexadecimal format with graphic equivalents to double tape mark.
4	Begin printing from current position of tape; print in graphic format to double tape mark.
5	Begin printing from current position of tape; print in hexadecimal format to double tape mark.
6	Begin printing from current position of tape; print in hexadecimal format and graphic equivalents to double tape mark.

When no further edit options are desired, the operator terminates the routine by typing END.

Device Assignments

◆ Not applicable.

Related Programming Systems

◆ If the output of this routine is transcribed to magnetic tape, the preedit option of the Tape Edit (TPEDIT) routine is used to print the tape at a later time.

TAPE EDIT (TPEDIT)

General Description

◆ The Tape Edit routine displays all, or selected portions, of a magnetic tape on the on-line printer. The contents of the tape may be displayed in Character mode (EBCDIC graphics), Hexadecimal mode, or both. At end-of-job the programmer can terminate the routine or enter additional parameters. In this way, different portions of the same tape (or another tape) can be edited.

Preset Functions

This routine is preset to perform the following edit functions for an input tape mounted on logical device EDT001:

Rewind input tape to BOT at the beginning and end of processing.

Edit all information recorded between the beginning of tape and a double tape mark.

Edit tape blocks of undefined length. (Formula for calculating maximum input area: available memory less size of Tape Edit routine.)

Edit and print data in both Character and Hexadecimal modes on a 132-character line printer.

Optional Functions

A number of optional functions are available and can be specified by the following parameter cards entered from the console typewriter or the card reader:

EDIT Parameter

Editing of the input tape based on a given number of blocks or files (tape marks).

Positioning of the input tape before and after printing.

Selecting of print mode and printer line size.

Printing of a preedited tape.

Processing of fixed-length, blocked records.

Page Heading Parameter(s)

Replacing or adding to the RCA-supplied title line.

General Description *(Cont'd)*

Input

◆ Input to the Tape Edit routine consists of a magnetic tape (labeled or unlabeled) that terminates with a double tape mark. For labeled tapes, all label blocks are treated as data blocks.

Input records may be fixed or undefined in length. If fixed, they may also be blocked.

When a preedited tape is printed, each input block generates one print line, and the first character of each input record must contain a write control character. The size of the input block cannot exceed 133 or 161 characters, depending on the printer being used.

When other than preset functions are desired, these functions must be specified by parameters entered from the console typewriter or card reader.

Output

◆ Output consists of the contents of a magnetic tape file printed and edited as specified by the parameters.

Except for preedited tapes, any block larger than the allowable maximum is truncated and so indicated by an * symbol on the listing. For preedited tapes, this condition is indicated by a typeout which provides the block number of the block that was truncated.

Equipment Configuration

Required

◆ Processor (65K)

Console typewriter

Card reader, or Videoscan document reader with card read feature

Magnetic tape device

Printer

Optional

◆ A magnetic tape device can be substituted for the printer.

Routine Parameters - General

◆ This routine requires no parameters for its standard preset options. For other options, the following parameters can be entered from the console typewriter or the card reader.

EDIT Parameter

Used to specify the number of blocks or files to be printed, to position the tape in a forward or reverse direction, to select print mode and printer line size, and to specify input record size.

Routine Parameters - General *(Cont'd)*

Page-heading Parameter(s)

Used to provide the text for the title line that appears at the top of the first page only. A header can be printed in addition to, or instead of, the standard RCA-supplied header line.

End Parameter

Used to signify the end of parameter information.

Routine Parameters - Detailed

EDIT Parameter

◆ *Format:*

ΔEDITΔSnnn,Oo,Cnnnnn,Ff,Pp,Ln,Rnnn

Entry	Meaning
ΔEDITΔ	Parameter identifier.
Snnn	Logical device number of input tape: nnn = 000-999 If omitted, EDT001 is assumed.
,Oo	Edit option to be executed (see page 4-18): o = 0-9 If omitted, option 0 is assumed.
,Cnnnnn	Number of input blocks or tape marks to be edited (Note: edit options refer to this field as NNNNN): nnnnn = 00001-99999
,Ff	Print mode: f = C Character and Hexadecimal mode. = G Character mode. = H Hexadecimal mode. = P Input tape is preedited. (In this case, the first byte of each block must contain a write control byte). If omitted, mode C is assumed.
,Pp	Printer line size: p = 132 = 160 If omitted, 132 is assumed.

EDIT Parameter (Cont'd)

Entry	Meaning
,Ln	Line spacing: n = 0-3 Number of lines to be spaced after each record. (Ignored if input is a preedited tape.) If 0, all printing is performed on consecutive lines. If omitted, double-line spacing occurs after each input record has been displayed. (This is also the preset option of this routine.)
,Rnnn	Input record format: nnn = 000 Undefined length blocks. = 012-999 Number of characters in a record. Used only when input is fixed-length and blocked. (Note: each record in the block will begin a new line on the output listing.) If omitted, undefined blocks are assumed.

Notes:

1. Commas must be used to separate all parameters after the first. Any or all parameter fields are optional.
2. Whenever an entry is omitted, the preset option (or the option used in a preceding edit parameter) is used.

Examples:

```
ΔEDITΔS003,O1,C00335,FG,P160,L2,R080

ΔEDITΔO1,P160,L1

ΔEDITΔO3,C00003,L3
```

The following options may be selected for the O entry in the edit parameter:

OPTION 0: Print Entire Tape (Preset)

The input tape is rewound to beginning of tape, and printing proceeds until a double tape mark is sensed. The input tape is then rewound. This option is preset in the Tape Edit routine to print the contents of the tape mounted on logical EDT001.

OPTION 1: Print NNNNN Blocks Starting at Beginning of Tape

The input tape is rewound, and NNNNN blocks are printed. The input tape is then rewound.

EDIT Parameter (Cont'd)

OPTION 2: Print Through NNNNN Tape Marks Starting at Beginning of Tape

The input tape is rewound, and printing takes place through the number of tape marks specified by NNNNN. When NNNNN tape marks have been printed, the edit terminates and the tape is rewound.

OPTION 3: Backspace NNNNN Blocks and Print to Current Position of Tape

The input tape is backspaced NNNNN blocks, and NNNNN blocks are printed. After printing, the tape is positioned as it appeared before the edit.

OPTION 4: Backspace NNNNN Tape Marks and Print Through NNNNN Tape Marks

The input tape is backspaced NNNNN tape marks and printed through the same number of tape marks.

OPTION 5. Print NNNNN Blocks Forward and Reposition Tape to Current Position

Starting from its current position, the input tape is read forward and NNNNN blocks are printed. After printing, the tape is backspaced NNNNN blocks to return it to its original position.

OPTION 6: Print NNNNN Tape Marks Forward and Backspace NNNNN Tape Marks

Starting from its current position, the input tape is read forward and printing takes place through NNNNN tape marks. After printing, the tape is backspaced the same number of tape marks.

OPTION 7: Unwind NNNNN Blocks

NNNNN blocks are skipped. No printing takes place.

OPTION 8: Unwind NNNNN Tape Marks

NNNNN tape marks are skipped. No printing takes place.

OPTION 9: Print from Current Position to Double Tape Mark and Reposition

Starting from its current position, the input tape is read forward and printing takes place until a double tape mark is sensed. The tape is then backspaced to its original position.

Page Heading Parameters

◆ *Format:*

ΔHDRnhtext

Entry	Meaning
ΔHDR	Parameter identifier.
n	Header number: n = 1 for first header. = 2 for second header.
h	Header type: h = R replace RCA header line. = A add to RCA header line (to be printed on the line immediately below.)
text	User-supplied header information. (Maximum number of characters per card is 66.)

Note:

Either or both cards may appear. The RCA-supplied header consists of one line and contains the program name and version number. The user-supplied header to replace the RCA line will also be one line - card 1 at the leftmost position on the page, followed by card 2 (if present). If a header is added, it will appear on the line immediately below the RCA title line. The HDR parameter(s) must appear before the associated EDIT parameter.

Examples:

To add a header to the RCA-supplied header:

ΔHDR1A....text....

To replace the RCA-supplied header:

ΔHDR1R....text....

ΔHDR2R....text....

END Parameter

◆ This parameter signifies the end of parameter input.

Format:

ΔEND

Inserting Parameters From Typewriter

◆ The EDIT and END parameters can be entered from the console typewriter; HDR parameters cannot. When entering parameters from the console, the leading space character is not typed.

Device Assignments

◆ Under Executive Control

SDN	Device Type	Remarks
EDTRDR	Card reader	Parameter input.
EDTnnn	Magnetic tape	Input tape, where nnn = 000-999.
EDTLST	Printer or magnetic tape	Output listings.

Under Monitor Control

SDN	Device Type	Remarks
SYSIPT	Card reader	Parameter input.
EDTnnn	Magnetic tape	Input tape, where nnn = 000-999.
SYSLST	Printer or magnetic tape	Output listings.

TAPE COMPARE (TPCOMP)

General Description

◆ The Tape Compare routine is a diagnostic aid used to compare information recorded on one magnetic tape with that of a second magnetic tape. It provides a printed listing of all portions of these tapes that are unequal.

Comparison of data is done on a decade basis, which allows for the printing of 5 groups of variant data on a 132-character line printer or 6 groups on a 160-character line printer.

Preset Functions

This routine has a preset option which provides the following standard functions for tapes mounted on logical devices COM001 and COM002.

Rewinding both tapes to BOT at the start and end of the comparison process.

Displaying of variant data in both hexadecimal and EBCDIC graphics on a 132-character line printer.

Terminating the comparison process when two consecutive tape marks are sensed. (If a double tape mark for one tape is sensed, and the other tape is not also positioned at its double tape mark, the data remaining on the second tape is displayed until the tape marks are encountered.)

Optional Functions

By the use of parameters (entered through the console typewriter or card reader) the following options may be elected:

1. Positioning of input tapes before, after, or during the comparison process.

   Tapes can be positioned in a forward or reverse direction based on a count of tape marks or blocks, or in a forward direction until a specified file name is found.
2. Terminating the comparison based on a given number of tape marks, blocks, or upon reaching a specified file name.
3. Designating a 160-character line printer; specifying print formats of hexadecimal, EBCDIC graphic, or a combination of both.
4. Processing of variable records or blocked fixed-length records.
5. Deallocating devices to permit assigning and processing of different tapes.

Input

◆ Input to the Tape Compare routine consists of two single-volume magnetic tapes that are recorded in the same format and packing density.

If other than the preset options are desired, appropriate parameters must be entered through the console typewriter or the card reader.

Output

◆ Output is a printed listing of any discrepancies that exist between the tapes being compared.

Equipment Configuration

Required

◆ Processor (65K).

Console typewriter.

Card reader, or Videoscan document reader with card read feature.

Magnetic tape devices (two required).

Printer.

Optional

◆ A magnetic tape device can be substituted for the printer, in which case the preedit option of the Tape Edit routine must be used to print the output tape.

Large blocks require more memory allocation. Ⓐ

Routine Parameters - General

◆ This routine requires no parameters for its standard preset option. For other options, the following parameters can be entered from the console typewriter or the card reader.

Postion Parameter(s)

Used to position input tapes.

Compare Parameter(s)

Used to designate that only certain areas of the input tapes are to be compared. This parameter can also be used to specify print line size and to select print formats.

Record Parameter

Used to signify that input records are variable in length or that blocked fixed-length records are to be processed on a logical record level.

Restart Parameter

Used to deallocate devices to permit processing of different tapes.

END Parameter

Used to signify the end of parameter information.

Routine Parameters - Detailed

Position Parameter

◆ *Format:*

ΔPOSnΔpppnnnnn,POS2Δpppnnnnn

Entry	Meaning
ΔPOS	Parameter identifier.
n	If only the tape assigned to COM001 is to be positioned, n = 1. If only COM002 is to be positioned, n = 2.

Position Parameter (Cont'd)

Entry	Meaning
Δppp	ppp = FTM read forward nnnnn tape marks. = RTM read reverse nnnnn tape marks. = FBK read forward nnnnn blocks. = RBK read reverse nnnnn blocks. = FID read forward to specified file. (The tape is positioned following the file label.) = BOT rewind to BOT. Note: nnnnn values appear in the next entry.
nnnnn or file-name	nnnnn = decimal number of tape marks or blocks (zero-filled). file-name = 17-character file ID as found in header label.
, POS2	Parameter identifier. This entry (and all following entries) is used only when both tapes require positioning.
Δppp	See format described above.
nnnnn or file-name	See format described above.

Examples:

ΔPOS1ΔFTM00002

ΔPOS2ΔFBK00333

ΔPOS2ΔFIDMASTERΔFILE

ΔPOS1ΔFBK00017,POS2ΔFBK00017

Compare Parameter

◆ *Format:*

ΔCOMΔstcnnnnn

Entry	Meaning
ΔCOMΔ	Parameter identifier.
s	s = 0 132-character print line. = 1 160-character print line.
t	Print format: t = H hexadecimal. = G EBCDIC graphics. = C hexadecimal and graphic format.
c	c = T terminate after reading nnnnn tape marks on both tapes. = B terminate after reading nnnnn blocks on both tapes. = F terminate after file x is found on both tapes. Note: nnnnn and file x values appear in the next entry.
nnnnn or file-name	nnnnn = decimal number of tape marks or blocks (zero-filled). file-name = 17-character file ID as found in header label.

Compare Parameter (Cont'd)

Examples:

ΔCOMΔ0CT00006

ΔCOMΔ1HB00672

ΔCOMΔ1GFSALESΔTRANSACTION

Note:

After the operation specified by a Compare parameter has been completed, the routine will accept, if provided, a Position parameter. At this point, the comparison process can be terminated by using an END card, or the programmer can call for another comparison of the input tapes by supplying additional Compare and Position parameters. For example:

COM	POS	COM	POS	COM	COM
END	COM	COM	COM	POS	COM
	POS	COM	COM	COM	POS
	END	END	END	POS	END
				END	

If two consecutive POS parameters are used, the preset COM function is implied and executed.

Record Parameter

◆ *Format:*

ΔRCDΔtΔnnnnn

Entry	Meaning
ΔRCDΔ	Parameter identifier.
t	t = F fixed-length records. = V variable-length records.
Δnnnnn	nnnnn = fixed-length record size (zero-filled). (This entry does not apply to variable-length records.)

Note:

This parameter is required when comparing variable-length records or blocked fixed-length records.

RESTART Parameter

◆ *Format:*

ΔRESTART

This parameter is used only when it is desired to deallocate the devices assigned to COM001 and COM002 in order to compare a new set of input tapes. The appropriate COM and POS parameters that apply to the new set of tapes must follow the RESTART parameter. For example:

RESTART Parameter (Cont'd)

```
ΔPOS     }
ΔCOM     }   first set of tapes to be compared.
ΔRESTART
ΔPOS     }
ΔCOM     }   second set of tapes to be compared.
ΔEND     }
```

END Parameter

◆ *Format:*

ΔEND

This parameter denotes the end of parameter information and must be used when POS, COM, or RCD parameters have been supplied. It is not required for preset options.

Parameter Examples

◆

1. The programmer wants to use the preset functions. No parameters are necessary. The preset functions are obtained by typing nΔ and depressing EOT in reply to the console message n 28PDA PARAM DEVICE.
2. The programmer wishes to compare the data contained between the fourth and fifth tape marks on one tape with the data contained between the first and second tape marks of the other tape. All discrepancies are to be printed in EBCDIC graphics on a 160-character line printer.

```
ΔPOS1ΔFTM00004,POS2ΔFTM0001

ΔCOMΔ1GT00001

ΔEND
```

Device Assignments

◆ Under Executive Control

SDN	Device Type	Remarks
COMRDR	Card reader.	Parameter input.
COMLST	Printer or magnetic tape.	Output listings.
COM001	Magnetic tape.	First input tape.
COM002	Magnetic tape.	Second input tape.

Under Monitor Control

SDN	Device Type	Remarks
SYSIPT	Card reader.	Parameter input.
SYSLST	Printer or magnetic tape.	Output listings.
COM001	Magnetic tape.	First input tape.
COM002	Magnetic tape.	Second input tape.

TEST DATA GENERATOR (DIAGDG)

General Description

◆ The Test Data Generator routine automatically prepares files of program test data generated onto punched cards, magnetic tapes, random access volumes, or paper tape. This routine can be used to produce single or multivolume files, or multifile volumes.

For output tapes, standard Spectra 70 labels can be automatically generated; the output tapes may be unlabeled; or the programmer may supply his own label set. For random access output, standard Spectra 70 user header and trailer labels may be supplied by the programmer.

Records generated for the output files can vary in length, contain up to 12 data fields, and can be blocked or unblocked. Records can also contain programmer-supplied data.

Preset Functions

This routine is not preset to perform any functions for a random access file. It is, however, preset to perform the following functions for a magnetic tape file mounted on logical TDG001:

Rewind the output tape to BOT.

Generate 1,000 unblocked, 80-byte records to the output tape. (The first field of each record is a 10-byte, zoned-decimal field, with each succeeding record incremented by 10. All other positions of the record contain the fill character X.)

Rewind the output tape to BOT and deallocate logical device TDG001.

Note:

If the output tape contains VOL and HDR labels, a purge-date check is made to determine if the tape is releasable. If not, an error halt occurs.

The preset function of this routine creates an output tape in the following format:

TM data TM TM

Optional Functions

The following options may be selected by programmer-supplied parameter cards:

1. Designating a random access volume as the output file.

General Description
(Cont'd)

2. Designating up to 12 data fields per file, which may vary in size and format.

3. Designating the length (fixed or variable) of the test records to be generated; blocking of test records; specifying the number of blocks to be generated per file or per volume.

4. Specifying for magnetic tape a standard-labeled file, an unlabeled file, or providing programmer-prepared label sets for the output file.

5. Specifying random access test records with Keys.

6. Providing programmer-prepared user header and trailer label sets for the random access output file.

7. Limiting the number of test records generated on random access by extent, by right-hand end address, or by number of records desired.

8. Producing multifile volumes or multivolume files.

9. Deallocating the output device when multivolume output is desired.

Input

◆ No input is required when all preset functions are used; for optional functions and random access output, the programmer must supply the appropriate parameter cards.

Output

◆ This routine is preset to produce a single-volume unlabeled file, generated to a magnetic tape mounted on logical device TDG001.

When optional functions are elected, the output can be generated to a random access volume, magnetic tape, paper tape, or punched cards.

Equipment Configuration

Required

◆ Processor (65K).

Console typewriter.

Card reader or Videoscan document reader with card read feature.

Disc storage unit or drum memory unit.

Optional

◆ An additional random access device (70/564, 70/565, 70/568), magnetic tape, card punch, or paper tape punch may be used as the output device.

A printer is required if a listing of routine parameters is desired at generation time.

Routine Parameters - General

◆ The Test Data Generator routine requires no parameters for its preset options, which apply to all output devices except random access devices. For other than preset options, and for random access output, the following parameters are entered from the card reader.

Routine Parameters-General *(Cont'd)*

File Parameter

For magnetic tape, punched cards, or paper tape, this parameter describes the record length (fixed or variable); the number of blocks to be generated and the number of records per block (fixed or variable); provides for logging of routine parameters, label generation, and the fill character for unused positions of the test records.

Label Parameters for Magnetic Tape Optional

These parameters contain programmer-supplied output tape labels.

Record Parameter

For random access devices, this parameter describes the record length (fixed or variable), the number of blocks to be generated, and the number of records per block (fixed or variable); provides for logging of routine parameters and the fill character for unused positions of the test record; provides for generation of records with or without Keys; and provides three options to terminate test data generation:

1. Data will be generated for all extents for a file.
2. Data will be generated for the first extent through a given right-hand end address.
3. A specified number of records will be generated.

Label Parameters for Random Access Devices

These parameters contain file identification and programmer-supplied user header and trailer labels.

Data Parameter

This parameter describes the number and format of the data fields to be generated within each output record.

Device Deallocation Parameter

This parameter permits the device assigned to TDG001 to be deallocated and another device assigned.

END Parameter

This parameter signifies the end of parameter information.

Routine Parameters - Detailed

FILE Parameter for Magnetic Tape, Punched Cards, or Paper Tape

◆ When other than preset functions are desired, a FILE parameter is mandatory for each file to be generated on magnetic tape, punched cards, or paper tape.

Format:

ΔFILEnΔaaaa,bbbb,cccc,dddd,fpt,eeee

FILE Parameter for Magnetic Tape, Punched Cards, or Paper Tape

Card Columns	Contents	Meaning
1		Not used, leave blank.
2-5	FILE	Parameter identifier.
6	n	File identifier (any alphanumeric character).
7		Not used, leave blank
8-11	aaaa	Minimum record length (0012-9999). See note 3.
12-16	,bbbb	Maximum record length (0012-9999). See note 3.
17-21	,cccc	Minimum number of records for each output block (0001-9999).
22-26	,dddd	Maximum number of records for each output block (0001-9999).
27-28	,f	Fill character to appear within unused positions of output records: f = any alphanumeric character.
29	p	Display generation parameters to printer: p = 0 no. = 1 yes.
30	t	Generate standard labels on output file: t = 0 no. = 1 yes (destroying existing labels). = 2 yes (retaining any existing labels). See note 1.
31-35	,eeee	Number of blocks to be generated for output file (0001-9999). See note 2.

Notes:

1. When automatic label generation is selected, the following labels are produced:

Label	Serial Number	Owner Name	File ID
VOL1	TDF001	TEST∆DATA	
HDR1	TDG001		*FILEx
EOV1/EOF1	TDG001		*FILEx

When automatic label generation is not selected, the output tape will be unlabeled, or the programmer may provide the label set to be used immediately following the File Parameter. See examples on page 4-31.

---

*x = character appearing in column 6 of File card.

FILE Parameter for Magnetic Tape, Punched Cards, or Paper Tape

2. If the columns following the "eeee" entry are blank, the output file will contain the number of blocks specified. However, if desired, the programmer can force an end-of-volume condition by specifying the number of blocks to be generated for each output volume up to eight volumes. In this case, the format of the File parameter is extended as follows:

Card Columns	Content	Meaning
31-35	,eeee	Number of blocks to be generated for first output volume (0001-9999).
36-40	,eeee	Number of blocks to be generated for second output volume (0001-9999).
41-45 • •	,eeee	Number of blocks to be generated for third output volume (0001-9999).
66-70	,eeee	Number of blocks to be generated for the eighth output volume (0001-9999).

For example, to generate a file consisting of 3 output volumes, each of which contain 50 blocks:

Columns	
31-35	,0050
36-40	,0050
41-45	,0050

When this option is used, the Test Data Generator automatically deallocates the output device after each volume has been generated. The console operator must then reallocate the next device to be assigned for TDG001.

Examples:

ΔFILEAΔ0080,0080,0001,0001,A11,0200

ΔFILEBΔ0040,0222,0003,0008,A11,3333

3. This routine provides for a maximum output block of 2,000 bytes. If larger blocks are desired, additional memory can be allocated at load time, or this routine can be processed through the Linkage Editor.

Label Parameters for Magnetic Tape

◆ If the automatic generation of labels has not been requested, and the programmer wishes to supply his own label set, label parameter cards are prepared as follows. Note that these cards must follow the File parameter.

Label Parameters for Magnetic Tape (Cont'd)

Format	Remarks
ΔVOL1.....text.....	
ΔHDR1.....text.....	
ΔUHL1.....text.....	optional
ΔUTL1.....text.....	optional
ΔEOV1.....text.....	optional
ΔEOF1.....text.....	

Notes:

1. Only one label of each type is permissible. If multiple labels are provided, only the last label of the set is accepted.

2. All label cards must be 80 characters and conform to Spectra 70 standards. Column 1 of each card must be blank; the information supplied in columns 2-80 will be placed in columns 1-79 of the output label.

Record Parameter for Random Access Devices

◆ A RECRD parameter is mandatory for each file to be generated on a random access device.

Format:

ΔRECRDΔaaaa,bbbb,cccc,dddd,fpk $\left\{\begin{array}{l}\text{,mmcccch}\\ \text{,rrrrr}\end{array}\right\}$

Card Columns	Content	Meaning
1		Not used; leave blank.
2-6	RECRD	Parameter identifier.
7		Not used; leave blank.
8-11	aaaa	Minimum record length (0001-9999). See note 1.
12-16	,bbbb	Maximum record length (0001-9999). See note 1.
17-21	,cccc	Minimum number of records for each output block (0001-9999).
22-26	,dddd	Maximum number of records for each output block (0001-9999).

Record Parameter for Random Access Devices (Cont'd)

Card Columns	Content	Meaning
27-28	,f	Fill character to appear within unused positions of output records: f = any alphanumeric character.
29	p	Display generation parameters to printer: p = 0 no. = 1 yes.
30	k	Indicates if a Key field is to be generated preceding the data field of each output record. k = 0 no Key to be generated. = 1-C Indicates which data field, is defined in the DATA parameters, is to be duplicated and used as the Key. (If k = 3, the third data field defined will be used as the Key.)
31-38		Determines the termination of a run:
	blank	All extents of this file as defined in the VTOC shall be filled with data. Note: A VOL card must follow and contain the file identification.
	,mmcccch	,mmcccch The address of the rightmost limit (right-hand end) of the file beyond which a record will not be generated, where: mm = 00 when output device is disc or drum. = 00-15 magazine number for mass storage unit. cccc = cylinder number (0000-4095). h = head number (0-9). Notes: 1. If a VOL card is supplied, all extents for this file preceding this address will be filled with data. 2. If a VOL card is not supplied the user may supply a starting address (the left-hand end) through the console typewriter. (See note 2.)

(Cont'd)

Record Parameter for Random Access Devices (Cont'd)

Card Columns	Content	Meaning
31-38 (Cont'd)	,rrrrr	,rrrrr total number of records to be generated for this file. Data generation will begin at the left-hand end of the file as defined in the VTOC. Note: A VOL card must follow and contain the file identification.

Notes:

1. This routine provides for a maximum output block of 2,000 bytes. If larger blocks are desired, additional memory can be allocated at load time, or this routine can be processed through the Linkage Editor.

2. All random access volumes must be initialized by the TDOS Random Access Volume Initializer.

   It is also recommended that the volume be allocated by the TDOS Random Access Storage Allocator, as the Volume Table of Contents (VTOC) is accessed by this routine for file generation. However, the following special option exists for the non-VTOC user:

   When the routine senses that the file will be generated to random access but does not find a VOL card immediately following the RECRD parameter, it types out the following message:

   4620A NO VOL LABEL

   The programmer can then enter the starting address (left-hand end) for the file through the console typewriter as follows:

Response	Meaning
C,mmcccch	mm = magazine number for mass storage = 00 for disc or drum.
	cccc = cylinder number.
	h = head number.

Note:

The right-hand end of the file must have been supplied in the RECRD parameter card.

Examples:

1. To fill all extents of a file, RCA PAYROLL, as defined in the VTOC with unblocked records:

   ΔRECRDΔ0100,0100,0001,0001,Δ10
   ΔVOLΔRCAΔPAYROLL

DATA Parameter (Cont'd)

Card Columns	Contents	Meaning
1		Not used; leave blank.
2-5	DATA	Parameter identifier.
6		Not used; leave blank.
7-8	nn	Length of data field (01-15).
9-12	pppp	Position of the leftmost character of data field relative to the first character in the record. (The first character of the record is considered position 0000.)
13	f	Format of data field: f = 0 alphabetic. = 1 decimal, packed. = 2 decimal, zoned. = 3 binary. = 4 ASCII. = 5 Baudot (teletype). = 6 Baudot (dataspeed).
14	s	Sequence of data field: s = 0 sequential for all above formats. = 1 random for all above formats. = 2 sequential for all above formats in groups incremented by 1 after each group (see iii entry below). = 3 sequential by List entries = 4 random by List entries.
15-17	iii	Increment value for data field: a. If "s" entry is 0: iii = 001-999. Increment is converted to a hexadecimal or decimal value depending on format of data to be generated. b. If "s" entry is a 1 or 4: iii = 000. c. If "s" entry is a 2: iii = 001-999. In this case, iii specifies the number of consecutive records in the file that are to have the same data field values. Each succeeding group of records will have its value incremented by 1.

DATA Parameter (Cont'd)

Card Columns	Content	Meaning
		d. If "s" entry is a 3: iii = 001-999. In this case, iii specifies the number of times the data field will be repeated before the next data field in the card is accessed.
18-21	xxxx	a. If sequential fields have been specified: xxxx = value of the data field in the first record. b. If random fields have been specified, this entry is left blank. c. If the "s" entry is a 3 or 4, this is the number of entries in the List.
22-37	,nnppppfsiiixxxx	Requirements for data field 2, in the same format described above. (See note 1.)
38-53	,nnppppfsiiixxxx	Data field 3 requirements.
54-69	,nnppppfsiiixxxx	Data field 4 requirements.

Notes:

1. If the "s" entry is a 3 or 4, columns 22-71 are considered to be the user's data for the List.
2. No more than four data fields may be indicated in any DATA parameter; however, a total of 12 fields will be accepted for each record.
3. Entries for data fields 2, 3, and 4 are optional.

Examples:

ΔDATAΔ090000200010001,050009200020300

ΔDATAΔ030000030020004RCATOSRCAPOS
(All data records: RCARCATOSTOSRCARCAPOSPOS)

Device Deallocation Parameter

◆ This parameter is used only to deallocate the device assigned to TDG001. It is required each time that a new file is to be generated starting on a new volume.

Format:

ΔEOD

END Parameter

◆ This parameter signifies the end of parameter input and must be used when other than preset functions are desired.

Format:

ΔEND

Considerations for use

◆ 1. If multiple FILE or RECRD parameters are supplied, and the Device Deallocation parameter (EOD) is not used, a multifile output volume is produced. If the EOD parameter is supplied, the next file will begin on a new volume. See examples.

2. This routine will not generate overflow records or blocks to a random access volume.

Parameter Examples

◆ 1. Generation of a Single-Volume File

ΔRECRD
ΔVOL
(optional user label cards)
ΔDATA
ΔEND

2. Generation of a Multivolume File

ΔRECRD
ΔVOL
(optional label cards)
ΔDATA
ΔEND

3. Generation of a Multifile Volume

ΔRECRD
ΔVOL
(optional label cards)
ΔDATA
ΔRECRD
ΔVOL
(optional label cards)
ΔDATA
ΔRECRD
ΔVOL
(optional label cards)
ΔDATA
ΔEND

Parameter Examples *(Cont'd)*

4. Generation of Multifiles, with each File on a Separate Volume

```
ΔRECRD
ΔVOL
(optional label cards)
ΔDATA
ΔEOD
ΔRECRD
ΔVOL
(optional label cards)
ΔDATA
ΔEOD
ΔRECRD
ΔVOL
(optional label cards)
ΔDATA
ΔEND
```

5. To generate a single volume file of eight-character records, associating two numeric codes with each of three alphabetic fields:

```
ΔRECRDΔ0008,0008,0001,0001,A10
ΔVOLΔSTATIONΔFILE
ΔDATAΔ05000003 0020003WFILΔWCAUΔKYWΔΔ
ΔDATAΔ03000523001000211Δ15Δ
ΔEND
```

1st data record:WFILΔ11Δ
2nd data record:WFILΔ15Δ
3rd data record:WCAUΔ11Δ
4th data record:WCAUΔ15Δ
5th data record:KYWΔΔ11Δ
6th data record:KYWΔΔ15Δ
7th data record:WFILΔ11Δ

Note that the records will be repeated starting with the seventh record generated until all extents of this file are filled with data records.

Device Assignments

◆ Under Executive Control:

SDN	Device Type	Remarks
TDGRDR	Card reader.	Parameter input.
TDG001	Random access, magnetic tape, card punch, or paper tape punch.	Output device.
TDGLST	Printer.	To display parameters.

Device Assignments *(Cont'd)*

Under Monitor Control:

SDN	Device Type	Remarks
SYSIPT	Card reader.	Parameter input.
TDG001	Random access, magnetic tape, card punch, or paper tape punch.	Output device.
SYSLST	Printer.	To display parameters.

Record Parameter for Random Access Devices (Cont'd)

2. To generate 2,000 unblocked records beginning at first extent of the file, RCA PAYROLL

   ΔRECRDΔ0100,0100,0001,0001Δ10,02000
   ΔVOLΔRCAΔPAYROLL

Label Parameters for Random Access Devices

◆ When the programmer wishes to generate data to areas defined in the Volume Table of Contents, a VOL parameter card must immediately follow the RECORD parameter.

Card Columns	Format
1-5	ΔVOLΔ
6-49	File Identification (as found in the Format 1 file label in the VTOC).

If the programmer wishes to supply user Header and Trailer labels, additional label parameter cards are prepared as follows:

Format

ΔUHL1. . . . text. . . .
ΔUTL0. . . . text. . . .

Notes:

1. User labels cannot be generated for mass storage.
2. Only one label of each type is permissible. If multiple labels are provided, only the last label of the set is accepted.
3. All user label cards must be 80 characters and conform to Spectra 70 standards. Column 1 of each card must be blank; the information supplied in columns 2-80 will be placed in columns 1-79 of the output label.

DATA Parameter

◆ One DATA parameter is required for each file described. The DATA parameter must immediately follow the FILE or RECRD parameter, or the final label card, for the file to which it refers.

Format:

ΔDATAΔnnppppfsiiixxxx,. . . . . nnppppfsiiixxxx

AUTOMATIC INTEGRATED DEBUGGING SYSTEM (TOSAID)

General Description

◆ The Automatic Integrated Debugging System (AIDS) provides either a console-controlled or an automatic method for testing TOS programs. The console-controlled method is controlled by parameters entered from the console while program testing is in progress. In this case the programmer may control the test session or he may direct the operator as to what parameters to use and when. The automatic method requires no operator intervention; parameters are entered automatically by means of the card reader or from a magnetic tape device.

Although AIDS runs under Executive control, the routine modifies Executive areas and cannot, therefore, be run in the multiprogramming mode.

It is not necessary to make any special changes to programs to be tested.

All programs run under AIDS are not altered in any way.

Automatic Testing

◆ All testing in the automatic system is controlled by parameters entered from the card reader. Each programmer can set up the tests for his program; then, all tests become part of the AIDS job stream. Each program is tested until all requests have been satisfied or an unrecoverable error halt occurs. AIDS then automatically proceeds to the next program, or terminates if there is no more input.

Preset Functions

None

Optional Functions

1. Automatic assignment of work and output devices to the program to be tested.
2. Generation of test data to tape.
3. Use of run-time parameters by the program to be tested.
4. Selection of information to be displayed (memory prints, tape edits) and the display medium (printer or tape).
5. Diagnostic functions (such as traces and snapshot prints) performed as specified by input parameters.
6. Patches applied to the program or stored in the AIDS program with linkage set up between the patch and the program.

Optional

◆ Magnetic tape devices may be substituted for the card reader and the printer.

Routine Parameters - General

◆ The parameters used for automatic and console-controlled testing are summarized in the tables below.

Table 4-1. Automatic Testing Parameters

Parameter	Function
Program ID	Gives the name of the program to be tested.
Device	Informs AIDS of devices needed by program under test.
Snapshot	Requests a snapshot of specified areas of memory.
Trace	Requests a trace of all or specified instruction areas in the program.
Patch	Adds or exchanges data in the program.
End Program	Defines the end of input parameters for the current program.
RTP	Informs AIDS that run-time parameters follow.
END AIDS	Indicates the end of all AIDS input.

Console-Controlled Testing

◆ Testing in the console system is accomplished by requests entered from the console typewriter. The request may be a test parameter or may cause AIDS to read a test parameter from the card reader. Control is returned to the console each time a test is performed. The programmer can then enter another test, return control to the program being tested, or terminate the program or AIDS. Every program to be tested in the console system must be loaded individually from the console.

Preset Functions

None.

Optional Functions

1. Printing of registers and selected parts of memory.
2. Displays to the console typewriter of registers and portions of memory.
3. Changing registers and portions of memory.
4. Inserting test points in the program to be executed a specified number of times before control is returned to the console.
5. Diagnostic functions (such as traces and snapshot prints) performed as specified by parameters entered from the console or the card reader.
6. Patches applied to the program or stored in the AIDS program with linkage set up between the patch and the program.

Input

◆ The input to this routine consists of (1) a program or series (batch) of programs to be tested, (2) the user's test data or test data generated from user requirements, and (3) routine parameters entered automatically or from the console.

Output

◆ Outputs from the AIDS routine are program diagnostic data which can be displayed on the typewriter, the printer, or written to magnetic tape.

Typewriter outputs consist of memory or register displays selected by console-controlled program testing.

Printer outputs consist of tape edits, memory prints, traces, and snapshots resulting from automatic or console-controlled input parameters. Also, all typewriter messages and replys are listed on the printer.

Memory prints, tape edits, trace output, and snapshot prints can be written to magnetic tape instead of the printer for subsequent printing.

Equipment Configuration

Required

◆ Processor (65K).

Console typewriter.

Printer.

Card reader.

Other devices required by the program to be tested.

Routine Parameters - General *(Cont'd)*

Table 4-2. Console-Controlled Testing Parameters

Parameter	Function
Continue	Gives control to test program and returns control to the programmer at segment loads.
Proceed	Gives control to test program but does not return control to the programmer at segment loads.
Open Diagnostic Device	Opens AIDS output tape.
Close Diagnostic Device	Closes AIDS output tape.
Read Device	Reads parameters from card reader or magnetic tape.
Memory Print	Prints registers and selected parts of memory.
Display Memory	Displays a limited portion of memory.
Change Memory	Makes limited changes to memory.
Display Registers	Displays general purpose, floating-point, and status registers.
Change Registers	Changes general purpose or floating-point registers.
Address Stop	Inserts a test point in the program to be executed a specified number of times, then returns control to the programmer.
Snapshot	Requests a snapshot of specified areas of memory.
Trace	Requests a trace of all or specified instruction areas in the program.
Patch	Adds or exchanges data in the program.
Write Tape Mark	Writes tape marks to the AIDS output tape.
End	Defines the end of input parameters, terminates a program under test, or terminates the AIDS routine.

Routine Parameters - Automatic Testing - Detailed

◆ All automatic testing parameters are entered from the card reader or magnetic tape. The following discussions refer to the program to be tested by AIDS as the "test program".

Program ID Parameter

◆ This parameter identifies the name of the test program and must be the first parameter submitted for the program.

Format:

ΔPROGΔpppppp

Card Column	Entry	Meaning
1-6	ΔPROGΔ	Parameter identifier.
7-12	pppppp	Name of test program.

Device Parameter

◆ This parameter specifies the input, output, and work devices used by the test program. A card is submitted for each magnetic tape device used. Cards are also submitted for card readers, punches, and printers. If more than one reader, punch, or printer is used, only one card has to be submitted for each device type. The Executive will request assignment of any additional readers, punches, or printers. Device cards for magnetic tape specify whether the tape is used for input, work, or output. For input tapes, this card is also used to indicate if the input tape is to receive test data generated by AIDS.

Device parameters must be entered before any test parameters.

Format (Magnetic Tape):

ΔDEVΔaaΔddddddΔFf,Oo,CnnnnnΔ. . .ΔbΔFf,Oo,CnnnnnΔ. .Δ e

Card Column	Entry	Meaning
1-5	ΔDEVΔ	Parameter identifier.
6-8	aaΔ	aa = OT output tape. = WT work tape. = TD input tape to receive test data generated by AIDS*. = IT input tape containing user-supplied test data.
9-15	ddddddΔ	Symbolic name used by test program for device.

---

*See page 4-59, Test Data section.

Device Parameter (Cont'd)

Card Column	Entry	Meaning
16-27 (See Notes 1 & 3)	Ff	Print format for tape prints: f = G EBCDIC graphics. = H hexadecimal. If blank, the format is hexadecimal with graphic equivalents.
	,Oo	Print option: o = 0 rewind to BOT; print to double tape mark. = 1 rewind to BOT; print x blocks. = 2 rewind to BOT; print x tape marks. = 3 rewind x blocks; print x blocks. = 4 rewind x tape marks; print x tape marks. = 5 print x blocks from current position. = 6 print x tape marks from current position. = 9 print to double tape mark from current position. If an invalid print option is given, the last block read or written is printed.
	,Cnnnnn	nnnnn = decimal count for print option (00000-99999) specified by O entry. If blank, a count of 00001 is assumed.
28-34		Not used; leave blank.
35-36	bΔ	Tape mark generation at normal termination: For output and work tapes (columns 6-7 = OT or WT). b = 1 write double tape mark. = blank do not write tape mark. For all other tapes, leave blank.

Device Parameter
(Cont'd)

Format (Card Reader, Punch, or Printer):

ΔDEVΔaaΔdddddd

Card Column	Entry	Meaning
1-5	ΔDEVΔ	Parameter identifier.
6-8	aaΔ	Device type: aa = CR card reader. = PU card punch. = PR printer.
9-14	dddddd	Symbolic name used by test program for device.

RTP Parameter

◆ *Format:*

ΔRTP

Device Parameter (Cont'd)

Card Column	Entry	Meaning
37-48 (See notes 2 and 3.)	Same as cols. 16-27.	Same as columns 16-27.
49-55		Not used; leave blank.
56	e	Tape mark generation at <u>abnormal</u> termination: For output tapes (columns 6-7 = OT). e = 1 do not write tape mark. = blank write double tape mark. For work tapes (columns 6-7 = WT). e = 1 write double tape mark. = blank do not write tape mark. For all other tapes, leave blank.
57-60	Rnnn	nnn = size of record for fixed-length, blocked records (001-999) on a tape to be printed. Leave blank for variable-length or unblocked records.

Notes:

1. Columns 16-27 contain tape printing information for normal termination.
2. Columns 37-48 contain tape printing information for abnormal termination. If abnormal printing is to be the same as normal printing, place an S in column 37.
3. If this field is blank, no printing occurs. Items can appear in any order, but the first item must begin in column 16 (or 37). Commas must separate each item.

Examples:

ΔDEVΔITΔSOURCE

ΔDEVΔOTΔMASTERΔFG,O1,C00100Δ...Δ1ΔO0

ΔDEVΔTDΔINPUT1ΔFH,O3,C01000Δ....ΔS

ΔDEVΔWTΔSTORESΔ....ΔFG,O0Δ..Δ1

Snapshot Parameter

◆ The Snapshot parameter requests a listing of portions of memory, the general purpose registers, and the floating-point registers. The programmer can specify an instruction and the number of times that this instruction is to be executed before the snapshot is taken. The snapshot is reapplied each time the segment named in the parameter is loaded until the total number of snapshots desired is obtained.

Format:

ΔSNAPSΔppppppΔssssssΔllllllΔrrrrrrΔfgΔxxxxxxΔn,s,t

Card Column	Entry	Meaning
1-7	ΔSNAPSΔ	Parameter identifier.
8-14	ppppppΔ	Name of test program.
15-21	ssssssΔ	Name of segment containing area to be printed. If blank, the root segment is assumed.
22-28	llllllΔ	Left-hand end of memory area to be printed (program-relative hexadecimal address).
29-35	rrrrrrΔ	Right-hand end of memory area to be printed (program-relative hexadecimal address).
36-38	fgΔ	Format of output listings: f = H hexadecimal. = G EBCDIC graphics. = C hexadecimal with graphic equivalents. g = 1 one byte per print group. = 2 two bytes per print group. = 4 four bytes per print group.
39-45	xxxxxxΔ	Address of instruction to be used as test point (program-relative hexadecimal).
46 . . .	n	Number of times test point is to be executed before first snapshot (0-99999).
	,s	Number of times test point is to be executed between additional prints (0-99999).
	,t	Total number of prints to be taken (0-99999).

Note:

More than one snapshot can be taken using the same test point by punching different area and format information in columns 22 to 38 for each additional area to be printed.

Examples:

ΔSNAPSΔPAYROLΔSEGMT6Δ003100Δ003FA0ΔC4Δ000100Δ1,1,2

ΔSNAPSΔSDUPΔΔΔΔΔΔΔ ΔΔΔ000050Δ0005EAΔH4Δ000050Δ50,10,5

TRACE Parameter

◆ This parameter provides a diagnostic listing of an instruction and its associated registers after the instruction has been executed. Every instruction in a program may be listed or a trace made only of a selected portion of the program. The programmer can specify an instruction in the program as a test point to be executed a certain number of times before the trace is made. The trace is reapplied each time the segment named in the parameter is loaded until the total number of traces required is obtained.

The use of the Trace parameter should be kept to a minimum.

Format:

ΔTRACEΔppppppΔssssssΔllllllΔrrrrrrΔxxxxxxΔn,s,t

Card Column	Entry	Meaning
1-7	ΔTRACEΔ	Parameter identifier.
8-14	ppppppΔ	Name of test program.
15-21	ssssssΔ	Name of segment containing area to be traced If blank, the root segment is assumed.
22-28	llllllΔ	Address of first instruction to be traced (program-relative hexadecimal).
29-35	rrrrrrΔ	Address of last instruction to be traced (program-relative hexadecimal).
36-42	xxxxxxΔ	Address of instruction to be used as test point (program-relative hexadecimal).
43....	n	Number of times the test point is to be executed before the area is traced the first time (0-999999).
	,s	Number of times the test point is to be executed between any additional traces (0-99999).
	,t	Total number of traces to be made (0-99999).

Note:

A complete trace can be requested by submitting the following parameter:

ΔTRACEΔpppppp

Examples:

ΔTRACEΔINVEN

ΔTRACEΔINVENΔΔCORDERΔ00A1F0Δ00A510Δ00A200Δ0,0,1

ΔTRACEΔAIROPTΔSEG2ΔΔΔ001000Δ00100EΔ00750AΔ100,100,3

PATCH Parameter

◆ Two forms of the Patch parameter are available. One causes a branch to a patch and the other replaces the original data used by the program with new data.

The add patch feature causes the test program to branch to instructions and constants stored by AIDS. These patches may be applied anywhere in the program and applied immediately or stored for future use. Registers to be used for referencing instructions and constants within the patch may be specified. There is no limit, other than storage area needed by AIDS, to the number or size of the patches to be added.

The Exchange patch feature replaces data in the test program. The data in the patch can be graphic or hexadecimal and replaces the program data on a byte-for-byte basis. No additional general purpose registers can be used with the exchange patch.

Patches are reapplied each time the segment named in the parameter is loaded.

Format (Exchange Patch):

ΔPATCHppppppssssssEellllllΔΔΔxx...xxss

Card Column	Entry	Meaning
1-6	ΔPATCH	Parameter identifier.
7-12	pppppp	Name of test program.
13-18	sssss	Name of segment to be patched. If blank, patch is applied to the root segment.
19-20	Ee	e = G patch information is graphic. = H patch information is hexadecimal.
21-26	llllll	Address of left-hand end of area to receive patch (program-relative hexadecimal).
27-29		Not used; leave blank.

PATCH Parameter (Cont'd)

Card Column	Entry	Meaning
30-78	xx...xx	Patch information. Graphic: Up to 48 characters plus a termination indicator. If the field contains the end of the patch information, a logical NOT (11,8,7 punch) must immediately follow the last character. If the field is not the last of the information, column 78 must be blank. Hexadecimal: Up to 48 characters. Any commas used are ignored. Column 78 must be left blank whether there are additional characters or not.
78-80	ss	Sequence number (01-99) of patch card when more than one card contains information for the same patch. If blank, AIDS assumes all information is on one card.

Note:

Both graphic and hexadecimal patch cards can be used for the same patch. All cards that apply to the same patch must have the same information in columns 1 to 18 and 21 to 26.

Example:

ΔPATCHTPPARMΔ... ΔEH00070AΔΔΔFFF04A21,03,A47E

Format - (Add Patch):

ΔPATCHppppppssssssAallllllicΔxx...xxss

Card Column	Entry	Meaning
1-6	ΔPATCH	Parameter identifier.
7-12	pppppp	Name of test program.
13-18	ssssss	Name of segment to be patched. If blank, the patch is applied to the root segment.
19-20	Aa	a = I patch information is instructions. = G patch information is graphic constants. = H patch information is hexadecimal constants.

PATCH Parameter (Cont'd)

Card Column	Entry	Meaning
21-26	llllll	Address of last instruction to be executed before patch (program-relative hexadecimal).
27	i	Number of general purpose register to be used with added instructions (0-F). If not needed, leave blank. (See notes 2 and 4.)
28-29	cΔ	Number of general purpose register to be used with added constants (0-F). If not needed, leave blank. (See notes 3 and 4.)
30-78	xx...xx	Patch information. Graphic Constants: Up to 48 characters plus a termination indicator. If the field contains the end of the graphic constants, a logical NOT (11,8,7 punch) must immediately follow the last character. If the field is not the last of the constants, column 78 must be blank. Instructions and Hexadecimal Constants: Up to 48 columns of hexadecimal constants and instructions. Commas may be used in instructions for convenience but they are ignored. Column 78 must be left blank whether there are additional constants or instructions or not.
79-80	ss	Sequence number (01-99) of patch card when more than one card contains information for the same patch. If blank, AIDS assumes all information is on one card.

Notes:

1. Both graphic and hexadecimal patch cards can be used for the same patch. All cards that apply to the same patch must have the same information in columns 1 to 18 and 21 to 29. When both instructions and constants are added, instructions must be entered first. Constants may not be added without preceding instructions.
2. The contents of i are stored and the address of the left-hand end of the added instructions is placed in i when the patch is made.
3. The contents of c are stored and the address of the left-hand end of the added constants is placed in c when the patch is made.
4. The contents of the registers specified by i and c before the patch was applied are not restored until all of the added instructions have been executed.

Example:

ΔPATCHDESU01ΔΔΔ ΔΔΔ AG040EB6Δ3Δ00$CONEND*

* = 11,8,7 punch

End Program Parameter

◆ This parameter indicates the end of test parameters for the current program. The programmer can specify that a memory dump is to be taken upon normal termination and its format. The format of the abnormal termination memory dump can also be specified.

Format:

ΔENDΔPROGΔfgΔmn

Card Column	Entry	Meaning
1-10	ΔENDΔPROGΔ	Parameter identifier.
11-13	fgΔ	Format of memory print upon normal termination: f = H hexadecimal. = G EBCDIC graphics. = C hexadecimal with graphic equivalents. = F floating-point. = M mnemonic. g = 1 one byte per print group. = 2 two bytes per print group. = 4 four bytes per print group. If no memory print is desired, leave blank.
14-15	mn	Format of memory print upon abnormal termination: Same as f and g. If blank, AIDS prints memory in full-word hexadecimal with graphic equivalents.

END AIDS Parameter

◆ This parameter indicates the end of the AIDS job stream.

Format:

ΔENDΔAIDS

Routine Parameters for Console-Controlled Testing - Detailed

◆ Parameters used for console testing are divided into two groups: immediate and latent. Immediate functions are entered from the console typewriter and executed as soon as they are entered, except for Address Stop. Latent functions are stored by AIDS and executed only when certain conditions have been satisfied. Two latent functions, Trace and Snapshot, can be entered from the typewriter, card reader, or magnetic tape; the third, Patch, can only be entered from the card reader or magnetic tape.

Continue Parameter

◆ This parameter indicates that control is to be returned to the test program at the last point of interruption or to a specific address. The programmer is given control again with the next load of a segment of the test program.

Format:

pΔ@CONΔaΔxxxxxx

Type Position	Entry	Meaning
1-2	pΔ	Program number of test program (1-6).
3-7	@CONΔ	Parameter identifier.
8-15	aΔxxxxxx	a = A absolute address. = P program-relative address. xxxxxx = six-character hexadecimal address of instruction where control is to be transferred. If blank, control is transferred to the last point of interrupt (address contained in P counter).

Proceed Parameter

◆ This parameter is the same as the Continue parameter except that the programmer does not regain control unless an Address Stop parameter has been previously entered.

Format:

pΔ@PROΔaΔxxxxxx

Type Position	Entry	Meaning
1-2	pΔ	Program number of test program (1-6).
3-7	@PROΔ	Parameter identifier.
8-15	aΔxxxxxx	a = A absolute address. = P program-relative address. xxxxxx = six-character hexadecimal address of instruction where control is to be transferred. If blank, control is transferred to the last point of interrupt (address contained in P counter).

Open Diagnostic Device Parameter

◆ An output diagnostic tape is opened using this parameter. A standard header label is written and the tape is used for memory prints, snapshots, and trace outputs.

Format:

pΔ@OPD

Type Position	Entry	Meaning
1-2	pΔ	Program number of test program (1-6).
3-6	@OPD	Parameter identifier.

Close Diagnostic Device Parameter

◆ This parameter writes a double tape mark and deallocates the magnetic tape assigned by the Open Diagnostic Device parameter. The output tape is rewound.

Format:

pΔ@CLD

Type Position	Entry	Meaning
1-2	pΔ	Program number of test program (1-6).
3-6	@CLD	Parameter identifier.

Read Device Parameter

◆ The Read Device parameter causes AIDS to read parameters and/or associated data from the card reader or magnetic tape. The input is read until an END parameter is recognized.

Format:

pΔ@RDVΔdd

Type Position	Entry	Meaning
1-2	pΔ	Program number of test program (1-6).
3-7	@RDVΔ	Parameter identifier.
8-9	dd	Device class: dd = 00 magnetic tape. = 05 card reader.

Memory Print Parameter

◆ This parameter causes a listing of selected portions of memory, the general registers, the floating-point registers, and the program status registers. This listing may be directed to the printer or to magnetic tape. If a tape device is used, it must have been assigned by an Open Diagnostic Device parameter.

Printer output is full-word hexadecimal with graphic equivalents, 48 bytes to the line. Duplicate lines are suppressed. Tape output is 133-character, unbatched records.

Format:

pΔ@DMPΔaΔllllllΔrrrrrrΔfΔn

Type Position	Entry	Meaning
1-2	pΔ	Program number of test program (1-6).
3-7	@DMPΔ	Parameter identifier.
8-9	aΔ	a = A absolute addresses. = P program-relative addresses.
10-16	llllllΔ	Address of left-hand end of memory area to be printed (hexadecimal).
17-23	rrrrrrΔ	Address of right-hand end of memory area to be printed (hexadecimal).
24-25	fΔ	Format in which memory is to be printed: f = M mnemonic. = S short-precision floating-point. = F long-precision floating-point. = H hexadecimal. = G EBCDIC graphics. = C hexadecimal with graphic equivalents.
26	n	Printer grouping factor: n = 1 one byte per print group. = 2 two bytes per print group. = 4 four bytes per print group. Not used when f is equal to M, S, or F.

Note:

To print registers and all of memory assigned to the program under test, enter pΔ@DMP only.

Display Memory Parameter

◆ This parameter can be used to display a memory area up to 99 bytes to the console typewriter. Twenty-four bytes are displayed per line with each line preceded by the hexadecimal address of the leftmost character in the line.

Format:

pΔ@DMYΔaΔ111111Δnn

Type Position	Entry	Meaning
1-2	pΔ	Program number of test program (1-6).
3-7	@DMYΔ	Parameter identifier.
8-9	aΔ	a = A absolute address. = P program-relative address.
10-16	111111Δ	Address of left-hand end of memory to be displayed (hexadecimal).
17-18	nn	Number of bytes to be displayed (1-99).

Change Memory Parameter

◆ This parameter allows limited changes to be made to memory during program testing. The changes are made on a byte-for-byte basis. The area to be changed is specified by giving an absolute or program-relative address of its left-hand end. A maximum of 52 bytes can be changed with each Change Memory parameter.

Format:

pΔ@CMYΔaΔ111111ΔdΔxx...xx

Type Position	Entry	Meaning
1-2	pΔ	Program number of test program (1-6).
3-7	@CMYΔ	Parameter identifier.
8-9	aΔ	a = A absolute address. = P program-relative address.
10-16	111111Δ	Address of left-hand end of memory to be changed (hexadecimal).
17-18	dΔ	d = H Hexadecimal data. = G Graphic data.
19-70	xx...xx	Hexadecimal or graphic data to be inserted. The last character of graphic data cannot be a space.

Display Registers Parameter

◆ One or more state 1 general purpose registers, the floating-point registers, or the program status registers are displayed on the typewriter by using this parameter. The general purpose registers are displayed in full-word hexadecimal, preceded by the register number (0-F); program status registers are printed in full-word hexadecimal, preceded by an identification tag.

Format:

pΔ@DRGΔrΔf

Type Position	Entry	Meaning
1-2	pΔ	Program number of test program (1-6).
3-7	@DRGΔ	Parameter identifier.
8-9	rΔ	Registers to be displayed: r = 0-F general purpose register. = G floating-point registers. = P program counter and IMR.
10	f	f = final general purpose register to be displayed (registers r through f will be displayed). If blank, only the register specified by r is displayed.

Change Registers Parameters

◆ This parameter allows the programmer to change the contents of the general purpose and the floating-point registers. The same data can be placed in more than one general purpose register by one parameter. A series of parameters is used to load the general purpose registers with different information.

Format (Change General Purpose Register):

pΔ@CRGΔrΔhhhhhhhh

Type Position	Entry	Meaning
1-2	pΔ	Program number of test program (1-6).
3-7	@CRGΔ	Parameter identifier.
8-9	rΔ	General purpose register to be changed (0-F).
10-17	hhhhhhhh	Data to be inserted into register (full-word hexadecimal).

Change Registers Parameter (Cont'd)

Format (Change More than One General Purpose Register):

pΔ@CRGΔrΔfΔhhhhhhhh

Type Position	Entry	Meaning
1-2	pΔ	Program number of test program (1-6).
3-7	@CRGΔ	Parameter identifier.
8-9	rΔ	First general purpose register to be changed (0-E).
10-11	fΔ	Last general purpose register to be changed (1-F).
12-19	hhhhhhhh	Data to be inserted into each register (full-word hexadecimal).

Format (Change Floating-Point Register):

pΔ@CRGΔrs.hhh...hhhΔEsee

Type Position	Entry	Meaning
1-2	pΔ	Program number of test program (1-6).
3-7	@CRGΔ	Parameter identifier.
8	r	Floating-point register to be changed (0,2,4, or 6).
9	s	Sign of number to be inserted in register (+ or -).
10	.	Decimal point.
11-25	hhh...hhhΔ	Hexadecimal value to be inserted in register (zero-filled).
26-29	Esee	s = sign of exponent (+ or -). ee = exponent ($00_{(16)}$- $65_{(16)}$). Value can range from -$65_{(16)}$ to + $64_{(16)}$.

Example:

5Δ@CRGΔ2+.000E074B604A00ΔE-08

Address Stop Parameter

◆ The Address Stop parameter specifies a test point (instruction address) in the test program where control is to be returned to the programmer. This parameter can cause the point to be executed a certain number of times before control is given to the programmer.

Up to three address stops can be stored by AIDS at one time. Address Stop parameters are not reapplied and must be entered each time they are to be used.

Format:

pΔ@STPΔaΔxxxxxxΔsΔnnnn

Type Position	Entry	Meaning
1-2	pΔ	Program number of test program (1-6).
3-7	@STPΔ	Parameter identifier.
8-9	aΔ	a = A absolute address. = P program-relative address.
10-16	xxxxxxΔ	Address of instruction to be used as test point (hexadecimal).
17-18	sΔ	Address stop identifier (1, 2, or 3). Identifies the test point as one of three which can be in AIDS at the same time. When a parameter is entered with the same identifier as an existing parameter, it replaces the existing parameter.
19-22	nnnn	Number of times the test point is to be executed before control is returned to the programmer (0-9999).

Snapshot Parameter

◆ This parameter provides a listing of specified portions of memory, the general purpose registers, and the floating-point registers. The programmer can specify an instruction address and the number of times the instruction is to be executed before the snapshot is taken. The Snapshot parameter is entered from the console typewriter, card reader, or magnetic tape device.

When the typewriter is used, the output is in full-word hexadecimal with graphic equivalents. The snap is taken as soon as the test point is satisfied.

When the card reader or magnetic tape is used, the programmer can select the format and grouping of the output. He can also specify additional snapshots after the first.

Format - Card:

Same as described for automatic testing; see page 4-42.

Format - Typewriter:

pΔ@SNPΔaΔllllllΔrrrrrrΔxxxxxxΔnnnn

Type Position	Entry	Meaning
1-2	pΔ	Program number of test program (1-6).
3-7	@SNPΔ	Parameter identifier.
8-9	aΔ	a = A absolute addresses. = P program-relative addresses.
10-16	llllllΔ	Address of left-hand end of memory area to be printed (hexadecimal).
17-23	rrrrrrΔ	Address of right-hand end of memory area to be printed (hexadecimal).
24-30	xxxxxxΔ	Address of instruction to be used as a test point (hexadecimal).
31-34	nnnn	Number of times the test point is to be executed before the area is printed (1-9999). If blank, 0 is assumed.

Trace Parameter

◆ This parameter provides a diagnostic listing of an instruction and its associated registers after the instruction has been executed. Every instruction in a program may be listed or a trace made only of a selected portion of the program. The programmer may specify a test point (instruction address) in the program that is to be executed a certain number of times before the trace is made.

The use of this parameter should be kept to a minimum and only when other AIDS functions cannot solve the problem.

This parameter can be entered from the console typewriter, card reader, or magnetic tape device. When a card reader or magnetic tape is used, additional traces after the first one can be specified.

Format - Card:

Same as described for automatic testing; see page 4-43.

Format - Typewriter:

pΔ@TRCΔaΔllllllΔrrrrrrΔxxxxxxΔnnnn

Type Position	Entry	Meaning
1-2	pΔ	Program number of test program (1-6).
3-7	@TRCΔ	Parameter identifier.
8-9	aΔ	a = A absolute addresses. = P program-relative addresses.
10-16	llllllΔ	Address of first instruction to be traced (hexadecimal).
17-23	rrrrrrΔ	Address of last instruction to be traced (hexadecimal).
24-30	xxxxxxΔ	Address of instruction to be used as a test point (hexadecimal).
31-34	nnnn	Number of times the test point is to be executed before the trace is made (1-9999). If blank, 0 is assumed.

Note:

A complete trace is made by submitting only pΔ@TRC in the parameter.

Patch Parameter

◆ The Patch parameter functions and format are identical to those of the automatic testing Patch parameter described on page 4-44.

Write Tape Mark Parameter

◆ This parameter causes a single or double tape mark to be written to a designated magnetic tape used by the test program. The tape may be rewound to BOT if desired.

Format:

pΔ@WTMΔddddddΔtΔr

Type Position	Entry	Meaning
1-2	pΔ	Program number of test program (1-6).
3-7	@WTMΔ	Parameter identifier.
8-14	ddddddΔ	Symbolic name used by test program for device.
15-16	tΔ	t = 0 do not write tape mark. = 1 write single tape mark. = 2 write double tape mark.
17	r	r = 1 rewind to BOT. = 2 do not rewind. = 3 rewind to tape mark. When t = 0, the tape is rewound to BOT and the r entry is ignored.

END Parameter

◆ Two End parameters are used. One is entered from the card reader or magnetic tape, the other from the typewriter.

Card or tape parameters are used to signify the end of Snapshot, Trace, and Patch input. An End card must follow each parameter or each set of parameters to be read following a Read Device statement.

An End parameter entered from the typewriter signifies that: (1) AIDS is to be terminated and control returned to the program under test, (2) AIDS and the program under test are to be terminated, or (3) the program under test is to be terminated and AIDS used to test another program.

Format - Card:

Δ@END

Format - Typewriter:

pΔ@ENDΔz

Type Position	Entry	Meaning
1-2	pΔ	Program number of test program.
3-7	@ENDΔ	Parameter identifier.
8	z	z = A terminate AIDS and return control to the test program. = B terminate AIDS and test program. = C terminate test program and continue AIDS routine.

Considerations For Use

General

◆ AIDS offers two methods of testing a program: automatic and console-controlled. The programmer must therefore select the method best suited to meet his needs and that will utilize both his and the processor's time effectively.

Automatic Testing

◆ In an automatic AIDS session, a minimum of operator intervention is required because all user data concerned with the test is on cards or tape, or both. The operator loads AIDS using the Executive and responds to AIDS messages to start the testing. The testing proceeds automatically until completed or an abnormal termination occurs. At this time, another program may be tested or the same program tested with different data.

The considerations for using automatic AIDS testing are discussed below.

Device Assignments

The programmer must supply a Device parameter for each magnetic tape used by the test program. He must also supply a Device parameter for card readers, punches, and printers. If a program uses more than one reader, punch, or printer, only one parameter is needed for each device type. The Executive will request assignment of any additional readers, punches, or printers.

Assignments are normally made by the Executive for the first program to be tested in a session. AIDS keeps these devices in a pool and automatically uses them for additional programs in the testing stream. Exceptions to this are that the assignment of an input device is always requested, and work and output device assignments are requested if they are not available in the device pool. Any new devices assigned are added to this pool.

If a device card is not supplied for a device used by the test program, the Executive asks for its assignment. When the program under test terminates, the device is deallocated and is not added to the AIDS device pool.

Test Data

Test data is available for program testing from the following sources:

1. User-supplied data recorded on cards or tape.
2. The TOS Test Data Generator routine (part of AIDS) may be used to generate test data to a tape that is used as input to the test program. The data is prepared according to Test Data Generator parameters (refer to page 4-2, Test Data Generator routine). The parameters must immediately follow the Device card that defines the tape that is to receive the test data.

Automatic Testing (Cont'd)

Regardless of the source, all data must be labeled as required by the program under test. Different sets of test data may be used to test the same program by repeating the Program ID parameter in the AIDS input and using new test data.

Run-Time Parameters

When a test program requires run-time parameters (RTP), they must be supplied as part of the AIDS job stream. The RTP's are preceded by the AIDS parameter RTP. The following run-time parameters are valid input: // FILES, // VOL, // TPLAB, // VDC, // END.

The format of these parameters is the same as described in the TDOS Operators' Guide, except for the // VDC card. The volume serial number in the // VDC parameter for a file allocated by AIDS must be all zeros. For user supplied files the // VDC serial number must be the same as the Volume serial number.

Tape Printing

AIDS does not save work or output tapes. If the programmer wants a record of any of these tapes he must indicate in the Device card associated with the tape that it is to be printed. The tapes will be printed, or written to an output tape, depending on the assignment of AIDOPT. AIDS can print both normal and preedited tapes. Partial printing of tapes is possible by using the count option in the Device parameter.

Memory Dumps

A memory dump is always taken by AIDS upon an abnormal termination. The programmer can also specify that a memory dump be taken upon normal termination.

The memory dump routine is part of AIDS and cannot be called in under Executive control. The dump is made to AIDOPT, which may be assigned to the printer or magnetic tape. When an abnormal memory dump is taken, it is preceded by a description of the type of error causing the dump and the contents of the P1 counter at that time.

AIDS Output

AIDS output is made to the printer or a magnetic tape depending on AIDOPT assignment when AIDS was initiated. When both AIDS output and the test program output are assigned to the same printer they will share it. The information and data contained in the AIDS output is listed below.

1. A listing of all AIDS parameters submitted and reasons for rejection if any were invalid.
2. Results of snapshots or traces that were requested.
3. The location and type of any interrupts that occurred.
4. All typewriter messages and replies.
5. All memory dumps.
6. Edited contents of all or parts of magnetic tapes selected for printing.

Console-Controlled Testing

◆ After AIDS has been loaded, the message AUTO OR CONSOLE? is typed. The programmer then supplies the name of the program to be tested and the symbolic names used by the program for the card reader and/or printer if these devices are to be shared with AIDS.*

If the program to be tested is in memory, AIDS types the message AIDS REQUEST REQUIRED; the programmer then starts his program testing. If the program is not in memory, AIDS idles until the test program is loaded and device assignments are made in the normal TOS Executive manner before AIDS input is requested. (If the program to be tested was loaded before AIDS, the Executive Change Priority routine must be used to give AIDS a higher priority than that of the program to be tested.)

The programmer begins testing by submitting any of the console-controlled testing parameters. The operation given in the parameter is executed and control is returned to the programmer, with the exception of the Address Stop and the latent parameters. If one of the latter two parameters is submitted or no input is desired at this time, control must be given to the program under test. Control is given to the program only by a Continue or Proceed parameter or a reply after an unsolicited interrupt. The programmer regains control when (1) a segment is loaded after a Continue parameter had been entered, (2) an Address Stop is satisfied, (3) an End of Job SVC is executed by the program under test, or (4) an unsolicited interrupt is typed in. These conditions of program and user control are described below.

Continue

When this parameter is entered, control is given to the program at (1) the address specified, (2) the start address of the program if it was just loaded, or (3) the address in the LPOV statement if a segment of the program was just loaded. The program then executes until another segment is loaded or an Address Stop is completed, at which time control is returned to the programmer.

Proceed

The Proceed parameter gives control to the program under test in the same manner as Continue except that control is returned only after an Address Stop has been completed.

Unsolicited Interrupt and Reply

The programmer can regain control at any time by entering an Executive or test program interrupt. AIDS then allows the programmer to type in a message to the program's unsolicited type-in logic or submit additional testing parameters. Control is returned to the program under test or AIDS depending on the type-in.

---

*The format for the reply to AUTO OR CONSOLE? message appears in the Operators' Guide.

Console-Controlled Testing (Cont'd)

Program Segment Loaded

When a segment of the program is loaded after a Continue parameter has been entered, a message is typed requesting input and giving the name of the segment just loaded. Any of the AIDS parameters can then be entered.

Address Stop

After the instruction specified as a test point in an Address Stop statement has been executed the required number of times, AIDS returns control to the programmer. AIDS types a message containing the address of the test point; the programmer can then request any of the AIDS functions.

End of Job SVC

The occurrence of an EOJ SVC in the program causes AIDS to return control to the programmer. AIDS types a message requesting input and giving the address of the supervisor call. Any of the AIDS functions can be entered at this time. If a Continue or Proceed parameter is entered, an address must be specified.

Restrictions

◆

1. Test point addresses specified for Address Stop, Snapshot, and Trace parameters must be the leftmost byte of the instruction.
2. If the test program modifies a location used as an AIDS test point, the function associated with the test point will not be executed.
3. A test point address within a program segment can not be used for more than one AIDS function. Identical test points can be used if they are in different segments.
4. Snapshots, address stops, and add patches can not be specified within an area to be traced.
5. The location specified by the address in an Add Patch parameter can not contain a test program SVC.

Device Assignments

◆ Console-Controlled Testing

SDN	Device Type	Remarks
AIDRDV	Card reader or magnetic tape.	Input device for Snapshot, Trace, and Patch parameters.
AIDOPT	Printer or magnetic tape.	Output device for memory print and snapshot and trace results.

Automatic Testing

SDN	Device Type	Remarks
AIDIPT	Card reader or magnetic tape.	Parameter input device.
AIDOPT	Printer or magnetic tape.	Output device.

Parameter Examples

Automatic Testing

- AIDIPT (Card Reader)

Card	Parameter
A1	ΔPROGΔTPPR
A2	ΔDEVΔTDΔPRIPT1ΔFG,O1,C00100Δ... ΔΔΔFH,O0
A3	Test Data Generator parameter cards
A4	ΔEND
A5	ΔDEVΔRDΔPRPRM
A6	ΔDEVΔPRΔPRLST
A7	Test cards - SNAPS, TRACE, PATCH
A8	ΔENDΔPROG
B1	ΔPROGΔX
B2	ΔDEVΔITΔINPUT1
B3	ΔDEVΔWTΔWORK1ΔΔ FG,O0Δ... ΔΔΔ SΔ... Δ1
B4	ΔDEVΔOTΔOUTPUTΔFG,O3,C01000Δ... ΔΔΔO0
B5	ΔRTP
B6	//ΔVOLΔINPUT1,STATS1
B7	//ΔTPLABΔ'RCAΔEDPΔ...Δ67248Δ67345'
B8	//ΔVOLΔOUTPUT,STATS2
B9	//ΔTPLABΔ'RCAΔEDPΔ...Δ67295Δ68020'
B10	//ΔEND
B11	Test cards - SNAPS, TRACE, PATCH
B12	ΔENDΔPROGΔΔΔΔH4
C1	ΔPROGΔY
C2	ΔDEVΔITΔSALESΔΔFG,O0Δ... ΔΔΔS
C3	ΔDEVΔPUΔORDER
C4	ΔDEVΔOTΔINVENTΔ.... ΔO2,C00050
C5	Test cards - SNAPS, TRACE, PATCH
C6	ΔENDΔPROGΔG2ΔG2
C7	ΔENDΔAIDS

The series of cards shown above would automatically test three programs named TPPR, X, and Y. An explanation of the function of each card follows.

Automatic Testing (Cont'd)

Card B5	- Indicates that test program run-time parameters follow.
Cards B6 thru B10	- Run-time parameters required by the test program.
Card 11	- AIDS test parameter cards.
Card 12	- Indicates that there are no more AIDS input cards for Program X. No memory dump will be made after a normal termination. Memory will be printed in four-byte hexadecimal groups if an abnormal termination occurs.

Automatic Testing (Cont'd)

Card A1 - Defines the first program to be tested as TPPR.

Card A2 - Informs AIDS that test data is to be generated on an input tape called PRIPT1. If a normal termination occurs, print the first 100 records on PRIPT1 in EBCDIC graphics. For an abnormal termination print the complete tape in hexadecimal.

Card A3 - Test Data Generator parameter cards for data to be generated onto PRIPT1.

Card A4 - The END card for the Test Data Generator parameters.

Card A5 - Informs AIDS that PRPRM, an input device for TPPR, is a card reader. Since AIDIPT is a card reader, AIDS will assume that the PRPRM device is the same card reader and no device assignment for PRPRM will be requested. If AIDIPT were magnetic tape, AIDS would request assignment of PRPRM.

When AIDS and program TPPR share the card reader, AIDS informs the operator to remove the AIDS control cards and insert the card input for the test program. When program TPPR testing has been completed, AIDS informs the operator to replace the remaining AIDS control cards.

Card A6 - Informs AIDS that PRLST (the TPPR output device) is the printer. If AIDOPT is also the printer, AIDS and TPPR will share it and PRLST will be assigned automatically.

Card A7 - The AIDS test parameters would appear at this point in the input.

Card A8 - Signifies that there are no more AIDS input cards. Because the additional fields of the card are blank, no memory dump will be taken after a normal termination but a memory dump will be made upon abnormal termination.

Card B1 - Informs AIDS that program X is to be tested.

Card B2 - Informs AIDS that an input tape containing programmer data is to be used and that it is called INPUT1. This tape is not be to printed when program X terminates.

Card B3 - Device card for a work tape named WORK1. The entire tape will be printed upon either a normal or abnormal termination.

Card B4 - Device card for an output tape called OUTPUT. At normal termination the tape will be rewound and the last 1000 records will be printed in EBCDIC graphics. If an abnormal termination occurs, the entire tape will be printed in hexadecimal with graphic equivalents.

Automatic Testing (Cont'd)

Card C1 - Defines program Y as the next program to be tested.

Card C2 - Indicates a programmer's input tape called SALES. The entire tape is to be printed in EBCDIC graphics upon both a normal and abnormal termination.

Card C3 - Informs AIDS that a card punch called ORDER is used by program Y.

Card C4 - Informs AIDS that an output tape called INVENT is used by program Y. The tape is not to be printed after normal termination. An abnormal termination will cause the tape to rewind to BOT and be printed through the first 50 tape marks.

Card C5 - AIDS test parameters.

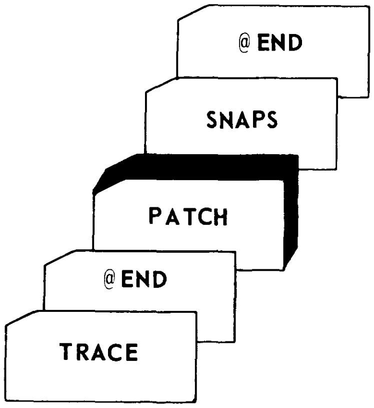
Card C6 - Indicates that there are no more AIDS input cards for program Y. A memory dump is to be taken upon normal or abnormal termination. Its format will be graphic in two-byte groups.

Card C7 - Indicates the end of the AIDS session. This card is placed at the end of the AIDS input by the operator.

In the preceding example all cards, except cards A9 and A10 could be on magnetic tape. In this case, AIDS would only access the card reader for the TPPR parameters and END card.

Console-Controlled Testing

◆ No specific examples of console-controlled testing input can be given since it is entirely up to the programmer which parameters are desired. The only input format that has to be followed is the use of the @END card when the TRACE, SNAPS, or PATCH parameters are entered from the card reader or magnetic tape.



AIDRDV

The first time a Read Device parameter is entered, AIDS will read the Trace parameter. The second time a Read Device is used, the Patch and Snaps parameters will be read.

Logging Messages

```
PROG JSMCD   INITIATED FOR TESTING WITH AIDS

DEV RD SYSIPT

DEV 01 SYSLST 00                   S                                NO COUNT,ONE INSERTED IF REQUIRED
```

Device Cards

```
                       AIDS REQUESTS FOR JSMCD
SNAPS JSMCD        000F34 000F68 C4 000F34 0,0,10

TRACE JSMCD        000F76 000F9A 000F72 0,0,6

PATCHJSMCD       EG00111E   EXCG PATCH                              NO END SENTINEL

PATCHJSMCD       EG00111E   EXCG PATCH-

END PROG C4
```

Test Parameters

```
TEST PROGRAM T/W MESSAGE
   ARE CARDS SYSIPT AVAIL?
Y
```

Console Typewriter Message and Response

```
         A DATA ERROR INTERRUPT  HAS OCCURRED AT PROG REL 000F3C ,P CTR 0057F4
¤¤
```

AIDS Error Interrupt Message

```
                        AIDS REQUESTS FOR JSMCK1

  IAPS JSMCK1        000A32 000AE9 C4 000B3C 0,1,10

 TRACE JSMCK1        000B70 000BB8 000B70 0,0,6

 PATCHJSMCK1      AI000C5234 D221202B202A,D211202A4000,98F12242,41E03018      01

 PATCHJSMCK1      AI000C52   47FF004C,9240202A                                02

 PATCHJSMCK1      AG000C52   EXECUTED ADD PATCH-                              03

 PATCHJSMCK1      EG000AEE   EXCG PATCH-
```

Test Parameters

Snapshot

```
                                              AIDS SNAPSHOT AT  P 00B3C

                                            IMR FFF38E03     IFR 00000000

                           0         1         2         3         4         5         6         7
 P1 GEN REGS            00000000  00003BA8  4F0045D6  0000000   00000000  00000000  00000000  00000000

                           8         9         A         B                   D         E         F
                        00000000  00000000  00000000  00000000  00000000  00000000  000046F8  00003D30

 FLT PT REGS      .0000000000000 E-40   .0000000000000 E-40   ,0000000000000 E-40   0000000000000 E-40

                     B 0     0 3 5 1   9 0 0 0   0 5 4 3   0 0 4 5   6 7 8 0   0 8 7 2   3 1 0 1   3 2 3 6   6
P 00A30    20CAC2F0  F0F3F5F1  F9F0F0F0  F0F5F4F3  F0F0F4F5  F6F7F8F0  F0F8F7F2  F3F1F0F1  F3F2F3F6  F6404040

           B 0 0 3   5 1 9 0   1 3 2 3   6 6
P 00A58    C2F0F0F3  F5F1F9F0  F1F3F2F3  F6F64040  40404040  40404040  40404040  40404040  40404040  40404040

P 00A80   *40404040  40404040  40404040  40404040  40404040  40404040  40404040  40404040  40404040  40404040

                                         B 0 0 3   5 1                 ? A B
P 00AD0    40404040  40404040  40404040  C2F0F0F3  F5F10000  000C0132  006FC1C2
EXECUTED ADD PATCH    }
EXCG PATCH            }      Test Program Output
```

Trace

```
  LC     OP LMGR B1   CB1     D1     1EA     X2    CX2   B2   CB2     D2     2EA   C          1                      2

* 0471E  47  D0                                           2   045D6   1DE    047B4 00
* 0471E  47  D0                                           2   045D6   1DE    047B4 02
* 04722  F2  36  2   045D6   0B4    0468A                 2   045D6   012    045E8 02    0000750F               F0F0F0F0F7F5F0
* 04728  F2  36  2   045D6   0B8    0468E                 2   045D6   019    045EF 02    0000750F               F0F0F0F0F7F5F0
* 0472E  FA  33  2   045D6   0B4    0468A                 2   045D6   0B8    0468E 02    0001500C               0000750F
* 04734  F2  36  2   045D6   0B8    0468E                 2   045D6   00B    045E1 02    0000700F               F0F0F0F0F7F0F0
* 0473A  FB  33  2   045D6   0B4    0468A                 2   045D6   0B8    0468E 02    0000800C               0000700F
* 04740  F2  36  2   045D6   0B8    0468E                 2   045D6   020    045F6 02    0000800F               F0F0F0F0F8F0F0
* 04746  FB  33  2   045D6   0B4    0468A                 2   045D6   0B8    0468E 00    0000000C               0000800F
* 0474C  47  70                                           2   045D6   1DE    047B4 00
* 04750  D2  05  2   045D6   0AE    04684                 2   045D6   004    045DA 00    C2F0F0F0F3F5           C2F0F0F0F3F5
* 04756  92  00  2   045D6   13F    04715                                          00    00
* 0475A  D2  06  2   045D6   02A    04600                 2   045D6   004    045DA 00    C2F0F0F0F3F5F8         C2F0F0F0F3F5F8
* 04760  D2  06  2   045D6   031    04607                 2   045D6   020    045F6 00    F0F0F0F0F8F0F0         F0F0F0F0F8F0F0

B0003580000800     Test Program Output

**
```

LC - instruction location (absolute)

OP - operation code

LMGR - instruction length, mask, or general register byte

B1 - first operand base register number

CB1 - contents of B1

D1 - first operand displacement

1EA - generated address for first operand

X2 - second operand index register number

CX2 - contents of X2

B2 - second operand base register number

CB2 - contents of B2

D2 - second operand displacement

2EA - generated address for second operand

C - condition code after instruction execution

1 - first ten bytes of data referenced by first operand

2 - first ten bytes of data referenced by second operand

Memory Print

```
                                                AIDS MEMORY PRINT

                                              IMR FFF3FE03     IFR 00000000

                           0          1          2          3          4          5          6          7
  P1 GEN REGS          00000000   000048B8   4F00577A   000048C0   00005F40   00006F40   8F006A52   00007184

                           8          9          A          B          C          D          E          F
                       00000000   00008BC8   00006F51   000050FA   AF005076   8F00980C   000057B8   00004AA0

  FLT PT REGS      .0000000000000 E-40       .0000000000000 E-40       .0000000000000 E-40       .0000000000000 E-40

               ¤ ¤                                                     |     :                     ¬           ?
A 04800    0010FFFF    0001000F    4001EFBA    00000000    000048B8    4F00577A    000048C0    00005F40    00006F40    8F006A52

                                         H         ?           &           &                                   ¢
A 04828    00007184    00000000    00008BC8    00006F51    000050FA    AF005076    8F00980C    000057B8    00004AA0    00000000

                                                                                                               ¬ ¤
A 04850    00000000    00000000    00000000    00000000    00000000    00000000    00000000    00005997    00005FFF    40000060

               U K                     ?           &                                                           U
A 04878    0001E4D2    0F01EDBE    00006F51    000050FA    00000000    00000000    00000000    00000000    0001E4FA    C0000000

                                       U                                           S Y S I     P T
A 048A0    00000000    00000000    0001E4B8    00000000    00000000    00000000    E2E8F2C9    D7E30510    000048E0    00003AA0

                 Y                                                                                   &
A 048C8    040048E8    00030000    05005920    00000048    00000080    00000000    050058D0    00000050    00000000    00000000

             0 C           S                                           F I L E     A
A 048F0    00F0C300    0000E283    00000000    00000000    000058D0    C6C9D3C5    C1404000    00040000    00000000    0000003C
```

5. DIAGNOSTICS-RANDOM ACCESS

SELF-LOADING RANDOM ACCESS EDIT

General Description

◆ The Self-Loading Random Access Edit routine is an emergency testing aid that provides an edited listing of selected areas of a random access device. The areas to be displayed, the listing format, and the input device type are determined from parameters entered by way of the console typewriter or from the card reader.

This routine contains its own bootstrap, loader, and device control. As a result, it is not dependent on any other programming system and is loaded without the Executive or Monitor.

Preset Functions

None.

Optional Functions

The input device can be specified as a mass storage unit, a drum unit or a disc storage unit. Output listings can be printed in graphic format, hexadecimal format, or a combination of both.

Input

◆ Input to this routine consists of the random access device to be edited (mass storage, disc, or drum unit) and input parameters that specify the editing functions desired.

All Random Access Devices (70/564, 70/565, 70/567, and 70/568) are handled indiscriminately. Alternate tracks are printed. If data is spread over two magazines on the 70/568, then an input parameter for each magazine must be used. Ⓐ

Ⓐ

Output

◆ Output of this routine is an edited listing of the random access areas defined by the input parameters.

When hexadecimal or hexadecimal/graphic format is specified, each print line contains 50 bytes of input data, arranged in 5 groups of 10 bytes each. For graphic format, 100 bytes of data are displayed per print line, in 5 groups of 20 bytes each.

The Count and Key fields of data records are always displayed in graphic format, with the Count field being first converted to a decimal number.

Track descriptor records are processed in the same manner as data records.

Equipment Configuration

Required

◆ Processor (65K)

Console typewriter

Card reader, or Videoscan document Reader with card read feature.

Random access device (Model 70/568, 70/564, or 70/565)

Printer

Optional

◆ A magnetic tape can be substituted for the printer as the output device.

Ⓐ A Model 70/567 Drum Memory Unit can be substituted for a Model 70/564, 70/565, or 70/568 Random Access Device.

Routine Parameters -

◆ Two parameters are used with this routine: the Format parameter and the End parameter.

The Format parameter (or parameters) defines the random access area to be edited and the format of the output listing. The End parameter denotes the end of parameter input and, as such, terminates this routine.

Routine Parameters - Detailed

Format Parameter

◆ This parameter defines the format of the output listings, the type of random access device, and the physical area of the device to be edited. Three formats are provided:

1. MASS STORAGE UNIT

Format:

Δf,cuu,R,nnncch,nnncch,mm

Entry	Meaning
Δf	Output listing format: f = H hexadecimal = G graphic = C hexadecimal and graphic
,cuu	Channel and unit number of input device.
,R	Input device is a mass storage unit.
,nnncch	First track to be edited: nnn = card number (000-255). cc = cylinder number (00-15). h = head number (0-7).
,nnncch	Last track to be edited: nnn = card number (000-255). cc = cylinder number (00-15). h = head number (0-7).
,mm	Magazine number (00-07).

Example:

Δ H,301,R,000045,002045,03

Format Parameter (Cont'd)

2. DRUM STORAGE UNIT

Format:

Δf,cuu,D,ccch,ccch

Entry	Meaning
Δf	Output listing format: f = H hexadecimal = G graphic = C hexadecimal and graphic
,cuu	Channel and unit number of input device.
,D	Input device is a drum storage unit.
,ccch	First track to be edited: ccc = cylinder number (000-255) h = head number (0-7)
,ccch	Last track to be edited: ccc = cylinder number (000-255) h = head number (0-7)

Example:

ΔH,204,D,0201,0221

3. DISC STORAGE UNIT

Format:

Δf,cuu,P,ccch,ccch

Entry	Meaning
Δf	Output listing format: f = H hexadecimal = G graphic = C hexadecimal and graphic
,cuu	Channel and unit number of input device.
,P	Input device is a disc storage unit.
,ccch	First track to be edited: ccc = cylinder number (000-202). h = head number (0-9).
,ccch	Last track to be edited: ccc = cylinder number (000-202). h = head number (0-9).

```
 PAGE: 0002                                           RANDOM ACCESS EDIT                                 007
                                         ①         ②    ③
 BLOCK  CHAR                           FORMAT  C INPUT DEVICE 140 LHE 0010   RHE 0011

④              ⑤
 00007            COUNT: (C1) 000  (C2) 001  (H1) 0  (H2) 1  (R) 003  (KL) 010  (DL) 00100
                  ⑥
                  KEY:V B U T W C V Z Q V
                      E5C2E4E3E6C3E5E9D8E5
                  ⑧
     ⑦            M V B U T W C V Z Q   V M M M M             M M M M M 1 9 0 4 4   1 2 2 5 5 M M M M M   M M M M M M M M M M
        00001     D4E5C2E4E3E6C3E5E9D8  E5D4D4D4D4000000012C  D4D4D4D4D4F1F9F0F4F4  F1F2F2F5F5D4D4D4D4D4  D4D4D4D4D4D4D4D4D4D4

                  M M M M M M M M M M   M M M M M M M M M M   M M M M M M M M M M   M M M M M M M M M M   M M M M M M M M M M
        00051     D4D4D4D4D4D4D4D4D4D4  D4D4D4D4D4D4D4D4D4D4  D4D4D4D4D4D4D4D4D4D4  D4D4D4D4D4D4D4D4D4D4  D4D4D4D4D4D4D4D4D4D4

 00008            COUNT: (C1) 000  (C2) 001  (H1) 0  (H2) 1  (R) 004  (KL) 010  (DL) 00100

                  KEY:E A S Z U H Q J B I
                      C5C1E2E9E4C8D8D1C2C9

                  M E A S Z U H Q J B   I M M M M             M M M M M 9 9 4 0 4   5 6 3 9 9 M M M M M   M M M M M M M M M M
        00001     D4C5C1E2E9E4C8D8D1C2  C9D4D4D4D4000000013C  D4D4D4D4D4F9F9F4F0F4  F5F6F3F9F9D4D4D4D4D4  D4D4D4D4D4D4D4D4D4D4

                  M M M M M M M M M M   M M M M M M M M M M   M M M M M M M M M M   M M M M M M M M M M   M M M M M M M M M M
        00051     D4D4D4D4D4D4D4D4D4D4  D4D4D4D4D4D4D4D4D4D4  D4D4D4D4D4D4D4D4D4D4  D4D4D4D4D4D4D4D4D4D4  D4D4D4D4D4D4D4D4D4D4
```

① Output listing format option specified in Parameter.

② Channel and Unit no. of input device (omitted if RAEDIT is used).

③ LHE and RHE cylinders and tracks requested (ccct).

④ Number of records printed.

⑤ Interpreted breakdown of count field:

(C1) Card no. if mass storage
(C2) Cylinder No.
(H1) Zero
(H2) Track No.
(R) Record No.
(KL) Length of Key Field
(DL) Length of Data Field

⑥ Key Field (if no Key, this line is omitted).

⑦ Byte Number of first byte of data field on this line.

⑧ Data field printed showing graphic and hexadecimal.

Note:

The Home Address is not printed. The Track Descriptor Record (RO) is treated as a data record. Any defective or alternate tracks are indicated as such on the listing. An EOF record is indicated by printing "END OF FILE."

Format Parameter (Cont'd)

Notes:

1. When entering these parameters from the console typewriter the leading space character is <u>not</u> used.
2. There is no restriction as to the number or type of parameters that can be submitted at run time.

Examples:

```
ΔH,201,P,2021,2021
```

END Parameter

◆ This parameter is mandatory and denotes the end of input parameters for the edit routine.

Format:

```
ΔEND
```

Note:

When this parameter is entered from the console typewriter, the leading space character is <u>not</u> used.

Operation:

After the routine is loaded, the following typeout occurs:

```
47PDA ENTER I/O PARAM DEVICE
```

To enter parameters from typewriter, reply with: T,ocuu and press EOT.

For card input replay with: C,ocuu and press EOT.

where: o = output listing device; P for printer or T for tape.
cuu = channel and device number of output device.

The routine proceeds to read edit parameters from the device specified until an END parameter is recognized.

Note:

When parameters are entered from the typewriter, the message ENTER INPUT PARAM is displayed for each input parameter. To terminate the routine the operator must respond with the message: END.

Device Assignments

◆ Not applicable.

Related Programming Systems

◆ If the output of this routine is transcribed to magnetic tape, the preedit option of the Tape Edit routine is used to print the output tape.

RANDOM ACCESS EDIT (RAEDIT)

General Description

◆ The Random Access Edit (RAEDIT) routine is a diagnostic aid that provides an edited listing of selected areas of a random access device. The areas to be displayed, the listing format, and the input device type are determined from parameters entered by way of the console typewriter or the card reader.

This routine provides the same functions as the self-loading version. The only difference between the two is the manner of loading the routine.

Preset Functions

None.

Optional Functions

The input device can be specified as a mass storage unit, a drum unit or a disc storage unit. Output listings can be printed in graphic format, hexadecimal format, or a combination of both.

Input

◆ Input to this routine consists of the random access device to be edited (mass storage, disc, or drum unit) and input parameters that specify the editing functions desired.

All of the Random Access Devices (70/564, 70/565, 70/567, and 70/568) are handled indiscriminately. Ⓐ

Output

◆ Output of this routine is an edited listing of the random access areas defined by the input parameters.

When hexadecimal or hexadecimal/graphic format is specified, each print line contains 50 bytes of input data, arranged in 5 groups of 10 bytes each. For graphic format, 100 bytes of data are displayed per print line, in 5 groups of 20 bytes each.

The Count and Key fields of data records are always displayed in graphic format, with the Count field being first converted to a decimal number.

Track descriptor records are processed in the same manner as data records.

Equipment Configuration

Required

◆ Processor (65K).

Console typewriter.

Card reader, or Videoscan document reader with card read reader.

Random access device (Model 70/568, 70/564, or 70/565).

Printer.

Optional

◆ A magnetic tape can be substituted for the printer as the output device.

The Model 70/567 Drum Memory Unit can be substituted for the Model 70/564, 70/565, or 70/568 Random Access Device. Ⓐ

Routine Parameters - General

◆ Two parameters are used with this routine: the Format parameter and the End parameter.

The format parameter (or parameters) defines the random access area to be edited and the format of the output listing. The End parameter denotes the end of parameter input and, as such, terminates this routine.

Routine Parameters - Detailed

Format Parameter

◆ This parameter defines the format of the output listings, the type of random access device, and the physical area of the device to be edited. Three formats are provided:

1. MASS STORAGE UNIT

Format:

Δf,R,nnncch,nnncch,mm,ssssss

Entry	Meaning
Δf	Output listing format: f = H hexadecimal = G graphic = C hexadecimal and graphic
,R	Input device is a mass storage unit.
,nnncch	First track to be edited: nnn = card number (000-255). cc = cylinder number (00-15). h = head number (0-7)
,nnncch	Last track to be edited: nnn = card number (000-255). cc = cylinder number (00-15). h = head number (0-7).
,mm	Magazine number (00-07).
,ssssss	Symbolic name of input device (six characters).

Example:

ΔH,R,000045,002045,03,SYS012

2. DRUM STORAGE UNIT

Format:

Δf,D,ccch,ccch,ssssss

Format Parameters (Cont'd)

Entry	Meaning
Δf	Output listing format: f = H hexadecimal = G graphic = C hexadecimal and graphic
,D	Input device is a drum storage unit.
,ccch	First track to be edited: ccc = cylinder number (000-255) h = head number (0-7)
,ccch	Last track to be edited: ccc = cylinder number (000-255) h = head number (0-7)
,ssssss	Symbolic name of input device (six characters).

Example:

ΔH,D,0201,0221,SYS015

3. DISC STORAGE UNIT

Format:

Δf,P,ccch,ccch,ssssss

Entry	Meaning
Δf	Output listing format: f = H hexadecimal = G graphic = C hexadecimal and graphic
,P	Input device is a disc storage unit.
,ccch	First track to be edited: ccc = cylinder number (000-202). h = head number (0-9).
,ccch	Last track to be edited: ccc = cylinder number (000-202). h = head number (0-9).
,sssssss	Symbolic name of input device (six characters).

Example:

ΔH,P,2021,2021,SYS022

Format Parameters (Cont'd)

Notes:

1. When entering these parameters from the console typewriter, the leading space character is not used.
2. There is no restriction as to the number or type of parameters that can be submitted at run time.

END Parameter

◆ This parameter is mandatory and denotes the end of input parameters for the edit routine.

Format:

ΔEND

Note:

When this parameter is entered from the console typewriter, the leading space character is not used.

Device Assignments

◆ Under Executive Control

SDN	Device Type	Remarks
RDPARM	Card reader	Parameter input.
RAOUT	Printer or magnetic tape	Output listings.
ssssss	Random access volume	Symbolic name of input device.

Under Monitor Control

SDN	Device Type	Remarks
SYSIPT	Card reader	Parameter input.
SYSLST	Printer or magnetic tape	Output listings.
ssssss	Random access volume	Symbolic name of input device.

Related Programming Systems

◆ If the output of this routine is transcribed to magnetic tape, the preedit option of the Tape Edit routine is used to print the output tape.

Output Examples

◆ The Random Access Edit output is the same as the Self- Loading Random Access Edit output except that the input device channel and unit number are not printed. Refer to page 5-4A.

70/568 MAGAZINE CARD CHECK (CARDCK)

General Description

◆ The 70/568 Magazine Card Check routine allows a rapid check of one or more 70/568 magazines for missing, duplicate, or unselectable cards.

Preset Functions

None.

Optional Functions

Any number of magazines can be analyzed.

Input

◆ Input to this routine consists of a 70/568 magazine and a parameter message entered by way of the console typewriter.

Output

◆ Output consists of console messages indicating missing, duplicate, or unselectable cards.

Equipment Configuration

Required

◆ Processor (65K).
Console typewriter.
70/~~568~~ Mass Storage Unit.

Routine Parameters - General

◆ Parameter data is supplied to this routine by means of the console typewriter. Initially, the following information is typed in:

1. The bin number (0-7) of the magazine to be analyzed.
2. The installation mnemonic assigned to the 70/568 on which the magazine to be checked is placed.

Routine Parameters - Detailed

◆ After the routine has been loaded, the following typeout occurs:

```
a CARDCK 2900A BIN NO ?
```

Respond:

```
aΔy
```

where: a = program priority number.
y = bin number (0-7) of the magazine to be analyzed.

The next typeout is a request for device assignment of the 70/568 unit.

Routine Parameters - Detailed *(Cont'd)*

The program then selects every card in the designated magazine (from 0 to 255). If the unit goes inoperable - caused by card absent, two-card select, or card not selectable - the following Executive typeout occurs:

X CARDCK 0101A mn INOPERABLE

where: mn = installation mnemonic assigned to the device.

Correct the inoperability of the 70/568 and return control to the program by responding:

XΔ1

The CARDCK routine will then produce the following typeout denoting the number of the card that caused the error:

a CARDCK 2901A mn b ccc

where:
a = program priority number.
mn = installation mnemonic.
b = bin number.
ccc = card number.

If device is inoperable at this typeout, correct the inoperability and depress EOT. Otherwise, merely depress EOT to continue the routine.

Upon completing the check of an entire magazine, the initial message will occur again:

a CARDCK 2900A BIN NO ?

If all desired magazines have been checked, respond:

aΔNO

This response will terminate the routine.

Considerations for Use

◆ 1. If 2901A typeout is due to a card absent condition, the missing card does not have to be replaced at this time because the program accesses the next card in sequence after the typewriter response is made.

2. The routine will analyze a magazine in approximately three minutes.

Device Assignments

◆ Under Executive or Monitor Control:

SDN	Device Type	Remarks
SYSMAG	Mass Storage Unit.	Symbolic name of device containing magazines to be checked.

6. SYSTEM MAINTENANCE

OBJECT MODULE LIBRARY UPDATE (OMLU)

General Description

◆ The Object Module Library Update (OMLU) is used to create, modify, copy, and/or display an Object Module Library (OML). To understand this routine the reader should be familiar with the following terms:

Object Module File (OMF)

An object module, in card image format, generated by one of the TOS language translators, such as the Assembler or COBOL compiler.

Object Module Library (OML)

A library of object modules (sequenced by module name) which have been reformatted by the OMLU routine.

Call Library

A composite library tape that contains the various libraries used in TOS. Library sections (when present) appear on this tape in the following order:

ASSEMBLY MACRO LIBRARY
COBOL LIBRARY
OBJECT MODULE LIBRARY
EXECUTIVE LIBRARY

The OMLU routine performs those functions necessary to create and maintain the Object Module Library section of a Call Library tape. In addition, this routine may also be used to display modules stored in an object module library.

All functions performed by the OMLU are completely dependent upon the type of parameters supplied by the programmer.

Preset Functions

None.

General Description *(Cont'd)*

Optional Functions

This routine provides the following options:

1. Merging object module libraries that appear on two or more library tapes.
2. Deleting object modules from library tapes.
3. Extracting object modules from library tapes.
4. Cataloging (adding) object module files produced by the language translators for inclusion in an OML.
5. Renaming object modules.
6. Displaying the coding for all or selected modules appearing in an OML.
7. Displaying an OML index. This index lists the names of modules contained in the library and provides information pertinent to each module such as its length and the EXTRN's and ENTRY's defined for that module.
8. Creating an initial Object Module library section for a Call Library Tape or updating the OML section of an existing Call Library Tape.

Input

◆ Input to this routine can consist of the following:

Routine parameters.

Object module files to be added to the OML.

Object Module Library to be merged onto the output tape.

A Call Library Tape.

See table 6-1, page 6-13 to determine when each type of input is required.

Output

◆ Output of this routine is a Call Library Tape that contains an OML section only or one that contains multiple library sections.

A program listing and module index can also be generated for all, or selected, modules in the object module library section.

Equipment Configuration

Required

◆ Processor (65K).

Console typewriter.

Magnetic tape devices (three required).

Card reader, or Videoscan document reader with card read feature.

Printer.

Optional

◆ Additional magnetic tape devices may be used as input files.

Magnetic tapes may be substituted for the card reader and for the printer.

Seven-level tapes with pack/unpack may be substituted for nine-level tapes.

Routine Parameters - General

◆ Except for the END parameter, all routine parameters are optional and need only be supplied when various functions are desired.

COPY Parameter

When used, this parameter copies selected libraries from an input Call Library Tape to the output tape. If omitted, this routine copies all library sections appearing on the input tape to the output tape.

MERGE Parameter

This parameter can be used to merge object modules from up to three input Object Module Libraries onto the output tape.

RENAME Parameter

This parameter can be used to change the name of a merge input module when that module is written to the output tape.

DELETO Parameter

When used, this parameter specifies that an object module appearing on a merge input tape is not to be included on the output tape.

EXTRACT Parameter

This parameter can specify that a particular object module from an input library tape is to be included in the output tape.

CATALO Parameter

This parameter can specify that an object module file is to be converted to OML format and added to the Object Module Library.

Routine Parameters - General *(Cont'd)*

DSPLY Parameter

This parameter can specify that an object module (or all object modules) is to be printed.

DSPLI Parameter

This parameter can specify that an object module index (or all object module indices) is to be printed.

END Parameter

This parameter indicates the end of input parameters.

Sequence of Parameters

◆ The following sequence of parameters must be maintained regardless of which options are selected.

Library Maintenance	Library Listings
COPY	DSPLY [I]
MERGE	END
RENAME	
DELETO	
EXTRACT	
CATALO	
DSPLY [I]	
END	

Multiple RENAME, DELETO, EXTRACT, CATALO, and DSPLY parameters may be used. It is not necessary that they be in module name sequence as long as all parameters of the same type appear together.

EXTRACT's should be grouped by device name and ordered in module name sequence for each device.

Object module files to be added to the library tape should appear on the input device (or devices) in alphanumeric sequence. CATALO parameters should be submitted by OMF name as they appear in the OMF.

NOMAP Parameter

◆ *Format:*

ΛNOMAP

This parameter specifies that a program map listing is not desired.

XREF Parameter

◆ *Format:*

ΛXREF

This parameter specifies that a cross-reference listing of the external symbols of a bound program is desired.

NOCTL Parameter

◆ *Format:*

ΛNOCTL

This parameter specifies that an automatic overlay control module is not to be produced. (The bound program must use LPOV to load overlays if VCON's have been used.)

NCAL Parameter

◆ *Format:*

ΛNCAL

This parameter specifies that the automatic library call is to be inhibited. If used, unresolved external references will not be satisfied, and error messages associated with these references will be suppressed.

This parameter is ineffective if it is used with the LIBRARY parameter.

LET Parameter

◆ *Format:*

ΛLET

This parameter specifies that a program is to be bound even though certain errors are detected that may prevent its execution. It allows unsatisfied EXTRNS to be bound. Ⓒ

Routine Parameters - Detailed

COPY Parameter

◆ This parameter is used to create an initial Call Library Tape containing an OML section only, or to designate that only selected library sections of an input Call Library Tape be copied to the output tape.

Format:

```
ΛCOPYΛbbbbbb,AL,EL,CL
     or
ΛCOPYΛNONE
```

Entry	Meaning
ΔCOPYΔ	Parameter identifier.
bbbbbb or NONE	bbbbbb = device name on which input Call Library appears. NONE = no input Call Library; generate CLT with an OML section only.
,AL	Copy Assembly Macro Library from input tape. (If this library is not desired on the output tape, the entry is omitted.)
,EL	Copy Executive library from input tape. (If this library is not desired on the output tape, this entry is omitted.)
,CL	Copy COBOL library from input tape. (If this library is not desired on the output tape, this entry is omitted.)

Notes:

1. When used, the COPY parameter must be the first parameter submitted.

2. If this parameter is not supplied, and a Call Library Tape is mounted on SYSLIB, all library sections on the Call Library Tape will be copied to the output tape.

Examples:

```
ΛCOPYΛNONE

ΛCOPYΛSYSLIB,EL

ΛCOPYΛSYSUT1,AL,EL
```

Device Assignments

◆ Under Monitor Control

Name	Type	Description
SYSIPT	Card Reader*	Device from which job stream is read.
SYSLIB	Magnetic Tape	Call Library containing a copy of the Executive system and an OML.
SYSLST	Printer*	Output device to which error messages and output listings are written.
SYSUT1	Magnetic Tape	Only required when no parameters are supplied to the Linkage Editor from SYSIPT. In this case all primary input must appear on SYSUT1. This device may also be used as a secondary input.
SYSUT2	Magnetic Tape	Device to which the output PLLT or SLLT is written.
SYSUT3	Magnetic Tape	Work tape for Linkage Editor processing.

*Magnetic Tape may be substituted.

Considerations for Use

◆ Considerations for use of the Linkage Editor should start during the design and coding phase of program preparation if there are to be multiple segments and overlays. Although the Linkage Editor provides the facility to construct overlay programs, the programmer should plan for the linkage and use of overlays during the writing of his program. TOS provides three assembly macros by which a program may call in overlays: LPOV, CALL, and SEGLD. In addition, a V-type constant can be used in loading an overlay at object execution time.

Execution of the LPOV macro causes the segment specified to be brought into memory. The coding of this macro also specifies the instruction to which control is transferred upon completion of the loading. When the LPOV macro is used, the specified segment is loaded regardless of whether it is already in memory. This macro is further described in the FCP and Executive Communication Macros Reference Manual, 70-00-608.

With the use of the CALL macro, overlay loading is performed by an Overlay Control Module which is generated by the Linkage Editor. With this macro, an overlay is not reloaded if it is already in memory. Upon completion of the loading process, execution of the program is continued with the instruction referenced by the operand in the CALL macro.

Considerations for Use *(Cont'd)*

The SEGLD macro also causes the Linkage Editor to generate an Overlay Control Module which performs the loading of the overlay. With this method, an overlay is reloaded regardless of whether or not it is already in memory. Upon completion of the loading process, control returns to the instruction following the macro or, optionally, to an instruction specified by the second operand of the SEGLD macro.

In addition to the above macros, the programmer may use V-type constants to effect overlay loading, which also causes the Linkage Editor to generate an Overlay Control Module. This method consists of loading Base Register 15 with a V-type constant whose symbol appears as an operand in the segment which is to be loaded.

In the absence of the NOCTL statement, the Linkage Editor assumes that automatic overlay loading (any method other than LPOV) is required. The Overlay Control Module maintains a record of the status of the program's overlay structure. If the LPOV macro is used, it communicates directly with the Executive and the overlay status record maintained by the Overlay Control Module may be invalid.

The Linkage Editor cannot detect an illogical use of these overlay methods.

The Linkage Editor allows any number of duplicate entries. Just which entry satisfies extrn depends upon the order or path taken to satisfy extrns. Ⓐ

Example

◆ Following is the control card and object module arrangement for a Linkage Editor run to create a System Load Library tape containing three explicitly-bound programs. Cross-reference and memory map listings are desired for each program.

The second program has a single region overlay. This will be called at object execution time by LPOV, thus eliminating the need for the Overlay Control Module to be bound by the Linkage Editor. It has a specific external reference, EXAAA, which is located on SYSUT1. Other external references are on SYSLIB, which is automatically searched.

```
ΔACTIONΔSYSLD
ΔPROGΔTEST1
[ TEST1 object module deck ]
ΔXREF
ΔENTRY

ΔPROGΔTEST2
[TEST2 object module deck ]
ΔOVERLAYΔNODEAD,TEST2A
[Object module deck for TEST2 overlay],
ΔLIBRARYΔSYSUT1 (EXAAA)
ΔXREF
```

Example
(Cont'd)

```
ΔNOCTL
ΔENTRY
ΔPROGΔTEST3
[ TEST3 object module deck ]
ΔXREF
ΔENTRY
```

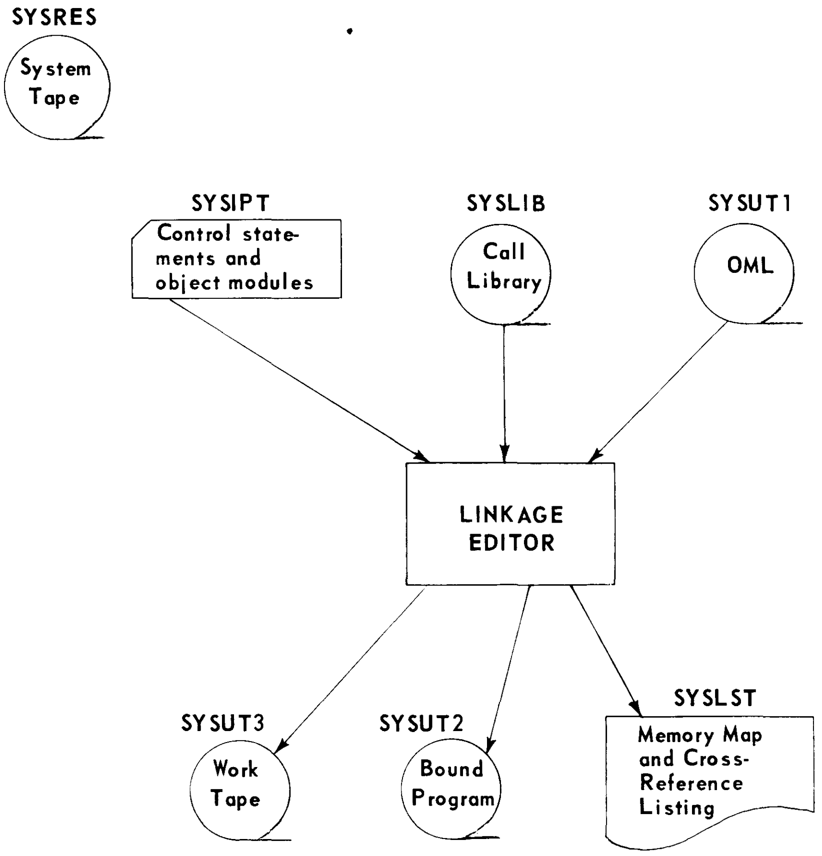


Figure 6-1. Example of Device Utilization

TAPE FILE MAINTENANCE (TPMAIN)

General Description

◆ The Tape File Maintenance routine maintains card or tape input files by adding, deleting, and changing data as specified by input parameters. Selected portions of input files may be copied to the output or deleted. Multiple input and output devices are also accommodated.

Preset Functions

None.

Optional Functions

The optional functions listed below are available with the TPMAIN routine.

1. Position input and output tapes before beginning file maintenance.
2. Copy selected portions of input files.
3. Handle multiple input and output devices.
4. Check, correct, or generate sequence numbers.
5. Change, add, or delete input data.
6. Make a list of all labels on a file.
7. Cause the routine to recognize checkpoint blocks on an input tape.

Input

◆ Input to TPMAIN is a group of parameters and one or more files contained on magnetic tape, paper tape, or the card reader. A mix of up to five input devices can be assigned for one routine session.

Output

◆ The output consists of all or portions of input files that have been changed as specified by the routine parameters. The output can be to magnetic tape, paper tape, or the card punch. Also, listings of various types can be prepared on the printer. Any number of output devices can be assigned for one routine session but only three devices can be used per function.

Equipment Configuration

Required

◆ Processor (65K).

Console typewriter.

Card reader, or Videoscan document reader with card read feature.

Magnetic tape device.

Optional

◆ Additional input and ouput devices.

Input Parameters - General

◆ All parameters are optional although some must be used only in conjunction with other parameters. At least one parameter must be submitted to perform any of the TPMAIN functions.

Checkpoint Parameter

This parameter causes rerun and restart records to be processed as legitimate blocks on tape. If this parameter is absent, checkpoint records are ignored except by the Position parameter, which always counts checkpoint records.

Position Parameter

Up to five input devices are positioned by this parameter. Magnetic tape devices may be moved in a forward or reverse direction a specified number of blocks or tape marks. Paper tape and card files are only positioned in the forward direction. When processing paper tape or card files, a /* in the first two positions of a block is recognized as a tape mark. Checkpoint records are counted as blocks by the Position parameter.

COPY Parameter

The Copy parameter specifies the input and output devices, their formats, and positioning (if they are magnetic tapes) before and after the copy operation. The parameter also defines the starting and ending points of the data to be copied. The copy begins with the record defined by the starting point and terminates after the record defined by the end point has been copied. The output may be made to three devices simultaneously.

This parameter also is used to obtain a listing of the VOL, HDR, and EOF labels contained on a file. When this option is selected, the output device must be TFMLST.

Sequence Parameter

This parameter is used with the Copy parameter to indicate that the copy function is to be based on sequence numbers. Operations selected by this parameter are (1) sequence checking of input, (2) generating new sequence numbers on the output, and (3) updating the input using sequence numbers. The point where processing starts is the record defined by the starting point in the Copy parameter.

When a sequence check is specified, all records from the start point up to and including the end point are checked and copied to the output. If a sequence error is found, it may be listed on the printer or the routine can be made to halt and cycle with an error message.

Sequence generation allows sequence fields to be inserted or changed. The first sequence number and the increment value for succeeding records can also be specified.

When updating is selected, input records can be replaced and new records inserted. Updating can only be performed when the input consists of fixed, 80-byte records. The update records must have the same format as the input. An update record replaces an input record when it has the same sequence number. When an update record sequence number does not match with an input record it is inserted in the file.

Input Parameters - General *(Cont'd)*

REMOVE Parameter

The Remove parameter specifies records that are to be deleted according to sequence numbers. One record or a series of records can be deleted by each parameter.

End of Update Parameter

This parameter indicates the end of update data associated with the Sequence parameter.

Replace Parameter

The Replace parameter is used with the Copy function to replace data within a record. This parameter also may be used to verify data in a record. The record that is changed is defined by the end point in the Copy parameter.

Delete Parameter

The Delete parameter specifies the area of an input file to be deleted. The parameter also defines the input and output devices, their formats, and positioning of input and output magnetic tape devices before and after the delete operation.

The input is copied from its present position up to, but not including, the record defined by the starting point in the parameter. The input is then positioned immediately after the record defined by the end point and another parameter is read. If there are no more parameters, the input is copied up to a double tape mark.

END Parameter

The End parameter terminates the TPMAIN routine.

Input Parameters - Detailed

Checkpoint Parameter

◆ *Format:*

ΔCHKPT

Note:

If the Checkpoint parameter is not used, checkpoint records are ignored by all parameters except Position.

Position Parameter

◆ *Format:*

$$\Delta\text{POSI}\Delta\text{nnndcpp,}\ \left\{\begin{array}{l}\text{THEN}\\ \text{STOP}\end{array}\right\}$$

Entry	Meaning
ΔPOSIΔ	Parameter identifier.
nnndcpp	nnn = last three characters of symbolic name for device to be positioned. d = F position device in forward direction. = R position device in reverse direction. (Magnetic tape only.) c = B position by block count. = T position by tape marks. p = number of blocks to be positioned (00001-99999). = number of tape marks to be positioned (01-99). <u>Note:</u> Up to four additional positioning operands can be included in the parameter. Each must be preceded by a comma.
,THEN or ,STOP	THEN = position device and read next parameter. STOP = position device and terminate routine.

COPY Parameter

◆ *Format:*

$$\Delta\text{COPY}\Delta\text{IDnnn/ba,}\ \left\{\begin{array}{l}\text{IFf/t/s}\\ \text{IFLABELS/u}\end{array}\right\}\ \text{,SP}\left\{\begin{array}{l}\text{BT}\\ \text{TMg}\\ \text{HLh/i}\\ \text{TLj/k}\\ \text{KYm/n/p/q}\end{array}\right\}\text{+r,}$$

$$\text{EP}\left\{\begin{array}{l}\text{BT}\\ \text{TMg}\\ \text{HLh/i}\\ \text{TLj/k}\\ \text{KYm/n/p/q}\\ \text{ET}\end{array}\right\}\text{+r,ODnnn/}\left\{\begin{array}{l}\text{cd}\\ \text{e}\end{array}\right\}\text{,OFf/t/s,}\ \left\{\begin{array}{l}\text{AND}\\ \text{THEN}\\ \text{STOP}\end{array}\right\}$$

COPY Parameter (Cont'd)

Entry	Meaning
ΔCOPYΔ	Parameter identifier.
IDnnn/ba	ID = Input device operand identifier. nnn = last three characters of symbolic name of input device. For magnetic tape input device: b = N do not rewind before beginning copy. = R rewind before beginning copy. A magnetic tape input device is always rewound the first time it is accessed. a = N do not rewind after copy. = R rewind after copy. = D rewind and disconnect after copy. When ba is not used, there is no positioning before or after the copy. If this operand is not used, the input device and format from the preceding Copy or Delete parameter are assumed.
,IFf/t/s or ,IFLABELS/u	IF = operand identifier. Input format: f = F fixed-length records. = V variable-length records. For fixed-length records (f-F). t = record length (00001-99999). For variable-length records (f-V). t = S undefined records. = B blocked records. s = maximum input block size (00012-99999). (See note.) IFLABELS = special operand identifier used to obtain listing of labels (VOL, HDR, EOF) on input device. When this is used, the output device and format must be, ODLST/C,OFF/00080/00080 (refer to page 6-69). u = maximum input block size (00012-99999). This operand must be used with the ID operand and cannot be used by itself.

COPY Parameter
(Cont'd)

Entry	Meaning
,SP $\left\{\begin{matrix}\text{BT}\\ \text{TMg}\\ \text{HLh/i}\\ \text{TLj/k}\\ \text{KYm/n/p/q}\end{matrix}\right\}$ +r	This operand defines the starting point for the copy function. SP = starting point operand identifier. BT = starting point is first record after VOL label, if a VOL label is used, or the first record on tape if a VOL label is not used. TM = starting point is a number of tape marks forward from the present input position. g = number of tape marks (01-99). HL = starting point is a specific header label. h = number of bytes to be compared in file identifier field of header label (01-17). i = the value to compare against the file identifier. The comparison begins with the left-hand end of the field. TL = starting point is a specific trailer label. j = number of bytes to be compared in file identifier field of trailer label (01-17). k = the value to compare against the file identifier. The comparison begins with the left-hand end of the field. KY = starting point is a record containing a specified key field. m = position of first byte of key field in the record (001-999). n = length of the field in bytes. (01-17 for EBCDIC, 01-08 for hexadecimal.) p = X hexadecimal field. C EBCDIC graphics field. q = the value to compare against the key field. If the field is EBCDIC, the length of the value must equal n. If the field is hexadecimal, the length of the value must equal 2n. +r = number of records to position forward in addition to previous starting point criteria (1-99999).

(Continued)

COPY Parameter (Cont'd)

Entry	Meaning
$,SP\left\{\begin{matrix}BT\\TMg\\HLh/i\\TLj/k\\KYm/n/p/q\end{matrix}\right\}+r$ (Cont'd)	The BT, TM, HL, TL, and KY fields are optional. If one of these is not used then the +r field must be used. The use of +r is optional with BT, TM, HL, TL, and KY. When the SP operand is not used the starting point is the present position of the input.
$,EP\left\{\begin{matrix}BT\\TMg\\HLh/i\\TLj/k\\KYm/n/p/q\\ET\end{matrix}\right\}+r$	This operand defines the end point for the Copy function. EP = end point operand identifier. BT,TM,HL,TL,KY, and +r have the same meaning as in the starting point (SP) operand except that they refer to the end point of the Copy function. ET = end point is a double tape mark. When the EP operand is not used the end point is a double tape mark.
$,ODnnn/\left\{\begin{matrix}cd\\e\end{matrix}\right\}$	OD = output device operand identifier. nnn = last three characters of symbolic name of output device. For magnetic tape output device: c = N do not rewind before copy. = R rewind before copy. d = N do not rewind after copy. = R rewind after copy. = D rewind and disconnect after copy. When cd is not used, there is no positioning before or after the copy. Format for printer: e = X hexadecimal. = C EBCDIC graphics. = not used, hexadecimal with graphic equivalents. If this operand is not used, the output device and format from the preceding Copy or Delete parameter are assumed.

MERGE Parameter

◆ This parameter must be supplied whenever this routine is used to update an existing OML section of a Call Library Tape. It is also required if modules are to be merged from two or more input Call Library Tapes to create an output library tape.

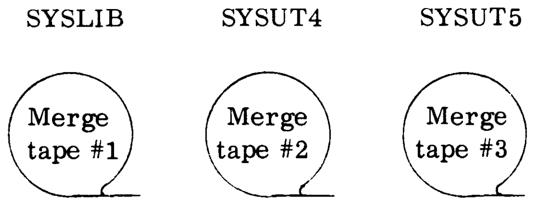
Format:

ΔMERGEΔn

Entry	Meaning
ΔMERGEΔ	Parameter identifier.
n	n = 1 one file to be copied onto output tape. = 2 two files to be merged onto output tape. = 3 three files to be merged onto output tape.

Notes:

1. When the merge function is used, library tapes must be mounted on the following devices:



2. If modules of the same name appear on different library tapes, only one version of the module is copied to the output tape. The module selected is determined by assigning SYSUT5 as the highest priority library tape, SYSUT4 as the next, and SYSLIB as the lowest. Thus, if module XYZ appears on SYSLIB and SYSUT4, only the version on SYSUT4 is placed on the output tape.

3. Only one MERGE parameter is permitted. A duplicate copy of an input Call Library Tape (SYSLIB) can be obtained by providing the following parameters:

ΔMERGEΔ1

ΔEND

Examples:

ΔMERGEΔ1

ΔMERGEΔ3

COPY Parameter (Cont'd)

Entry	Meaning
,OFf/t/s	OF = output format operand identifier. f = F fixed-length records. = V variable-length records. For fixed-length records (f=F). t = record length (00001-99999). For variable-length records (f=V). t = S undefined records. = B blocked records. s = maximum output block size (00012-99999). This operand is only used with the OD operand. Up to three pairs of OD and OF operands can be submitted with each Copy parameter.
,AND or ,THEN or ,STOP	AND = read next parameter before performing operations specified by this or previous unexecuted parameters. A maximum of 15 parameters may be linked by AND operands. THEN = perform operations specified by this and previous unexecuted parameters before reading next parameter. STOP = perform operations specified by this and previous unexecuted parameters. Terminate the TPMAIN routine.

Examples:

```
ΔCOPYΔID001/NN,IFF/00100/00500,SPBT,EPBT+5000,OD002/RN,
OFF/00100/00300,AND
```

```
ΔCOPYΔSPHL05/FILEA,EPTM01,ODLST/X,OFV/B/00150,THEN
```

```
ΔCOPYΔID004/RD,IFLABELS/00400,ODLST/C,OFF/00080/00080,STOP
```

Note:

The maximum input block size in the first Copy (or Delete) parameter in a TPMAIN session must be that of the largest input block size that will be encountered during the entire session. It does not have to be the true block size for the particular Copy or Delete function.

Sequence Parameter

◆ *Format:*

ΔSEQΔCK,FLnn,FPnnn,PER,PAL,HLT,GEN,FLnn,FPnnn,FBv,

$$\text{FIi,UPD,FLnn,FPnnn,PC,} \left\{ \begin{array}{l} \text{AND} \\ \text{THEN} \\ \text{STOP} \end{array} \right\}$$

Entry	Meaning
ΔSEQΔ	Parameter identifier.
CK	Check sequence numbers of input records. The check begins at the starting point given in the Copy parameter.
,FLnn	FL = field length operand identifier. nn = number of bytes in sequence field (01-17). If not used, a field length of five is assumed.
,FPnnn	FP = field position operand identifier. nnn = position of first byte of sequence field in the record (001-999). If not used, a byte position of 76 is assumed.
,PER	List only out of sequence records on the printer. Only the first 80 bytes of the record are printed.
,PAL	List all input records on the printer. Out of sequence records are flagged. Only the first 80 bytes of the record are printed.
,HLT	TPMAIN is to halt and cycle with error 3406A when an out of sequence record is found. HLT is assumed if neither PER nor PAL are used.
,GEN	Generate new sequence numbers for the output records. The new numbers begin on the record defined by the starting point in the Copy parameter.
,FLnn	FL = field length operand identifier. nn = number of bytes in sequence field to be generated (01-17). If not used, a field length of five is assumed.
,FPnnn	FP = field position operand identifier. nnn = position of first byte of sequence field to be generated in the record (001-999). If not used, a byte position of 76 is assumed.

Sequence Parameter (Cont'd)

Entry	Meaning
,FBv	FB = field value operand identifier. v = value of first sequence number to be generated. Its length must be the same as that given in the FL operand. If not used, the first generated sequence number is all zeros.
,FIi	FI = sequence increment operand identifier. i = increment value. It can be from 1 to 17 bytes in length but not longer than the length given in the FL operand. If not used, a value of 10 is assumed.
,UPD	Update records based on sequence numbers. The records to be used to update the file follow the parameter containing the operand THEN or STOP. Note: Refer to End of Update parameter, page 6-73.
,FLnn	FL = field length operand identifier. nn = number of bytes in sequence field of update records (01-17). If not used, a length of five is assumed.
,FPnn	FP = field position operand identifier. nn = position of first byte of sequence field in update record (01-80). If not used, a byte position of 76 is assumed.
,PC	List changes made to the file on the printer.
,AND or ,THEN or ,STOP	Same as for the Copy parameter.

Examples:

```
ΔSEQΔCK,PAL,GEN,FB00100,FI20,THEN

ΔSEQΔCK,FL12,FP043,UPD,FL12,FP43,PC,AND

ΔSEQΔGEN,FL09,FP097,FI5,STOP
```

REMOVE Parameter

◆ *Format:*

ΔREMOVE,a,b

Entry	Meaning
ΔREMOVE	Parameter identifier.
,a	The sequence number of the first record to be deleted. It must be the same length as specified in the FL operand of the Sequence parameter.
,b	The sequence number of the last record to be deleted. It must be the same length as specified in the FL operand of the Sequence parameter. If not used, only the record containing the sequence number given in a is deleted.

Examples:

ΔREMOVE,00150

ΔREMOVE,1070,2013

End of Update Parameter

◆ *Format:*

ΔENDUPD

Note:

This parameter must follow the last update card used with the update operand of the Sequence parameter.

Replace Parameter

◆ *Format:*

$$\Delta\text{REP}\Delta\text{ATbbbb/nn/f,NOWt,WASd,}\left\{\begin{array}{c}\text{AND}\\\text{THEN}\\\text{STOP}\end{array}\right\}$$

Entry	**Meaning**
ΔREPΔ	Parameter identifier.
ATbbbb/nn/f	AT = data position operand identifier. bbbb = position of first byte of data to be replaced (0001-9999). n = number of bytes to be replaced: 01-24 for EBCDIC graphic data. 01-12 for hexadecimal data. f = C data to be replaced is EBCDIC. = X data to be replaced is hexadecimal.
,NOWt	NOW = replacing text operand identifier. t = text which is to replace existing data in record. If the data is EBCDIC, the text length must equal n. If the data is hexadecimal, the text length must equal 2n.
,WASd	WAS = replaced data operand identifier. d = value of data which is to be replaced. Its length must be the same as the data that is replacing it. This value is compared to the data to be replaced and an error results if they are not equal. If not used, no comparison is made.
,AND or ,THEN or ,STOP	Same as for the Copy parameter.

Examples:

ΔREPΔAT0076/07/C,NOW24AD767,THEN

ΔREPΔAT0142/03/X,NOW7CAAE2,WAS7BCA23,AND

Delete Parameter

◆ This parameter defines an area of a file that is to be deleted.

Format:

$$\Delta\text{DEL}\Delta\text{IDnnn/ba,IFf/t/s,SP}\left\{\begin{matrix}\text{BT}\\ \text{TMg}\\ \text{HLh/i}\\ \text{TLj/k}\\ \text{KYm/n/p/q}\end{matrix}\right\}\text{+r,}$$

$$\text{EP}\left\{\begin{matrix}\text{BT}\\ \text{TMg}\\ \text{HLh/i}\\ \text{TLj/k}\\ \text{KYm/n/p/q}\\ \text{ET}\end{matrix}\right\}\text{+r,ODnnn/}\left\{\begin{matrix}\text{cd}\\ \text{e}\end{matrix}\right\}\text{,OFf/t/s,}\left\{\begin{matrix}\text{AND}\\ \text{THEN}\\ \text{STOP}\end{matrix}\right\}$$

The meanings for the operands in the Delete parameter are the same as in the Copy parameter. Sequence and Replace parameters cannot be used with the Delete function.

END Parameter

◆ *Format:*

ΔEND

Note:

This parameter must be used to terminate TPMAIN if a STOP operand has not been used in the last parameter.

Device Assignments

◆ Under Executive Control:

SDN	**Device Type**	**Remarks**
TFMRDR	Card reader.	Parameter input and file input if desired.
TFMLST	Printer.	Output device.
TFMOPT	Card punch.	Output device.
TFMnnn	Any.	Additional input and output devices.

Under Monitor Control:

SDN	**Device Type**	**Remarks**
SYSIPT	Card reader.	Parameter input. Can also be used for file input.
SYSLST	Printer.	Output device.
SYSOPT	Card punch.	Output device.
TFMnnn	Any.	Additional input and output devices.

Considerations for Use

General

◆

1. Input tape labels and data must have been recorded in the same mode.
2. The TPMAIN routine does not handle multivolume files.
3. Trailer label block counts are not updated by TPMAIN.
4. No updating record can have 'REMOVE,' in the first seven characters.
5. The first input parameter must be CHKPT, COPY, POSI, or DEL. If CHKPT is used, it must precede all other parameters.
6. Paper tape and card files must end with double tape marks (a /* in positions 1 and 2 of two consecutive records).

Labels

◆

1. All labels except VOL are treated as data records by TPMAIN.
2. VOL labels are not copied to output tapes or created on output tapes. VOL labels that are present on output tapes remain there unchanged. Purge dates are checked when a rewind of the output is specified.
3. When a file is being copied that contains data blocks with a maximum size that is smaller than the labels and the labels are desired in the output, the output block size must be defined as label size.
4. If output tapes are to be used with standard label processing, the programmer must update the labels accordingly.

Input and Output Devices

◆

1. The first time a magnetic tape input device is accessed it is rewound and positioned at the first record after the VOL labels. If there are no VOL labels, the tape is positioned to the first record on tape.
2. One input device and up to three output devices can be assigned for each set of parameters. (A set is a series of parameters tied together by AND operands.)
3. Only one starting point (SP) and ending point (EP) can be used in each set of parameters.
4. Magnetic tape outputs are closed with double tape marks only if they are defined in a parameter set that ends with a STOP operand. It is the programmer's responsibility to close all other tapes.

Parameter Examples *(Cont'd)*

5. Change byte number 520 in the 379th block on a tape that contains standard labels. TPMAIN will count the HDR label and following tape mark as additional blocks, therefore, the change will be made to the 381st block.

```
// ASSGN TFMRDR, R1
// ASSGN TFM001, 01
// ASSGN TFM002, 02
// END
ΔCOPY ID001/RN,IFV/S/01100,EPBT+381,OD002/RN,
 OFV/S/01100,AND
ΔREP AT0520/01/X,NOWOB,WASOC,THEN
ΔCOPY ID001/ND,IFV/S/01100,0D002/ND,OFV/S/01100,
 STOP
```

Parameter Examples

◆ 1. Copy a tape containing different or unknown formats with a maximum block size of 5000 bytes.

```
// ASSGN TFMRDR, R1
// ASSGN TFM001, 01
// ASSGN TFM002, 02
// END
ΔCOPY ID001/RD,IFV/S/05000,OD002/RD,OFV/S/05000,
 STOP
```

2. Copy a tape with standard labels containing variable, blocked data. The data is to be reblocked from 1000 to 1800 bytes. Separate COPY parameters must be used for the HDR and EOF records since these are fixed length.

```
// ASSGN TFMRDR, R1
// ASSGN TFM001, 01
// ASSGN TFM002, 02
// END
ΔCOPY ID001/RN,IFF/00080/01000,EPTM01,OD002/RN,
 OFF/00080/00080,THEN
ΔCOPY ID001/NN,IFV/B/01000,EPTM01,OD002/NN,
 OFV/B/01800,THEN
ΔCOPY ID001/ND,IFF/00080/00080,0D002/ND,
 OFF/00080/00080,STOP
```

3. Create a multifile volume from three single-file volumes. All tapes are standard format with labels.

```
// ASSGN TFMRDR,R1
// ASSGN TFMIN1,01
// ASSGN TFMIN2,02
// ASSGN TFMIN3,03
// ASSGN TFMOUT,04
// END
ΔCOPY IDIN1/RD,IFV/S/03500,EPTM03,ODOUT/RN,
 OFV/S/03500,THEN
ΔCOPY IDIN2/RD,IFV/S/03500,EPTM03,ODOUT/NN,
 OFV/S/03500,THEN
ΔCOPY IDIN3/RD,IFV/S/03500,ODOUT/ND,OFV/S/03500,
 STOP
```

4. Delete a block of records from a data file. The records begin with the second tape mark and end with the fourth tape mark. The second tape mark is to remain in the file.

```
// ASSGN TFMRDR, R1
// ASSGN TFM001, 01
// ASSGN TFM002, 02
// END
ΔDEL ID001/RD, IFV/S/01000,SPTM02 + 1, EPTM04,
 OD002/RD, OFV/S/01000,STOP
```

COBOL LIBRARY UPDATE (CLU)

General Description

◆ The COBOL Library Update (CLU) routine is used to create or update a COBOL Source Library, and copy and/or delete other libraries that are contained on a TOS Call Library Tape. CLU produces a new COBOL Library by merging, extracting, and/or deleting COBOL entries from a maximum of three Call Library Tapes.

Each entry contained in the STANDARD COBOL Library consists of a name followed by one or more COBOL source statements. Entries may be defined for each of the four sections in the library. These sections, any one of which may or may not be present, appear in the following order:

ENVIRONMENT DIVISION

DATA DIVISION

PROCEDURE DIVISION

SOURCE PROGRAMS

Entries within each section are arranged in alphanumeric sequence by entry name. See Appendix B for a description of the format of the COBOL Library Section.

The USASI COBOL Library differs from the STANDARD only because it is comprised of one section that contains all of the divisional and program entries.

Preset Functions

None.

Optional Functions

The following optional functions are provided:

1. To initially create a COBOL Source Library from card or magnetic tape input.
2. To add, delete, or replace entire entries within an existing COBOL Source Library.
3. To add or delete source statements within a specific entry of an existing COBOL Source Library.
4. To provide merge and/or extract capabilities during the update processing of existing CALL Library Input Tapes.
5. To print and/or punch selected entries within a COBOL Source Library.
6. To print a listing of all input statements received through the job input stream.
7. To print a directory or abstract listing of all entries within a COBOL Source Library.

Input

◆ There are three types of input to the CLU routine:

COBOL Source Statements

COBOL source statements must be written in the format described in the TOS/TDOS COBOL Reference Manual (70-00-607). All source statements must be 80-byte cards, or card images on paper tape or magnetic tape.

Source statements must be ordered by sequence number, if any, within an entry name. Sequence numbers for each statement are reassigned by the routine beginning with 000010, with each succeeding line incremented by 10. (If desired, original source sequence numbers may be retained by the use of the SAVE option in the CATALS parameter card.)

Call Library Tape(s)

An input COBOL Library resides on a Call Library.

Routine Parameters

All routine parameters are sequenced in ascending order by section and entry name within the section. All parameters are optional with the exception of the END parameter.

Output

◆ With the exception of the INPUT STATEMENT LISTING which is always produced, CLU does not produce a standard output. The output may be any one or a combination of the following:

1. Input Statement Listing.
2. Call Library Output Tape (Optional).
3. A COBOL Entry Listing (Optional).
4. A COBOL Directory Listing (Optional).
5. Punched Output (Optional).

CLU selectively punches a COBOL Source Library Entry or Entries as they appear on the CALL Library output tape. Each entry that is punched is preceded by a CATALS control statement containing the section number and entry name.

RENAME Parameter

◆ This parameter can be used to change the name of a module on a merge library tape when the module is transcribed to the output tape.

Format:

ΔRENAMEΔSYSxxx,aaaaaa,bbbbbb

Entry	Meaning
ΔRENAMEΔ	Parameter identifier.
SYSxxx	xxx = LIB module appears on SYSLIB. = UT4 module appears on SYSUT4. = UT5 module appears on SYSUT5.
,aaaaaa	aaaaaa = old name.
,bbbbbb	bbbbbb = new name.

Notes:

1. This parameter can only be used in conjunction with the MERGE parameter.
2. Multiple RENAME parameters may appear.
3. If the old name appears on more than one input merge file, the module from the unspecified library is treated as a distinct object module to be placed on the output tape (unless that module is later deleted by a delete parameter).

Example:

ΔRENAMEΔSYSLIB,ABLE,BAKER

ΔRENAMEΔSYSUT4,X1,X2

DELETO Parameter

◆ This parameter can be used to specify that the module named is <u>not</u> to be placed in the output library. It can only be used to delete modules which appear on library merge tapes SYSLIB, SYSUT4, and SYSUT5.

Format:

ΔDELETOΔaaaaaa

Entry	Meaning
ΔDELETOΔ	Parameter identifier.
aaaaaa	Name of module to be deleted.

Notes:

1. This parameter can only be used in conjunction with the MERGE parameter.
2. Multiple Delete parameters may appear.

Examples:

ΔDELETOΔPAYROL

ΔDELETOΔSAM

Equipment Configuration

Required

◆ Processor (65K).

Console Typewriter.

Magnetic Tape Devices (2).

CARD Reader.

Card Punch.

Printer.

Optional

◆ Additional magnetic tapes can be used as input devices.

Seven-level magnetic tape devices may be substituted for all input and output tapes.

Routine Parameters - General

◆ The following parameters are used with the CLU routine. Only those parameters required for a specific run need be submitted.

ENVIRON Parameter

This is an optional parameter that is used to alter the preset options of the CLU program. There are two types of ENVIRON parameters:

1. The first type of ENVIRON parameter is identified by CL as the first operand. This parameter must be first if used.
2. The second type of ENVIRON parameter is identified by AL, OL, or EL as the first operand.

CATALS Parameter

This parameter is used to designate that an entry is to be added to the COBOL Library.

DELETS Parameter

This parameter is used to delete a COBOL Source Library Entry or Entries from a particular input merge tape or from the output tape.

EXTRAC Parameter

This parameter is used to copy a COBOL Source Library Entry or Entries from an input extract tape to the output tape.

Routine Parameters - General *(Cont'd)*

STARTC Parameter

This parameter is used to apply corrections to an existing entry in a COBOL Source Library.

INSERT Parameter

This parameter is used to insert single or multiple source statements into an existing COBOL Source Library Entry.

DELETE Parameter

This parameter is used to remove single or multiple source statements from an existing COBOL Source Library Entry.

DSPLYE Parameter

This parameter is used to obtain a listing of source statements of all or selected entries within a COBOL Source Library section.

PUNCHE Parameter

This parameter is used to obtain card-punch output for source statements of all or selected entries within a COBOL Source Library section.

DSPCHE Parameter

This parameter is used to obtain a listing and card-punch output for source statements of all or selected entries within a COBL Source Library section.

END Parameter

This parameter denotes the end of parameter input to this routine.

Restrictions

◆ The following restrictions apply to the routine parameters:

1. The ENVIRON parameter, if present, must precede all other parameters. Within the ENVIRON parameter the CL operand, if used, must be first.
2. Parameters following the ENVIRON parameter must be in ascending order sequenced by section number and entry name.
3. Parameters containing the same section number and entry name must conform to the following:
   a. EXTRAC and DELETS are exclusive.
   b. EXTRAC and DELETS precede DSPLYE and/or PUNCHE or DSPCHE parameters.

Restrictions (Cont'd)

c. DSPLYE and/or PUNCHE or DSPCHE precede CATALS or STARTC parameters.

d. STARTC precede INSERT and/or DELETE parameters.

e. DSPLYE and/or PUNCHE or DSPCHE parameters that request the printing and/or punching of an entire section (*ALL) must precede all other parameters for that section.

Routine Parameters - Detailed

ENVIRON Parameter

◆ The ENVIRON parameter serves two functions: (1) to modify preset COBOL Source Library processing, and (2) to modify priority of coping the other Call Library Libraries.

1. If the ENVIRON CL parameter is not used, the following are assumed:
   a. One input merge library.
   b. No extract function.
   c. One output library.
   d. A standard COBOL Source Library on output.
   e. No directory listing.

   If multiple merge, extract, punch and/or print only or directory listing are required, the ENVIRON CL card is required and must be first control card.

   Format:

   ΔENVIRONΔCL, USASI, M=x, E=x, CD, NOPT

Entry	Meaning
ΔENVIRONΔ	Parameter identifier.
CL	Specifies modification of COBOL Source Library processing.
, USASI	Specifies format of COBOL Source Library. When used, the routine processes all COBOL divisional entries and all source program entries in one section (Section 1). If COBOL Libraries contain sections other than section 1, routine processes only section 1 of those libraries. The use of this operand allows the user to have 30 byte entry names in parameters; however, only the first 8 bytes of these entry names are utilized. If not used, the routine processes Standard COBOL Source Library.

ENVIRON Parameter (Cont'd)

Entry	Meaning
,M=x	Specifies number of input merge tapes. Value assigned may be 0,1,2, or 3. An 0 value indicates no input merge tapes. Value given specifies number of input tapes and priority. Given M=3, routine assigns MERGE3, MERGE2, and MERGE1 to input merge tapes and assigns highest priority to MERGE3. (Preset to 1.)
,E=x	Specifies number of input extract tapes. Value assigned may be 0, 1, 2, or 3. An 0 value indicates no input extract tapes. Value given specifies number of input extract tapes and priority. Given E=3, routine assigns XTAPE3, XTAPE2, and XTAPE1 to input extract tapes and assigns highest priority to XTAPE3. (Preset to 0.)
,CD	Optional entry used to produce a COBOL Directory Listing containing names of all entries within the COBOL Source Library.
,NOPT	Optional entry used to inhibit the creation of an output tape. When used, this routine is limited to printing and punching.

Note:

The total number of merge and extract tapes cannot exceed three.

Examples:

```
ΔENVIRONΔCL,M=2,CD

ΔENVIRONΔCL,E=1,CD

ΔENVIRONΔCL,M=2,E=1,CD

ΔENVIRONΔCL,NOPT
```

2. The second form of the ENVIRON parameter modifies the preset function for copying ASSEMBLY MACRO, OBJECT MODULE, and EXECUTIVE libraries. If these parameters are not used the libraries are copied from the highest priority merge tape on which each appears. When used this parameter must precede the first non-ENVIRON parameter.

ENVIRON Parameter (Cont'd)

Format:

ΔENVIRONΔx,y

Entry	Meaning
ΔENVIRONΔ	Parameter identifier.
x	Identifies library that is to have modified processing. It must be one of following: AL = Assembly Macro Library. OL = Object Module Library. EL = Executive Library.
y	Identifies which preset options must be changed prior to copying the AL, OL, or EL libraries. MERGEx - Indicates which merge tape contains the specified library that must be copied to output tape; x=1, 2, or 3. XTAPEx - Indicates which extract tape contains the specified library that must be copied to output tape; x=1, 2, or 3. DELETE - Inhibits the copying of specified library to output tape.

Note:

This parameter is preset to copy the AL, OL, and EL libraries from the highest priority merge tape.

Examples:

ΔENVIRONΔAL,MERGE2

ΔENVIRONΔOL,DELETE

CATALS Parameter

◆ The CATALS parameter is used to catalogue an entry into the COBOL Library.

Format:

ΔCATALSΔCx,name,a,b

Entry	Meaning
ΔCATALSΔ	Identifies parameter.
cx	Specifies section into which the entry is to be catalogued; x = 1 to 4 if Standard COBOL Source Library was indicated, or x = 1 if USASI COBOL Source Library was indicated.
,name	Name of entry being catalogued; one to eight characters, the first of which must be alphabetic. If name is USASI COBOL Source Library, the entry name may range from one to thirty characters.
,a	Optional: Specifies processing of the sequence number field of COBOL source statements for this entry. If this operand is not used routine assigns new sequence number to output source statements beginning with 000010 and incrementing by 10. Routine does not sequence check input. a = SEQNCE - Specifies that sequence checking of input is to be taken but not retained. If spaces are encountered in sequence number field, the routine processes that statement as a continuation of the previous source statement. a = SAVE - Specifies that sequence checking of input is to be taken and retained.
,b	Optional: Specifies printing and/or punching of this entry. b = DSPLYE - Specifies the printing of all output COBOL source statements for this entry. b = PUNCHE - Specifies the punching of all output COBOL source statements for this entry. First card punched is CATALS control card. b = DSPCHE - Specifies the printing and punching of all output COBOL source statements for this entry. First card punched is CATALS control card.

CATALS Parameter (Cont'd)

Examples:

ΔCATALSΔC2,SOUR1,SEQNCE,DSPLYE

ΔCATALSΔC1,SOUR2,DSPCHE

DELETS Parameter

◆ The DELETS parameter is used to delete a COBOL Source Library entry or entries from an input merge tape or output tape.

Format:

ΔDELETSΔa,Cx,name, ..., name

Entry	**Meaning**
ΔDELETSΔ	Identifies parameter.
a	Specifies which tape contains the entries to be deleted; a = MERGEx - Specifies which input merge tape contains entries to be deleted; x = 1, 2, or 3. a = OUTPUT - Specifies that entries must be deleted from the output tape.
,cx	Specifies library section number, x = 1, 2, 3 or 4. If USASI is used, x must equal 1.
,name	Name of entry to be deleted (one to thirty characters). Multiple names are permitted.

Note:

When processing a Standard COBOL Library, entry names may not exceed eight characters.

Examples:

ΔDELETSΔMERGE2,C3,SOUR1

ΔDELETSΔOUTPUT,C2,SOUR2

INSERT Parameter

◆ The INSERT parameter is used to insert single or multiple source statements into an existing COBOL Library entry directly after the source statement that matches the sequence number specified in the INSERT parameter.

Format:

INSERTΔa

Entry	Meaning
INSERTΔ	Identifies parameter.
a	A one- to six-character numeric sequence number field. Field must be terminated by a space or period.

Notes:

1. A STARTC parameter must precede the INSERT parameter.
2. One or more source statements must follow each INSERT parameter.

Examples:

INSERTΔ50

INSERTΔ000050.

DELETE Parameter

◆ The DELETE parameter is used to remove single or multiple source statements from an existing COBOL Library entry.

Format:

DELETEΔa

Entry	Meaning
DELETEΔ	Identifies parameter.
a	A single field or several fields containing sequence numbers or sequence number ranges of source statements to be deleted from an existing COBOL Library entry. If more than one field, fields must be separated by a comma followed by one or more spaces. Last field must be followed by space or period. If a field consists of a range of sequence numbers, the first number is separated from the last number by a hyphen. Sequence numbers are from one to six numeric characters.

DELETE Parameter (Cont'd)

Notes:

1. Any source statements that follow a DELETE parameter are inserted immediately after the last statement deleted.
2. A STARTC parameter must precede the DELETE parameter.

Examples:

DELETEΔ100.

DELETEΔ110.

DELETEΔ000120,Δ170-200.

DELETEΔ250-300,Δ350.

DELETEΔ390,ΔΔΔ420-000460,Δ500,Δ550-590.

DSPLYE Parameter

◆ This parameter is used to obtain a listing of the source statements of all or selected entries within a COBOL Library section.

Format:

$$\Delta\text{DSPLYE}\Delta\text{Cx} \left\{ \begin{array}{l} ,\text{*ALL} \\ ,\text{name},\ldots\text{.},\text{name} \end{array} \right\}$$

Entry	Meaning
ΔDSPLYEΔ	Identifies parameter.
cx	Specifies library section number; x = 1, 2, 3, or 4. If USASI is used, x must equal 1.
,name or ,*ALL	Name of entry to be displayed (one to thirty characters). If first name is *ALL, all entries for section specified are displayed. Multiple names are permitted.

Note:

When processing a Standard COBOL Library, entry names may not exceed eight characters.

Examples:

ΔDSPLYEΔC2,SOUR3

ΔDSPLYEΔC3,*ALL

PUNCHE Parameter

◆ This parameter is used to punch source statements of all or selected entries within a COBOL Library section.

Format:

$$\Delta\text{PUNCHE}\Delta\text{Cx} \left\{ \begin{array}{l} ,\text{*ALL} \\ ,\text{name},\ldots.,\text{name} \end{array} \right\}$$

Entry	Meaning
ΔPUNCHEΔ	Identifies parameter.
cx	Specifies library section number; x = 1, 2, 3, or 4. If USASI is used, x must equal 1.
,name or ,*ALL	Name of entry to be displayed (one to thirty characters), If first name is *ALLΔ, all entries for section specified are displayed. Multiple names are permitted.

DSPCHE Parameter

◆ This parameter is used to obtain a listing and punch output of source statements of all or selected entries within a COBOL Library section.

Format:

$$\Delta\text{DSPCHE}\Delta\text{Cx} \left\{ \begin{array}{l} ,\text{*ALL} \\ ,\text{name},\ldots.,\text{name} \end{array} \right\}$$

Entry	Meaning
ΔDSPCHEΔ	Identifies parameter.
cx	Specifies library section number; x = 1, 2, 3, or 4. If USASI is used, x must equal 1.
,name or ,*ALL	Name of entry to be displayed (one to thirty characters). If first name is *ALLΔ, all entries for section specified are displayed. Multiple names are permitted.

END Parameter

◆ The END parameter is mandatory and denotes the end of input parameter for the CLU routine.

Format:

ΔENDΔ

Considerations for Use

◆ 1. Entries appearing on more than one input will be copied to output first from the extract tape if specified, then from the highest priority merge tape.

2. Multiple ENVIRON parameters relating to the same Library are not permitted; that is there may be only one ENVIRONΔCL parameter and one ENVIRONΔAL parameter, etc.

3. If a format error is detected in a DELETE parameter, it is flagged and rejected; all source statements occurring between the rejected DELETE parameter and the next parameter are bypassed.

4. *Tape Labels*

   The CLU routine performs a purge-date check on the output tape (SYSUT2). A single TM or a volume label set (VOL, HDR, TM) will be acceptable for purge date checking.

   If no VOL label is present, a dummy VOL label is written; if a VOL label is present, it is retained.

   A standard HDR label is written with the file identifier of SYSLIB. This will overlay any existing HDR on the output tape. A TM is written following the HDR label.

5. *Record Compression and Blocking*

   All source statements written to the output library are compressed. Compression is accomplished by replacing each field of two or more spaces by a one-byte counter. The compressed records then blocked up to 489 bytes per block.

   If a character with a hexadecimal value of less than $40_{(16)}$ appears in the source statement, compression is not attempted and the entire card is written to tape.

Device Assignments

◆ Under Monitor Control:

SDN	Device Type	Remarks
SYSIPT	Card reader or magnetic tape.	Parameter input and COBOL source statement.
SYSLST	Printer or magnetic tape.	Output listings.
SYSOPT	Card punch or magnetic tape.	Punched entries from COBOL Source Library.
OUTPUT	Magnetic tape.	Updated TOS CALL Library.
WKTAPE	Magnetic tape.	Storage.
MERGE1 to MERGE3	Magnetic tape.	TOS CALL Libraries.
XTAPE1 to XTAPE3	Magnetic tape.	TOS CALL Libraries.

Under Executive Control:

SDN	Device Type	Remarks
CRDRDR	Card reader.	Parameter input and COBOL source statements.
PRINTR	Printer.	Output listings.
CRDPCH	Card punch.	Punched entries from COBOL source Library.
OUTPUT	Magnetic tape.	Updated TOS CALL Library.
WKTAPE	Magnetic tape.	Storage.
MERGE1 to MERGE3	Magnetic tape.	TOS CALL Libraries.
XTAPE1 to XTAPE3	Magnetic Tape.	TOS CALL Libraries.

Note:

The total number of input tapes consisting of MERGE and/or EXTRACT tapes cannot exceed three.

Input Statement Listing

◆ Following is a layout of the input statement listing produced by CLU.

```
MM/DD/YY  COBOL Library Update - VERxxx  PAGE yyyy

INPUT STATEMENT LISTING                  ERROR FLAGS

sss data...................data                ffff
sss data...................data                ffff
```

where:

xxx = The CLU program version number.

yyyy = The page number (0001-9999).

sss = A sequence number (001-999) assigned sequentially to control statement, and is used to associate error records which appear at the end of the listing with the control statements.

data = Represents either 80 column control and source statements or error records.

ffff = The four character error code appended to control and source statements.

COBOL Entry Listing

◆ Following is a layout of the COBOL Entry Listing produced by CLU.

```
MM/DD/YR  COBOL LIBRARY UPDATE - VERxxx  PAGE yyyy

COBOL ENTRY LISTING SECTION z  ENTRY n..n d...d

data....................................data
data....................................data
```

where:

xxx = The CLU program version number.

yyyy = The page number (0001-9999).

z = COBOL Library section number.

n..n = A 1 to 8 character entry name.

d..d = The date (MM/DD/YR) that the entry was created, or the date of last modification.

data = The 80 column source statement.

STARTC Parameter

◆ The STARTC parameter is used to apply corrections to an existing entry in a COBOL Source Library. If the STARTC parameter is not followed by an INSERT or DELETE parameter, the entry specified in the STARTC parameter is either copied directly to output tape (SAVE option used) or resequenced and then copied to output tape (SAVE option not used).

Format:

ΔSTARTCΔCx,name,SAVE,a

Entry	**Meaning**
ΔSTARTCΔ	Identifies parameter.
Cx	Specifies COBOL Library Section number, x = 1, 2, 3 or 4. If USASI is used, x must equal 1.
,name	Name of entry to be corrected (one to thirty characters). Standard COBOL Library name may not exceed eight characters.
,SAVE	Optional: When used, the routine retains input sequence numbers. All source statements that are inserted must contain sequence numbers. If not used, the routine resequences all source statements before they are written to tape.
,a	Optional: Specifies printing and/or punching of this entry. a = DSPLYE - Specifies printing of all output COBOL source statements for this entry. a = PUNCHE - Specifies punching of all output COBOL source statements for this entry. a = DSPCHE - Specifies printing and punching of all output COBOL source statements for this entry.

Example:

ΔSTARTCΔC2,SOUR2,SAVE,DSPLYE

COBOL Directory Listing

◆ Following is a layout of the COBOL Directory Listing produced by CLU.

```
MM/DD/YR  COBOL LIBRARY UPDATE - VERxxx  PAGE yyyy

COBOL DIRECTORY LISTING  SECTION z

(n....n   d...d).........................(n....n   d....d)
(n....n   d....d)........................(n....n   d....d)
(n....n   d....d)........................(n....n   d....d)
```

where:

xxx = The CLU program version number.

yyyy = The page number (0001-9999).

z = The COBOL Library section number.

n..n = The entry name.

d..d = The date (MM/DD/YR) that the entry was created, or the date of last modification.

SOURCE LIBRARY UPDATE (SLU)

General Description

◆ The Source Library Update (SLU) routine is used to maintain or display programs stored on source library tapes created by the TOS Assembler or a previous SLU routine. These tapes contain Assembly source language statements.

Source programs may be reordered, renamed, added, deleted, printed, punched, or simply copied to an output library. Individual statements within source programs may be resequenced, added, deleted, or replaced. Also, portions of individual statements may be replaced.

Preset Functions

This routine is preset to execute the following functions:

To process one input source library tape (MERGE1).

To produce an output source library in fixed-length, blocked format (SYSUT5). Each block contains five source statements.

To process source programs in sequence by program name.

To print a directory of the programs appearing on the output library (SYSUT5).

Optional Functions

The following options can be selected by means of routine parameters:

To process multiple input: merge tapes, extract tapes, and punched cards.

To reorder out-of-sequence input tapes.

To rename, add, delete, print, or punch input source programs.

To compress, replace, delete, add, or resequence source statements.

To replace portions of individual statements.

To print a directory of programs contained on an input merge tape.

To create a second source program output tape (OPTOUT).

To process programs in an unsequenced mode.

EXTRACT Parameter

◆ This parameter is used to extract a particular object module from an existing Object Module Library section for placement onto the output library. The module to be extracted must appear on a Call Library mounted on SYSUT1, SYSUT6, or SYSUT7.

If desired, modules to be extracted may also be renamed in the process of being transcribed to the output library.

Format:

ΔEXTRACTΔSYSUTn,aaaaaa,bbbbbb

Entry	Meaning
ΔEXTRACTΔ	Parameter identifier.
SYSUTn	n = 1 module appears on SYSUT1. = 6 module appears on SYSUT6. = 7 module appears on SYSUT7.
,aaaaaa	aaaaaa = name of input module.
,bbbbbb	bbbbbb = new name of module when placed on the output library. (If omitted, the original name is retained.)

Note:

Multiple EXTRACT parameters may appear.

Examples:

ΔEXTRACTΔSYSUT1,PAYROL

ΔEXTRACTΔSYSUT6,PAYROL,SALARY

CATALO Parameter

◆ This parameter is used to add an object module file produced by a language translator to an OML. During this process the file is converted to object module library format.

OMF's to be cataloged can appear on one of four input tapes. If desired, modules can also be renamed before being transcribed to the output library.

Format:

ΔCATALOΔSYSxxx,aaaaaa,bbbbbb

Input

◆ Input to this routine can consist of the following:

1. Routine parameters.

2. An existing source library tape.

   This tape may be the output of a previous Source Library Update routine or it may be the output from the TOS Assembler.

3. Source Programs.

   Source programs may be entered from up to eight tape input devices and from the card reader (SYSIPT).

   Any combination of merge or extract tape inputs are accommodated.

Output

◆ Output of this routine can consist of the following:

1. A Source Library Tape (SYSUT5).

   This tape has the following format:

   a. Standard VOL label.

   b. Standard HDR label with "SOURCE∆LIB∆∆∆∆∆∆" as the file identifier.

   c. Tape mark.

   d. 80-character STARTC program identifier block.

   e. Source statements blocked 5 statements per block or compressed to a maximum of 480 bytes per block.

   f. Tape mark separating each program.

   g. Double tape mark after the last program.

2. An optional output tape (OPTOUT)

   This tape contains programs selected by the OPTOUT parameter or the STARTC parameter. It has the following format:

   a. Standard VOL label.

   b. Standard HDR label with "SOURCE \LIB \OPTOUT" as the file identifier.

   c. Tape mark.

   d. 80-character STARTC program identifier block.

Output (Cont'd)

e. 80-character, unblocked source statements.

f. Tape mark separating each program.

g. Double tape mark after the last program.

3. Source programs punched on 80-column cards.

4. Printed listings:

a. Environmental Map.

This listing displays the magnetic tape devices required and the processing options specified for each.

b. Source Program and Corrections Listing.

A source program listing is provided for programs as specified by the ENVIRON, PRINT, or STARTC parameters. This listing occurs after all update processing has been completed. Corrections made to programs are also listed.

c. Control Requests Not Honored Listing.

This listing indicates processing options that could not be honored or that all options were performed. It is provided at the end of the routine.

d. Directory of Source Programs.

This listing is provided whenever an output library (SYSUT5) is produced. It consists of the program name, version number, and version date for each program on the output tape. If an output library tape is not provided, and only one merge input tape is used, a directory of MERGE1 can be obtained. (Refer to the D operand in the ENVIRON parameter.)

Equipment Configuration

Required

◆ Processor (65K)
Console typewriter
Two magnetic tape devices
Card reader or Videoscan document reader with card read feature.
Printer
Card punch

Optional

◆ Additional magnetic tapes can be used as input devices.

Seven-level magnetic tape devices may be substituted for all input and output tapes.

Routine Parameters-General

◆ There are two levels of routine parameters corresponding to the two processing levels of this routine: Level I and Level II.

General library processing is specified by Level I parameters. These parameters are read and stored prior to processing.

Individual program processing is specified by Level II parameters. Each Level II parameter is read and processed in the order of appearance. Source statements following a STARTC parameter are considered replacements or insertions if the program named in the STARTC card has been selected from a merge or extract tape. If the program named in the STARTC card was not selected, the source statements are considered as a new program.

All parameters are optional; however, the following restrictions apply:

1. Level I parameters must precede Level II parameters.
2. When supplied, the ENVIRON parameter must be first. Other Level I parameters may follow in any order.
3. Level II parameter sets must appear in sequence by program name if processing in a sequenced mode. If processing in an unsequenced mode, they must be in the same order as the input programs.
4. The first parameter of each Level II set must be STARTC. The last parameter of each set must be ENDC.
5. Source statements associated with a STARTC parameter must be in their insertion or replacement order.
6. The END parameter (or // card if running under Monitor) must be the last card.

Summary of Level I Parameters

Parameter	Use
ENVIRON	To specify general library processing. When supplied, it must be the first parameter.
RENAME	To copy a source program from an input tape to SYSUT5 under a new name, or under both an old name and a new name.
DELETE	To prevent the writing of an input source program to SYSUT5.
EXTRACT	To copy a source program from an extract tape to SYSUT5.
OPTOUT	To copy any source program from an input device to an optional output tape.

Routine Parameters-General *(Cont'd)*

Summary of Level I Parameters (Cont'd)

Parameter	Use
REORDR	To provide sequencial processing of out-of-sequence programs on the input tape specified.
PRINT	To cause specific source programs to be listed.
PUNCH	To cause specific source programs to be punched.

Summary of Level II Parameters

Parameter	Use
STARTC	To identify a source program and parameter set. To specify general processing for a source program. This parameter must be the first of each level II parameter set supplied.
DELETE	To delete individual source statements by sequence numbers.
COL	To replace portions of a source statement.
ENDC	To indicate the end of a Level II parameter set.

Note:

Source statements may be supplied for the following processing:

To replace input source statements by sequence numbers.

To insert a source statement immediately before the first input source statement with a higher sequence number.

To add (insert) a complete source program to the output library.

END Parameter

This parameter indicates the end of all parameter input to this routine.

Routine Parameters-Detailed

ENVIRON Parameter

Ⓒ

◆ *Format:*

*ENVIRONΔT=t,M=m,E=e,N=n,S=s,C=c,A=a,L=x,P=p,D=d,R=r

Ⓐ

Ⓐ

Entry	**Meaning**
*ENVIRONΔ	Parameter identifier.
T=t	Specifies type of library to be processed. If present it must be the first operand in the *ENVIRON statement. t=F Used to create FORTRAN source output. =C Used to create COBOL source output. =S Used in all other cases.
M=m	Number of input merge tapes: m = 0-8 merge tapes required (preset=1). Note: The number of merge tapes plus the number of extract tapes must not exceed eight. If M=0 and E is not used, all input source programs are assumed to be on SYSIPT.
,E=e	Number of input extract tapes: e = 1-8 extract tapes required (preset=no extract tapes). Note: See note above.
,N=n	Output source library on SYSUT5: n=Y yes (preset). =N no.
,S=s	Sequence of output source library: s=Y source programs on output library must be in sequence by program name (preset). =N source programs do not need to be in sequence on the output library. Note: When S=Y, the input source programs also must be processed in sequence, and REORDR parameter may be required. When S=N, only one input tape is allowed.
,C=c	Output record format on SYSUT5: c=N blocked card-image records (preset). =Y blocked compressed records.
,A=a	Optional output tape on OUTPUT: a=N no (preset). =Y yes.

ENVIRON Parameter (Cont'd)

Entry	Meaning
,L=x	x=N do not list source programs unless specified by a PRINT or STARTC parameter (preset). =Y List all programs selected by Level I parameters
,P=p	p=N do not punch source programs unless specified by a PUNCH or STARTC parameter (preset). =Y punch all programs selected by Level I parameters.
,D=d	Directory of MERGE1 input tape: d=N not required (preset). =Y list all program names on input tape MERGE1. *Note:* If D=Y, operands N=N and M-1 must be used.
,R=r	SYSUT5 and OPTOUT validity check: r=N not required (preset). =Y Check SYSUT5 and OPTOUT tapes. *Note:* When this option is specified the routine reads SYSUT5 and OPTOUT in reverse and checks the byte count on the tape against what was written. Since the byte count check is a read reverse operation the routine cannot check the byte count on seven-level tapes. When SYSUT5 or OPTOUT is a seven-level tape, only the block count is checked.

Note:

All operands supplied after the first must be preceded by a comma.

Examples:

```
*ENVIRONΔM=3,E=2,C=Y,A=Y
*ENVIRONΔM=1,N=N,S=N,P=Y,D=Y
*ENVIRONΔM=2,A=Y,L=Y,R=Y
```

RENAME Parameter

◆ *Format:*

$$\text{*RENAME}\Delta \begin{Bmatrix}\text{MERGEn}\\ \text{XTAPEn}\end{Bmatrix}\text{,aaaaaaaa,bbbbbbbb,SAVE}$$

RENAME Parameter (Cont'd)

Entry	Meaning
*RENAMEΔ	Parameter identifier.
MERGEn or XTAPEn	Symbolic input device name; where n=1-8.
,aaaaaaaa	Old name (one to eight characters) of source program.
,bbbbbbbb	New name (one to eight characters) of source program on SYSUT5.
,SAVE	Optional. Causes the program to be written <u>twice</u> to SYSUT5, once under its old name and once under its new name.

Examples:

```
*RENAMEΔMERGE1,PROGA,PROGB
*RENAMEΔXTAPE3,TSM3,MCDA,SAVE
```

Notes:

1. If a program is renamed and not saved, the old name may not be used by any other Level I parameters.
2. An Extract parameter is not required for a program being renamed from an Extract input.

DELETE Parameter

◆ *Format:*

*DELETEΔ $\left\{\begin{matrix}\text{MERGEn}\\\text{SYSUT5}\end{matrix}\right\}$,pppppppp,......,pppppppp

Entry	Meaning
*DELETEΔ	Parameter identifier.
MERGEn or SYSUT5	MERGEn = delete specified program(s) from merge tape, where n = 1-8. SYSUT5 = inhibit specified program(s) from appearing on SYSUT5.
,pppppppp	Name of program (one to eight characters) to be deleted. <u>Note</u>: Multiple names can be specified with a comma used to separate each name. Program names may appear in any order.

DELETE Parameter (Cont'd)

Examples:

*DELETEΔMERGE3,PROGAA,IFOPT2

*DELETEΔSYSUT5,DSCSRT,OMALL

*DELETEΔMERGE5,SRTGEN3

Note:

If a program is deleted from MERGEn, no other Level I actions can be specified for that program. If a program is deleted from SYSUT5, other Level I actions (PRINT, PUNCH, OPTOUT) may be specified.

EXTRACT Parameter

◆ *Format:*

*EXTRACTΔXTAPEn,pppppppp,.....,pppppppp

Entry	Meaning
*EXTRACTΔ	Parameter identifier.
XTAPEn	Symbolic device, where n = 1-8.
,pppppppp	Name of the program (one to eight characters) to be extracted from the device specified. Note: Multiple names can be specified with a comma used to separate each name. Program names may appear in any order.

Examples:

*EXTRACTΔXTAPE2,OUTFIL

*EXTRACTΔXTAPE3,PROGAB,BUSTOUT

REORDR Parameter

◆ *Format:*

*REORDR $\begin{Bmatrix}\text{MERGEn}\\\text{XTAPEn}\end{Bmatrix}$

Entry	Meaning
*REORDRΔ	Parameter identifier.
MERGEn or XTAPEn	Symbolic device name: MERGEn = reorder programs on merge tape, where n = 1-8. XTAPEn = reorder programs on extract tape, where n = 1-8. Note: Multiple devices may be specified with a comma used to separate each device name.

Notes:

1. All devices specified after the first must be preceded by a comma.
2. Any or all input devices may be reordered.

REORDR Parameter (Cont'd)

3. A maximum of 399 programs per device may be reordered.
4. A merge tape does not require reordering if RENAME or DELETE processing corrects the out-of-sequence condition.
5. An extract tape does not require reordering if RENAME processing corrects the out-of-sequence condition or if the programs are processed in the order that they appear on the tape.
6. A single input tape does not require reordering if S=N is used in the ENVIRON parameter.

Examples:

```
*REORDRΔXTAPE1

*REORDRΔMERGE2,MERGE3,XTAPE2
```

OPTOUT Parameter

◆ *Format:*

*OPTOUTΔ $\left\{\begin{matrix}\text{MERGEn}\\\text{XTAPEn}\end{matrix}\right\}$,pppppppp,...,pppppppp

Entry	Meaning
*OPTOUTΔ	Parameter identifier.
MERGEn or XTAPEn	MERGEn = merge tape containing program(s) to be written to optional output tape (OPTOUT), where n = 1-8. XTAPEn = extract tape containing program(s) to be written to optional output tape (OPTOUT), where n = 1-8.
,pppppppp	Name of program (one to eight characters) selected for the optional output tape (OPTOUT). Note: Multiple names can be specified with a comma used to separate each name. Program names may appear in any order.

Examples:

```
*OPTOUTΔMERGE4,TPEDIT,TPINIT,TOSGEN

*OPTOUTΔXTAPE3,FMSRT,VMSRT
```

PRINT Parameter

◆ *Format:*

$$\text{*PRINT}\wedge \begin{Bmatrix}\text{MERGEn}\\ \text{XTAPEn}\end{Bmatrix} \text{,pppppppp,..,pppppppp}$$

Entry	Meaning
*PRINT Δ	Parameter identifier.
MERGEn or XTAPEn	MERGEn = merge tape containing program(s) to be printed, where n = 1-8. XTAPEn = extract tape containing program(s) to be printed, where n = 1-8.
,pppppppp	Name of the program (one to eight characters) to be printed. Note: Multiple names can be specified with a comma used to separate each name.

Examples:

*PRINT∧XTAPE3,RINGO

*PRINT∧MERGE1,PROG1,IPROGA

PUNCH Parameter

◆ *Format:*

$$\text{*PUNCH}\Delta \begin{Bmatrix}\text{MERGEn}\\ \text{XTAPEn}\end{Bmatrix} \text{,pppppppp,..,pppppppp}$$

Entry	Meaning
*PUNCHΔ	Parameter identifier.
MERGEn or XTAPEn	MERGEn = merge tape containing program(s) to be punched, where n = 1-8. XTAPEn = extract tape containing program(s) to be punched, where n = 1-8.
,pppppppp	Name of the program (one to eight characters) to be punched. Note: Multiple names can be specified with a comma used to separate each name.

CATALO Parameter (Cont'd)

Entry	Meaning
ΔCATALOΔ	Parameter identifier.
SYSxxx	xxx = IPT OMF appears on SYSIPT. = UT1,UT6, or UT7 OMF appears on device specified.
,aaaaaa	aaaaaa = object module name.
,bbbbbb	bbbbbb = new name of module when placed on the OML. (If omitted, the original name is retained.)

Notes:

1. Catalog parameters should be submitted in sequence by OMF name as they appear on the input file.
2. If the name entry is omitted (see example 1), all object modules on the input file will be added to the library.
3. If any of the merge library tapes contains a module with the same name as the module being cataloged, the module being cataloged has priority and the other modules are deleted.

Examples:

ΔCATALOΔSYSIPT

ΔCATALOΔSYSIPT,PAYROL

ΔCATALOΔSYSUT6,ABLE,BAKER

DSPLY Parameter

◆ This parameter can be used to obtain a program listing or an index of any or all modules in an OML section of a Call Library Tape.

When used with updating parameters such as COPY, MERGE, and EXTRACT, this parameter displays the modules appearing on the output library tape. When used without updating parameters, a listing of the modules contained in the library mounted on SYSLIB is provided.

Formats:

ΔDSPLYΔaaaaaa

ΔDSPLYIΔaaaaaa

Entry	Meaning
ΔDSPLYΔ or ΔDSPLYIΔ	Parameter identifier. DSPLY causes object modules to be printed; DSPLYI causes only an object module index to be produced.
ALL or aaaaaa	ALL = display all modules appearing in library. aaaaaa = display module aaaaaa only.

PUNCH Parameter (Cont'd)

Examples:

```
*PUNCHΔMERGE3,TPTPC
*PUNCHΔXTAPE1,CDRAC,CDRAD,CDRAE
```

STARTC Parameter

◆ *Format:*

```
*STARTCΔpppppppp,S=s,F=f,G=(n,a,b,c),V=(n,c),R=(c,e),P=p,L=x,A=a
```

Entry	**Meaning**
*STARTCΔ	Parameter identifier.
pppppppp	Name of the program (one to eight characters to be processed by the subsequent parameters. (Refer to page 6-104, Insertion and Replacement Parameters.)
,S=s	Sequence check input: s=Y check the sequence field described by the F operand. (This check is made after all updating has been performed and before any new sequence numbers are generated.) =N no input sequence check required (preset).
,F=f	f = length of input sequence field (4-8). The field must end in column 80. If not used, columns 73-80 are assumed.
,G=(n,a,b,c)	Generate output sequence numbers: n = length of generated sequence field (4-8). The field must end in column 80. If not used, columns 73-80 are assumed. a = increment for sequence numbers (1-999). If not used, an increment of 100 is assumed. b = value of first sequence number. If not used, zeros are assumed. The length of this value must be the length given in n. c = value of input source statement sequence No. after which sequence generation is to begin. If not used, sequence number generation begins with the first source statement. The length of this value must be the length given in F above. Note: If sequence numbers are desired with all of the preset functions, only the entry ',G' is used.

STARTC Parameter (Cont'd)

Entry	Meaning
,V=(n,c)	Program version number: n = version number to appear in the output STARTC card (001-999). If not used, a version number of 001 is assigned. c = first column of each output source statement read from SYSIPT to contain the version number (01-80). If not used, the version number will be in the STARTC card only. Note: The version number is included in source statements only when the columns are blank.
,R=(c,e)	Source statement addition constant: c = column of each output source statement where new data is to start (01-80). e = constant (any EBCDIC character) that is to be placed in every output source statement (one to six characters). Note: Parenthesis are not valid characters.
,P=p	p = Y punch this program. = N do not punch this program (preset). Note: This operand overrides PUNCH parameter and the P operand in the ENVIRON parameter.
,L=x	x = Y list this program. = N do not list this program (preset). Note: This operand overrides PRINT parameter and the L operand in the ENVIRON parameter.
,A=a	a = Y write program to OPTOUT. = N do not write program to OPTOUT (preset). Note: If A=Y, then the A=Y operand in the ENVIRON parameter is required also.

Ⓒ

Note

All operands except ppppppp are optional.

DELETE Parameter (Cont'd)

2.
```
*STARTCΔPROGA,F=4
*DELETE,0020,0030
*ENDC
```

Input Merge or Extract Tape	SYSUT5 (Output)
*STARTCΔPROGA	*STARTCΔPROGA
.....0010	0010
.....0014	0014
.....0020	0040
.....0035	
.....0030	
.....0040	

3.
```
*STARTCΔPROGB1,F=5
*DELETEΔ00105,00125
*ENDC
```

Input Merge or Extract Tape	SYSUT5 (Output)
*STARTCΔPROGB1	*STARTCΔPROGB1
....00100	00100
....00107	00104
....00104	00130
....00110	
....00115	
....00125	
....00130	

STARTC Parameter (Cont'd)

Examples:

```
*STARTCΔSTOCKUP,F=7,S=Y,G=(7,500,0000500),V=(005)
*STARTCΔSTOCKUP,P=Y,L=Y,A=Y
*STARTCΔTOS441,V=(002),R=(45,0004),L=Y
```

DELETE Parameter

◆ *Format:*

*DELETE $\left\{\begin{matrix}, \\ \text{or} \\ \Delta\end{matrix}\right\}$ x,y

Entry	Meaning
*DELETE	Parameter identifier.
, or Δ	, = delete all statements in the program beginning with the one containing sequence number x through and including the one containing sequence number y. Sequence numbers x and y must be used in the program. Δ= delete only those statements in the program with sequence numbers equal to or greater than x and equal to or less than y. Sequence numbers x and y do not have to be used in the program.
x,y	x = sequence number of first statement to be deleted (four to eight characters). y = sequence number of last statement to be deleted (four to eight characters). Optional.

Notes:

1. To delete one statement, the x entry only need be supplied.
2. The length of the sequence number fields must agree with the length (F entry) used in the STARTC parameter.

Examples:

```
1. *STARTC\PROGA,F=4    Input Merge or        SYSUT5
   *DELETE,0025          Extract Tape         (Output)
   *ENDC
                        *STARTC\PROGA     *STARTC\PROGA
                        .....0010         .....0010
                        .....0014        ......0014
                        .....0020         .....0020
                        .....0025         .....0030
                        .....0030         .....0040
                        .....0040
```

COL Parameter

◆ *Format:*

Card Columns	Content	Meaning
1-4	*COL	Parameter identifier.
5-6	cc	Column of input source statement where data replacement is to start (01-71).
7-71	new data	Replacement data to be placed into the input source statement. (This data will begin in the column specified by cc and end when column 71 of this parameter or column 71 of the source statement is reached, whichever occurs first.)
72		Not used; leave blank.
73-80	sequence number	Sequence number (eight positions) of the input source statement which is to receive the data. (This number must end in column 80 and be the same length as given in the STARTC card.)

Example:

To change the comments portion, column 40 through 71, of source statement 00505:

```
*COL40LOAD∧R3∧WITH∧R5∧PLUS∧8∧∧....∧∧∧00505
```

ENDC Parameter

◆ *Format:*

```
*ENDC
```

This parameter indicates the end of Level II parameters for the program named in the *STARTC∧ parameter. The end of Level II parameters is also indicated by a new *STARTC card, an *END card, or a // card if running under Monitor.

END Parameter

◆ *Format:*

```
*END
```

This parameter indicates the end of parameter input. A // card will indicate the end of parameters when the routine is run under Monitor.

Consideration for Use

◆ This section discusses a number of areas that should be considered before using the SLU routine. The programmer is advised to review these points before preparing his parameter input. In addition, the format of the source library tape, as described in Appendix B, page B-18, should be reviewed.

Sequencing of Source Programs

◆ When only one library or the card reader is used as input, programs may be in or out of sequence. However, when multiple input devices are used, source programs must appear in sequence by program name on each device.

Sequencing of input programs may be accomplished by the Level I parameters REORDR, RENAME, and DELETE.

Source Program Processing

◆ When processing multiple input, the merge/extract logic selects programs in sequence and according to device priority. Merge and extract tapes have ascending priority and extract tapes have higher priority than merge tapes. MERGE1 has lowest priority and XTAPE 8 has the highest priority. Source programs on the card reader are processed according to the output sequence.

Level I Processing

◆ Processing is determined by the Level I parameters supplied. Routine input and output is determined as well as general processing of the library. Parameters, except for ENVIRON, may be in any order.

Level II Processing

◆ Processing is specified by the Level II parameters for the program named in the STARTC parameter. Only programs selected from an extract or merge tape by Level I parameters or that appear on SYSIPT may be processed.

Parameters must be in the order of the input.

Replacement and Insertion of Source Statements

◆ Source statements that follow a STARTC parameter which names a program selected from a Merge or Extract tape are processed as follows:

1. Source statements with the same sequence number replace the old source statements.
2. Source statements with new sequence numbers are inserted in alphanumeric sequence.
3. Any source statements that do not have sequence numbers are inserted immediately after the preceding source statement that contained a sequence number. For example:

Input Tape	SYSIPT (Input)	SYSUT5 (Output)
*STARTC PROGA	*STARTC PROGA,F=4	*STARTC PROGA
.....0010	0015	0010
.....0020	0030	.....0015 (new)
.....0040	0040	0020
.....0050	.....----	.....0030 (new)
	.....----	.....0040 (new)
	*ENDC	.....---- (new)
		.....---- (new)
		0050

Replacement and Insertion of Source Statements (Cont'd)

Note:

If an output sequence check had been specified, blank sequence fields as shown above would cause an error. To prevent this, sequence number generation should be specified in the STARTC card.

Source statements that follow a STARTC parameter that <u>does not</u> name a program selected from a Merge or Extract tape are added to the output tape (SYSUT5) as a new program.

Restrictions

◆ 1. The maximum number of programs that can be reordered on a merge tape is 399.

2. Source statements may not start with '*STARTC', '*ENDC', '*DELETE,', '*DELETEΔ', '*COL', or *END when supplied for Level II pocessing.

3. Multifile volumes or multivolume files may not be processed.

4. Any program without an END card will have one generated in the output, except when 'C' is specified in the T operand of the *ENVIRON statement. Ⓐ

5. The compressed format output of the SLU is not accepted as input to the Assembler. The blocked format of SYSUT5 or the OPTOUT format must be used.

Parameter Examples

◆ 1. To use the preset functions to produce a duplicate copy of a Source Library Tape.

```
//ΔASSGNΔSYSIPT,R1
//ΔASSGNΔSYSLST,L1
//ΔASSGNΔMERGE1,01
//ΔASSGNΔSYSUT5,02
// END
*END
```

2. To copy a source library tape adding programs PROGA and PROGH from SYSIPT:

```
//Δrun-time parameters (same as example 1.)
*STARTCΔPROGA
Source input for PROGA
*ENDC
*STARTCΔPROGH
Source input for PROGH
*ENDC
*END
```

3. To merge two source library tapes and extract programs OMALL and RINGO from a third. The output library is to be in compressed format and a listing of all programs is desired:

Parameter Examples *(Cont'd)*

```
//ΔASSGNΔSYSIPT,R1
//ΔASSGNΔSYSLST,L1
//ΔASSGNΔMERGE1,01
//ΔASSGNΔMERGE2,02
//ΔASSGNΔXTAPE1,03
//ΔASSGNΔSYSUT5,04
//ΔEND

*ENVIRONΔM=2,C=Y,L=Y,E=1
*EXTRACTΔXTAPE1,OMALL,RINGO
*END
```

4. To punch two programs, PROGR from XTAPE1 and PROGT from XTAPE2. Program PROGR is to be sequence-checked and written to the optional output tape OPTOUT. Program PROGT is to be updated and have new sequence numbers generated.

```
//ΔASSGNΔSYSIPT,R1
//ΔASSGNΔSYSLST,L1
//ΔASSGNΔXTAPE1,01
//ΔASSGNΔXTAPE2,02
//ΔASSGNΔOPTOUT,03
//ΔASSGNΔSYSOPT,P1
//ΔEND

*ENVIRONΔM=0,E=2,A=Y
*EXTRACTΔXTAPE1,PROGR
*EXTRACTΔXTAPE2,PROGT
*STARTCΔPROGR,S=Y,F=6,V=(004),P=Y,A=Y
*ENDC

*STARTCΔPROGT,G=(5,200,00300),P=Y,F=5
*DELETE,00750,01200
*DELETEΔ02250

 {Source statements to}
 {   update PROGT     }

*ENDC
*END
```

5. To extract a program from XTAPE1 and write it to SYSUT5 under both a new name and under the old name. The program under the new name is to be updated.

Parameter Examples

```
//∧ASSGN∧SYSIPT,R1
//∧ASSGN∧SYSLST,L1
//∧ASSGNΔXTAPE1,01
//∧ASSGNΔSYSUT5,02
//∧END
*ENVIRONΔM=0,E=1
*RENAMEΔXTAPE1,PROGAV1,PROGAV2,SAVE
*STARTCΔPROGAV2

 { Source statements to }
 {   update PROGAV2     }

*END
```

6. The following example illustrates a Monitor job stream consisting of the Source Library Update, Assembler, Linkage Editor, and execution of one program. The SLU will create an updated source Library tape from inputs MERGE1 and XTAPE1 to SYSUT5.

   Programs PROGA, PROGM, and PROGP from SYSUT5 are assembled and bound as three separate programs. PROGM is then executed.

```
//ΔSTARTM
//ΔASSGNΔSYSLST,L1
//ΔASSGNΔSYSUT1,01
//ΔASSGNΔSYSUT2,02
//ΔASSGNΔSYSUT3,03
//ΔASSGNΔSYSLIB,04
//ΔASSGNΔSYSUT5,05
//ΔJOB ΔΔΔΔ ASSMBL,LNKEDT,AND RUN
//ΔASSGNΔMERGE1,06
//ΔASSGNΔXTAPE1,07

//ΔEXECΔSLU

 { SLU parameters to produce          }
 { updated Source Library on SYSUT5   }

*END

//ΔPARAM  INPUT=SYSUT5
//ΔASSMBL
*STARTCΔPROGA,ASSEMBLE
*STARTCΔPROGM,ASSEMBLE
*STARTCΔPROGP,ASSEMBLE

//ΔLNKEDT
ΔPROGΔPROGA
ΔINCLUDEΔSYSUT1(PROGA)
ΔPROGΔPROGM
ΔINCLUDEΔSYSUT1(PROGM)
ΔPROGΔPROGP
ΔINCLUDEΔSYSUT1(PROGP)
//ΔEXECΔPROGM,UT2
//ΔENDMON
```

Device Assignments

◆ Under Executive or Monitor Control

SDN	Device Type	Remarks
SYSIPT	Card reader (see note).	Parameter and source statement input.
MERGE1 to MERGE8	Magnetic tape.	Source Library input for Copy or Merge functions. The highest numbered input has the highest priority in the case of duplicate program names.
XTAPE1 to XTAPE8	Magnetic tape.	Source Library input for Extract functions. The highest numbered input has the highest priority in the case of duplicate program names. Extract tapes have a higher priority than Merge tapes.
SYSUT5	Magnetic tape.	Source Library output.
OPTOUT	Magnetic tape.	Optional Source Library output.
SYSOPT.	Card punch (see note).	Source program card output.
SYSLST	Printer (see note).	Listing output.

Note:

When running under Monitor, magnetic tape may be substituted for SYSIPT, SYSOPT, and SYSLST.

DSPLY Parameter (Cont'd)

Notes:

1. This parameter must immediately precede the END parameter.
2. If a module to be displayed has been renamed, the new name must be used in the parameter card.

Examples:

```
ΔDSPLYΔALL
ΔDSPLYIΔALL
ΔDSPLYΔPAYROL
ΔDSPLYIΔBAKER
```

END Parameter

◆ This parameter is mandatory and must be used to signify the end of input parameters for the OMLU routine.

Format:

```
ΔEND
```

Parameter Examples for OMLU Functions

◆ 1. Creating an initial Call Library Tape that contains an Object Module Library section only.

a. Where OMF's appear on SYSIPT and SYSIPT is assigned to the card reader.

```
ΔCOPYΔNONE
ΔCATALOΔSYSIPT
(OMF decks inserted here)
ΔEND
```

b. Where OMF's appear on multiple input tapes:

```
ΔCOPYΔNONE
ΔCATALOΔSYSUT1  )
ΔCATALOΔSYSUT6  }     As Required
ΔCATALOΔSYSUT7  )
ΔEND
```

Parameter Examples for OMLU Functions *(Cont'd)*

2. Creating an initial Object Module Library section for an existing Call Library Tape:

   a. Where OMF's appear on SYSIPT and the Call Library Tape appears on SYSLIB:

   ```
   ΔCATALOΔSYSIPT

   ΔEND
   ```

   b. Where OMF's appear on multiple input tapes and the Call Library Tape appears on SYSLIB:

   ```
   ΔCATALOΔSYSUT1  ⎫
                   ⎪
   ΔCATALOΔSYSUT6  ⎬   As Required
                   ⎪
   ΔCATALOΔSYSUT7  ⎭

   ΔEND
   ```

3. Generating a Call Library Tape that contains the OML section from SYSLIB and the Assembly Macro Library and Executive Library sections from SYSUT4:

   ```
   ΔCOPYΔSYSUT4,AL,EL

   ΔMERGEΔ1

   ΔEND
   ```

4. Generating a Call Library Tape that contains the Assembly Macro Library section from SYSUT4 and object modules from SYSLIB and SYSUT4:

   ```
   ΔCOPYΔSYSUT4,AL

   ΔMERGEΔ2

   ΔEND
   ```

5. Generating an initial OML section by extracting input modules from various OML's:

   a. Where OML's appear on SYSUT1 and SYSUT6 and other libraries on SYSLIB:

   ```
   ΔEXTRACTΔSYSUT1,JOB

   ΔEXTRACTΔSYSUT6,CREDIT

   ΔEND
   ```

   b. Where OML's appear on SYSUT6 and SYSUT7 and other libraries on SYSUT4:

   ```
   ΔCOPYΔSYSUT4,AL,EL

   ΔEXTRACTΔSYSUT6,ONE

   ΔEXTRACTΔSYSUT6,TWO

   ΔEXTRACTΔSYSUT7,SEVEN

   ΔEND
   ```

Parameter Examples for OMLU Functions *(Cont'd)*

6. Adding object module files to an existing OML section:

   a. Where OMF's appear on SYSIPT and the existing OML on SYSLIB.

   ```
   ΛMERGEΛ1
   ΛCATALOΛSYSIPT
   ΛEND
   ```

   b. Where OMF's appear on SYSUT1 and existing OML's on SYSLIB and SYSUT4:

   ```
   ΛMERGEΛ2
   ΛCATALOΛSYSUT1,JOE
   ΛCATALOΛSYSUT1,SAM
   ΛEND
   ```

7. Merging an OML with other libraries, deleting two object modules and renaming another object module.

   The OML appears on SYSLIB and the other libraries on SYSUT4:

   ```
   ΛCOPYΛSYSUT4,AL,EL
   ΛMERGEΛ1
   ΛRENAMEΛSYSLIB,JOB03,JOB33
   ΛDELETOΛJOB22
   ΛDELETOΛJOB23
   ΛEND
   ```

8. Generating a duplicate copy of a Call Library Tape and displaying an index of the modules appearing in the OML section:

   a. Where Call Library Tape appears on SYSLIB.

   ```
   ΛMERGEΛ1
   ΛDSPLYIΛALL
   ΛEND
   ```

   b. Where OML appears is on SYSLIB and other libraries on SYSUT1.

   ```
   ΛCOPYΛSYSUT1,AL,EL
   ΛMERGEΛ1
   ΛDSPLYIΛALL
   ΛEND
   ```

Device Assignments

◆ When operating under Monitor, SYSIPT must be assigned to the device that contains the job stream OMLU parameters. When running under Executive control, SYSIPT must be assigned to the card reader and SYSLST to the printer.

If seven-level tapes are used, the pack/unpack mode must be set ON when the devices are assigned.

Table 6-1 shows the device assignments for various OMLU options.

Table 6-1. Device Assignments

Options / Devices	Display	Generating an initial OMF or updating an existing OML		Device Type
		No. OMF's	OMF's	
SYSIPT (Sys. input)	X	X	X	Card Reader*
SYSLST (Sys. output)	X	X	X	Printer**
SYSLIB (input)	X	M1	M1	Magnetic Tape
SYSUT1 (input)		E	EA	Magnetic Tape
SYSUT2 (output)		X	X	Magnetic Tape
SYSUT3 (work)			X	Magnetic Tape
SYSUT4 (input)		M2	M2	Magnetic Tape
SYSUT5 (input)		M3	M3	Magnetic Tape
SYSUT6 (input)		E	EA	Magnetic Tape

X = always required.

M1 = required whenever an existing OML is being updated.

M2 = required when merging two OML's.

M3 = required when merging three OML's.

E = required when object modules are to be extracted from an OML mounted on this device.

EA = optional. May contain input OMF or OML depending upon type of parameters used.

---

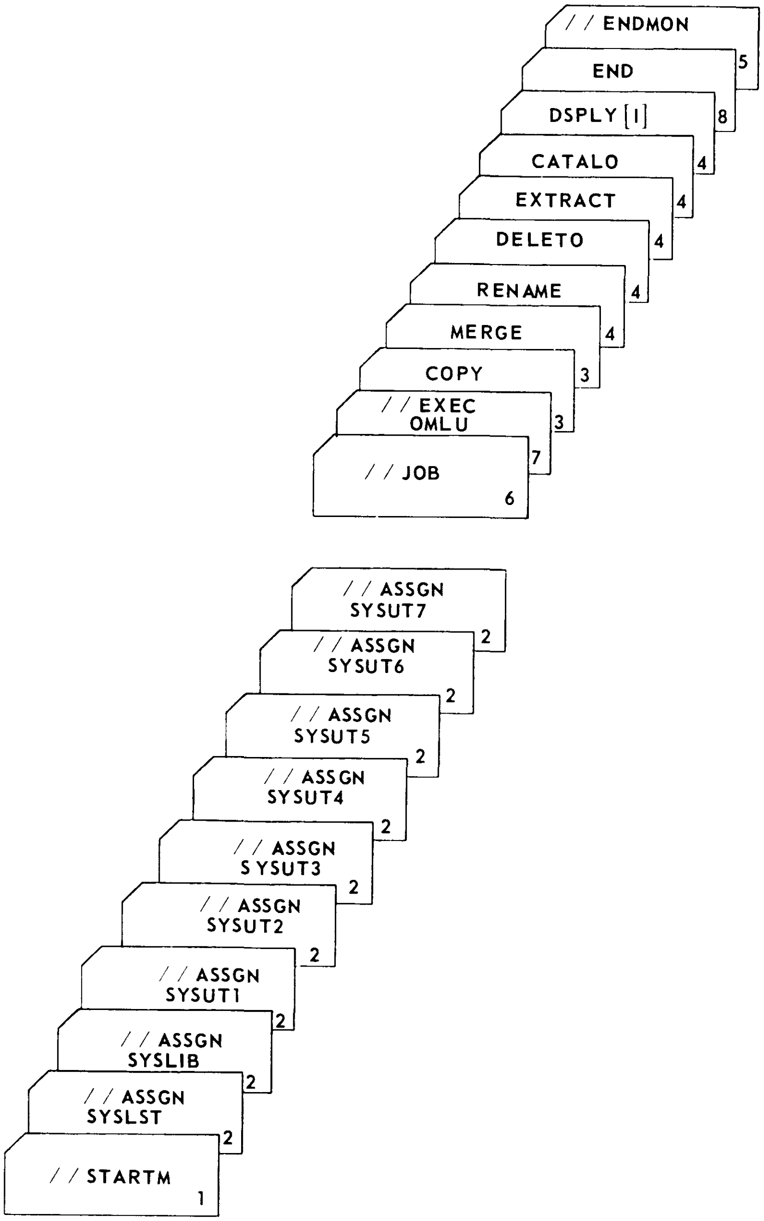
*A magnetic tape device can be assigned in place of card reader if parameters appear on tape and the routine is operating under Monitor control.

**A magnetic tape device can be assigned in place of the printer if displays are to be printed off-line and the routine is operating under Monitor control.

Job Stream Sequence

◆ Parameters identified by the numbers 1-8 would be present under the following conditions:

1. If OMLU is the first job in the stream and operating under Monitor.
2. If not already assigned.
3. Optional parameter; only one per program.
4. Optional Parameter; may have many of each type.
5. If OMLU is the last job under Monitor.
6. Replaced by //\END if not operating under Monitor.
7. Used only when operating under Monitor.
8. Required only when not operating under Monitor.



MACRO LIBRARY UPDATE (MLU)

General Description

◆ The Macro Library Update (MLU) is used to create or maintain an Assembly Macro Library section of a Call Library tape, and to copy or delete other libraries on the Call Library Tape. For a detailed description of the library tape, refer to Appendix B.

The routine is designed to operate under either TOS or TDOS, and either Monitor or Executive Control.

Preset Functions

None.

Optional Functions

The following functions may be performed as specified by routine parameters provided at run time:

1. To create an Assembly Macro library from card or magnetic tape input.
2. To add, delete, or replace entire macros.
3. To add, delete, or replace statements within specified macros.
4. To print and/or punch all or selected macros within a priority.
5. To print a record of corrections being applied to specific macros. (Always produced if corrections are applied).
6. To print a directory of all macros contained on the output Assembly Macro library. (Always produced if output library is produced.)
7. To merge and/or extract macros.
8. To move a macro from one priority to another.
9. To provide an abstract of the macros.
10. To merge, delete, and/or extract entire libraries.

Input

◆ Inputs to the MLU routine consist of control parameters, up to four Call Library Tapes, and macro source statements.

Output

◆ Output may be any combination of:

1. A Call Library Tape consisting of the Assembly Macro library alone or with any or all of the other libraries.

2. Printer reports
    a. Control statement listing with diagnostics.
    b. Environmental map.
    c. Macro Directory listing of output library.
    d. Source Statement Corrections listing.
    e. Macro listing (optional).
    f. Abstract listing (optional).

3. Punched macro decks.

4. Console typewriter messages

Equipment Configuration

Required

◆ Processor (65K).

Console typewriter.

Magnetic tapes (at least one in addition to SYSRES).

Card reader.

Printer.

Optional

◆ Up to six additional magnetic tape units.

Card punch.

When running under Monitor, magnetic tape may be substituted for the card reader, card punch, and printer.

Routine Parameters - General

◆ Following are the parameters for the MLU at least one of which must be used in addition to the END parameter.

ENVIRON Parameter

This parameter is used to alter preset options and to modify priority of copying the other Call Libraries.

Routine Parameters - General *(Cont'd)*

CREATS Parameter

This parameter is used to create an initial Assembly Macro Library section.

CATALS Parameter

This parameter is used to add assembly macros to the library.

EXTRACT Parameter

This parameter is used to extract macros from call library tapes designated as extract, and to change priority.

DELETS Parameter

This parameter is used to delete assembly macros.

STARTC Parameter

This parameter is used to delete, add, or replace individual macro statements.

CDELET Parameter

This parameter is used to delete macro statements in an assembly macro.

DSPLY Parameter

This parameter is used to print all or selected macros in the Assembly Macro Library.

PUNCH Parameter

This parameter is used to punch all or selected macros in the Assembly Macro Library.

DSPCH Parameter

This parameter is used to print and punch all or selected macros in the Assembly Macro Library.

END Parameter

The END parameter is used to signify the end of the parameters.

Routine Parameters - Detailed

ENVIRON Parameter

◆ The ENVIRON parameter serves two functions; to modify preset Assembly Macro Library processing and to modify priority of copying the other Call Library libraries.

1. If the ENVIRON MACRO parameter is not used, the following are assumed:
    - a. One input merge library.
    - b. One output library.
    - c. No abstract function.
    - d. No extract function.

    If multiple merge, extract, punch, and/or print only or abstracts are required the ENVIRON MACRO card is required and must be the first control card.

    Format

    ΔENVIRONΔMACRO,M= $\left\{\begin{matrix}(a,b,c,d)\\ x\end{matrix}\right\}$,E= $\left\{\begin{matrix}(a,b,c,d)\\ x\end{matrix}\right\}$

    ,LET=NONE,ABS,NOPT

Entry	Meaning
ΔENVIRONΔ	Control statement identifier.
MACRO	Specifies modification of Assembly Macro Library processing.
,M=(a,b,c,d) or ,M=x	Optional entry to specify extract tape and assign priority from low to high. Permissible values are 1 to 4 to correspond to INPUT1 to INPUT4. Optional entry to designate single merge tape. Values are 1 to 4.
,E=(a,b,c,d) or ,E=x	Optional entry to specify extract tape and assign priority. This entry must be present to use the EXTRACT function. Values are the same as above, with the restriction the same tape may not be designated both merge and extract.
,LET=NONE	Optional entry. Specifies that run is to type message indicated by COMM (see page 6-20) and terminate if any error condition occurs.
,ABS	Optional entry. Used to produce an abstract in addition to the directory listing. (See page 6-20G for example of Abstract listing.)
,NOPT	Optional entry. Specifies no tape output. Only punch and list functions will be performed.

ENVIRON Parameter (Cont'd)

Examples

ΔENVIRONΔMACRO, M=(1, 4, 2), E=3, ABS

ΔENVIRONΔMACRO, M=1, LET=NONE

ΔENVIRONΔMACRO, M=4, NOPT

Multiple ENVIRONΔMACRO parameters are permitted.

2. The second form of the ENVIRON card modifies the preset functions for copying COBOL, OBJECT MODULE, and EXECUTIVE libraries. If these cards are not used the libraries are copied from the highest priority merge tape on which each appears. When used the ENVIRON library-name cards must follow all other control cards except the END card. Other cards between ENVIRON library-name and END are ignored.

Format

$$\Delta\text{ENVIRON}\Delta \quad \left\{\begin{matrix}\text{COBOL}\\ \text{OBJECT}\\ \text{EXEC}\end{matrix}\right\} \quad , \quad \left\{\begin{matrix}\text{M=x}\\ \text{DELETE}\end{matrix}\right\}$$

Entry	**Meaning**
ΔENVIRONΔ	Control Statement identifier.
COBOL or OBJECT or EXEC	Identifies library to have modified processing.
, M=x or , DELETE	Identifies source of library, INPUT1 to INPUT4, according to value of x. Indicates that library is NOT to appear on output Call library.

Examples

ΔENVIRONΔCOBOL, M=1

ΔENVIRONΔEXEC, DELETE

One ENVIRON library-name card is permitted for each library.

ENVIRON Parameter (Cont'd)

COMM Statement

This optional statement is used in conjunction with LET=NONE option of the ENVIRON MACRO control to indicate the message to be typed upon error condition.

Format

ΔCOMMΔ any user message

Multiple COMM statements will not be flagged; however, only the last will be used. If the LET=NONE option is not used, the COMM statement will be ignored. If the COMM statement is used, it must immediately follow the ENVIRON MACRO parameter.

Examples

```
ΔENVIRONΔMACRO,M=(2,3),LET=NONE

ΔCOMMΔERROR=TERMINATION
```

CREATS Parameter

◆ This parameter indicates that an initial Assembly Macro Library is to be created. Its use nullifies STARTC, DELETS, and EXTRACT functions. Macros on input tapes will not be processed when CREATS is specified. The CREATS parameter must immediately follow the ENVIRON.

Format

ΔCREATSΔAL

CATALS Parameter

◆ The CATALS parameter specifies that the following macro is to be cataloged into the Assembly Macro Library. A macro may not be cataloged into a priority if the name already exists within the priority. The existing macro must first be DELETED and then cataloged. The macro being cataloged should contain a header statement (MACRO) and a trailer statement (MEND). A CATALS parameter must precede each macro being cataloged.

Format

ΔCATALSΔAx,name,SEQNCE,SAVE,AUTO

CATALS Parameter (Cont'd)

Entry	Meaning
ΔCATALSΔ	Identifier.
Ax	Priority indicator. x = 1 to 4 corresponding to priority into which macro is to be cataloged.
,name	Name of macro being cataloged, one to five characters.
,SEQNCE	Optional entry; if present, input sequence is checked on columns 73-80.
,SAVE	Optional entry to retain sequence numbers of input. Also causes sequence check.
,AUTO	Optional entry to retain flowcharting Comment cards on Macro Library. A period and an asterisk (.*) in columns 1 and 2 identify a flowcharting comment.

Examples

ΔCATALSΔA1,ALICE,SEQNCE

ΔCATALSΔA4,BETTY,SAVE,AUTO

EXTRACT Parameter

◆ The EXTRACT parameter is used to copy specific macros from a specified Extract tape into designated priorities on the output macro library. If the named macro does not appear on the designated tape, an error flag will be generated. Macros that are EXTRACTED have priority over merges.

Format

ΔEXTRACTΔINPUTx,(a,name,b),...,(a,name,b)

Entry	Meaning
ΔEXTRACTΔ	Identifier.
INPUTx	Indicates input from which macro is to be extracted. x may range from 1 to 4.
,(a,name,b)	At least one entry required; multiple entries are permitted. a=the priority into which macro "name" is to be placed. b=the priority from which macro "name is to be taken.

EXTRACT Parameter (Cont'd)

Notes

1. Tape designated by INPUT operand must have been designated by E=operand of ENVIRON MACRO parameter.
2. Multiple EXTRACT parameters are permitted.

Examples

ΔENVIRONΔMACRO,M=1,E=(3,4)

ΔEXTRACTΔINPUT3,(1,BETTY,1),(2,DORIS,1)

ΔEXTRACTΔINPUT4,(2,ALICE,2)

DELETS Parameter

◆ The DELETS parameter can be used to inhibit the copying of specified macros from a designated input, thus using the macro on a lower priority merge tape, or to inhibit writing specified macros to the output.

Format

ΔDELETSΔ $\left\{\begin{matrix}\text{INPUTx}\\ \text{OUTPUT}\end{matrix}\right\}$,Ax,name,...,name

Entry	**Meaning**
ΔDELETSΔ	Identifier.
INPUTx or OUTPUT	Determines whether macro is to be deleted from specified input merge tape or from output. x may range from 1 to 4.
,Ax	Priority from which macro is to be deleted. x indicates priority 1 to 4.
,name	Name of macro to be deleted (one to five characters). Multiple name entries are permitted.

Note

Input tape must be designated in the M=operand of the ENVIRONΔMACRO parameter.

Examples

ΔENVIRONΔMACRO,M=(1,4)

ΔDELETSΔINPUT4,A4,DTFEN

ΔDELETSΔOUTPUT,A2,CAROL

STARTC Parameter

◆ The STARTC parameter is used to initiate changes to an existing macro. It is used in conjunction with CDELET.

Updates are standard format MACRO source cards. If the sequence number (cols. 73-80) of the update card match the sequence number of a macro statement, the new statement replaces the old on the output. If the sequence number does not match, the statement is added. Update cards with blank sequence fields are inserted immediately following the card with a sequence number. A sequence number is required on the first update card, unless it is to precede the first existing statement. Any number of updates may follow a STARTC parameter.

Format

ΔSTARTCΔAx,name,SAVE

Entry	Meaning
ΔSTARTCΔ	Identifier.
Ax	Priority indicator. x may range from 1 to 4 to indicate priority.
,name	Name of macro to be changed (one to five characters).
,SAVE	Optional entry used to retain existing sequence numbers.

Notes

1. One STARTC parameter is required for each macro to be corrected.
2. The STARTC parameter automatically produces a correction listing. (See page 6-26E.)
3. If the SAVE option is not selected the output macro will be renumbered in increments of 100. The MEND card will have sequence number 99999999 in every case.

Example

ΔSTARTCΔA4,OWNER,SAVE

CDELET Parameter

◆ The CDELET parameter, if used, must follow a STARTC parameter. It is used to delete one or a series of macro statements. Any number of CDELET parameters may be used with one STARTC.

Format

ΔCDELETΔxxxxxxxx,yyyyyyyy

Entry	Meaning
ΔCDELETΔ	Identifier.
xxxxxxxx	a. Sequence number of macro statement to be deleted if this is only operand. b. Lowest sequence of a series of statements to be deleted if followed by second operand.
,yyyyyyyy	Optional entry. Last of a series of macro statements to be deleted.

Note

If only the first operand appears and no matching sequence number is present an error flag will be generated. If both operands are used an error flag will occur only if no sequence number within the range is present.

Examples

ΔCDELETΔSNAP0007,SNAP0013

ΔCDELETΔ0000700

DSPLY Parameter

◆ This parameter will cause MLU to produce a listing of all or selected macros in a priority section of the output Assembly Macro Library.

Format

$$\text{ΔDSPLYΔAx,} \begin{Bmatrix} \text{*ALL} \\ \text{name1,name2..,namen} \end{Bmatrix}$$

DSPLY Parameter (Cont'd)

Entry	Meaning
ΔDSPLYΔ	Identifier.
Ax	Priority indicator. x may range from 1 to 4.
,*ALL or ,name1	To display all macros in specified priority. To list specific named macro.

Notes

1. Multiple name entries maybe specified or multiple DSPLY parameters may be used.
2. If named macro is not present, or if it is a duplicate name, error flags will be generated.

Examples

ΔDSPLYΔA1,*ALL

ΔDSPLYΔA4,DTFEN,OWN

PUNCH Parameter

◆ This parameter is used to punch all or selected macros within a priority.

Format

$$\Delta\text{PUNCH}\Delta\text{Ax} \left\{ \begin{array}{l} \text{,*ALL} \\ \text{,name1,name2..,namen} \end{array} \right\}$$

where operands signify the same as DSPLY parameter.

DSPCH Parameter

◆ This parameter combines the functions of DSPLY and PUNCH.

Format

$$\Delta\text{DSPCH}\Delta\text{Ax.} \left\{ \begin{array}{l} \text{,*ALL} \\ \text{,name1,name2..,namen} \end{array} \right\}$$

where operands signify the same as DSPLY parameter.

END Parameter

◆ The END parameter signifies the end of input to MLU. This parameter is not required when running under Monitor.

Format

ΔENDΔ

Examples *(Cont'd)*

2. To produce a Call Library from the Output CLT of SYSGEN with modifications.

   a. No Executive Library.

   b. COBOL Library and User Macros from LIBA.

   c. Add macro OWNA in priority 3.

   d. Modify macro OWNB priority 2.

   e. Replace macro OWNC priority 1.

   f. Delete SNAPS from output priority 3.

   g. Abstract updated library.

```
// ASSGN SYSIPT,R1  (Control, changes)
// ASSGN SYSLST,L1
// ASSGN INPUT1,01  (LIBA)
// ASSGN INPUT2,02  (Output CLT from SYSGEN)
// ASSGN OUTPUT,03
// ASSGN WORK,04
// END
```

```
 1. ΔENVIRONΔMACRO,M=(1,2),ABS
 2. ΔDELETSΔINPUT1,A1,OWNC
 3. ΔCATALSΔA1,OWNC,AUTO
 4. (New Macro OWNC)
 5. ΔSTARTCΔA2,OWNB
 6. ΔCDELETΔ00001000,00001300
 7. (New statements)
 8. ΔCATALSΔA3,OWNA,SAVE
 9. (New macro OWNA)
10. ΔDELETSΔOUTPUT,A3,SNAPS
11. ΔENVIRONΔCOBOL,M=1
12. ΔENVIRONΔEXEC,DELETE
13. ΔEND
```

Examples
(Cont'd)

The above parameters will produce the following:

1. Merge tapes on INPUT1 and INPUT2, eliminating macros on INPUT1 which have duplicates on INPUT2 (card 1);
2. Delete OWNC from INPUT1 to allow inserting new OWNC (2,3,4);
3. Apply deletions and corrections to OWNB (5,6,7);
4. Add OWNA to output and retain sequence numbers (8,9);
5. Delete generated macro SNAPS from output (10);
6. Copy COBOL library from low priority INPUT1 (11);
7. Delete Executive Library (12).

Device Assignments

◆ Executive or Monitor

SDN	Device	Remarks
SYSIPT	Card Reader*	Parameters, corrections, additions.
SYSLST	Printer*	Listings.
SYSOPT	Card Punch*	Punched Macros.
INPUT1 to INPUT4	Magnetic Tape	TOS Call Libraries either Merge or Extract.
OUTPUT	Magnetic Tape	Updated TOS Call Library.
WORK	Magnetic Tape	Work Tape.

*May be replaced by magnetic tape when operating under Monitor.

CONTROL STATEMENT LISTING

```
DATE    MM/DD/YY          MACRO LIBRARY UPDATE - VERSION XX                  PAGE XXXX

CONTROL STATEMENTS LISTING                                                   ERROR FLAGS

ENVIRON          M=(4,3,1),E=2,LET=NONE

DELETS           INPUT3,A1,NAMEA

EXTRACT          INPUT2,(A2,NAMEB,A3)

DSPLY            A3,NAMEB,NAMEA

STARTC           A1,NAMED,SAVE

STARTC           A3,NAMEA

END
```

MLU Output Examples

DEVICE ENVIRONMENT MAP OF CONTROL STATEMENTS

```
DATE     MM/DD/YY        DEVICE ENVIRONMENTAL MAP                             PAGE XXXX

DEVICE NAME        FILE HEADER        FILE DATE        TYPE                   ACTIONS REQUESTED

INPUT1             SYSLIB             YYDDD            MERGE        PRIOR     MACRO            DEL
                                                                      1       NAMEA            X
                                                                      1       NAMEB            E(1)

INPUT3             SYSLIB             YYDDD            MERGE

INPUT4             SYSLIB             YYDDD            EXTRACT      MACRO     DESTINATION                ORIGIN
                                                                    NAMEB          1                       3

OUTPUT             SYSLIB             YYDDD            OUTPUT       PRIOR     MACRO            DEL
                                                                      4       NAMEA            X

SYSLST                                                 OUTPUT       PRIOR     MACRO            PRT
                                                                      3       NAMEX            E(1)
                                                                      3       NAMEY            X
                                                                      4       *ALL             X

SYSOPT                                                 OUTPUT       PRIOR     MACRO            PCH
                                                                      3       NAMEY            X

SYSIPT                                                 INPUT        PRIOR     MACRO            UPDAT     CATAL
                                                                      1       NAMEM            X
                                                                      2       NAMEO                        X
```

(1) NOTE: E signifies that the action cannot be taken.
X signifies that the action will be taken.

MLU Output Examples *(Cont'd)*

```
                     MACRO AND MACRO CORRECTIONS LISTING

DATE     MM/DD/YY        MACRO LIBRARY - PRIORITY SECTION X                  PAGE XXXX

MACRO - NAMEA

LABEL    OPCODE     OPERAND     COMMENTS                 SEQ. NO.    REMARKS       ERR FLAGS

         MACRO      VER004      MM/DD/YY    MM/DD/YY     00000100

&A       NAMEA      &B                                   00000100

         CNOP       2,4                                  00000300    REPLACED

         CNOP       1,2                                  00000300

&A       SVC        11                                   00000500    DELETED

         DC         A(&B)                                00000400                  ERR 3617

         DC         '12'                                 00000600    INSERTED

         MEND                                            99999999    GENERATED
```

MLU Output Examples *(Cont'd)*

MACRO LIBRARY DIRECTORY LISTING

```
DATE     MM/DD/YY          MACRO LIBRARY DIRECTORY                             PAGE XXXX



                           PRIORITY X

NAMEA            NAMEB            NAMEC            NAMED

NAMEE            NAMEF
```

MLU Output Examples *(Cont'd)*

MACRO LIBRARY ABSTRACT LISTING

```
DATE     MM/DD/YY          MACRO LIBRARY ABSTRACT                           PAGE XXXX

MACRO NAME           PRIORITY         VERSION NUMBER         VERSION DATE         UPDATE DATE

NAMEA                    1                 003               MM/DD/YY             MM/DD/YY

NAMEB                    1                 002               MM/DD/YY             MM/DD/YY

NAMEC                    2                 003               MM/DD/YY             MM/DD/YY
```

MLU Output Examples *(Cont'd)*

Considerations for Use

◆ 1. The ENVIRONΔMACRO, if used, must be the first parameter encountered.

2. Macros to be cataloged require a MACRO header statement as the first statement and a MEND as the last statement. Recommended format of the MACRO header is:

Column	Content	Meaning
1-5	spaces	position
6-10	MACRO	identifier
11	space	
12-14	VER	operand identifier
15-17	nnn	three-digit version number
18	space	
19-26	MM/DD/YY	date of version
27-72	spaces	
73-80		sequence number/optional

If the above format is not followed the version number and creation date will not appear on the abstract listing.

3. Macros appearing on more than one source input will be copied to output first from EXTRACT source if specified, then from the highest priority MERGE tape.

Examples

◆ 1. To produce a new Call Library using RCA macros and libraries from CLT produced by SYSGEN and adding user macros from an existing Call Library.

```
// ASSGN SYSIPT,R1      (Control cards)
// ASSGN SYSLST,L1
// ASSGN INPUT1,01      (Existing Call Library)
// ASSGN INPUT2,02      (Output Call Library SYSGEN)
// ASSGN OUTPUT,03      (Updated Call Library)
// END
ΔENVIRONΔMACRO,M=(1,2)
ΔEND
```

The above will cause macros and libraries which appear on both inputs to be copied from INPUT2. Macros which do not appear on INPUT2 (that is, user Macros) will be copied from INPUT1. Errors would be flagged, but processing would continue.

LOAD LIBRARY UPDATE (LLU)

General Description

◆ The Load Library Update (LLU) routine is used to generate a load library tape or to update an existing load library by adding, deleting, replacing, or altering programs.

The following library tapes can be maintained by this routine: (See Appendix B for formats.)

System Load Library (SLL)

Program Load Library (PLL)

Executive Load Library (ELL)

Preset Functions

None.

Optional Functions

The following options may be selected by supplying appropriate parameter cards at execution time:

1. Creating or updating a System, Program, or Executive Load Library tape.
2. Selecting programs or complete libraries to be merged and copied to the output library. The programmer may rename programs and loads during the updating process. In addition, the contents of Program and Load Descriptor blocks can be changed and patches made to specific text areas within designated program loads.
3. A complete program listing or a library abstract can be requested of the updated library tape, or these listings can be obtained of an input library when no maintenance functions are performed.

Input

◆ Input to this routine consists of load library tapes (System Load Library, Program Load Library, or Executive Load Library), and control parameters on SYSIPT which specify the functions desired.

Input (Cont'd)

One or more of the following input tapes can be used (see COPY, MERGE, EXTRACT, and DISPLAY parameters to determine when each is required):

SYSUT1	LL to be merged.
SYSUT4	LL to be merged.
SYSUT5	LL to be merged.
SYSUT6	LL containing programs to be extracted.
SYSUT7	LL containing programs to be extracted.
SYSUT8	LL containing programs to be extracted.
SYSLIB	LL Source of Executive Library and overlays. If this device is not present, an input device specified by the COPY parameter must contain these routines.

Note:

SYSRES cannot be used as an input to this routine.

Output

◆ Output of this routine consists of a load library tape (System Load Library, Program Load Library, or Executive Load Library). A listing of programs on the library tape or an abstract of program and load descriptor blocks may be obtained as an option.

Equipment Configuration

Required

◆ Processor (65K).

Console typewriter.

Magnetic tape devices (three required).

Card reader, or Videoscan document reader with card read feature.

Printer.

Optional

◆ Magnetic tape devices (up to six additional) may be used as input devices.

Magnetic tapes can also be used in place of the card reader and the printer.

Note:

If seven-level tapes are used, the pack/unpack mode must be set through device assignments.

See table 6-2 for a listing of the devices required for various LLU options.

Routine Parameters - General

◆ Parameters are entered from SYSIPT and must be sequenced by type. All are optional, but at least one parameter other than END must be supplied to effect any result. The parameters used by this routine are summarized below.

COPY Parameter

This parameter specifies the type of library to be produced, its name, the creation date and version number, and frequency of interspersed Executive overlays (for SLL).

REORDER Parameter

This parameter specifies that an input device containing programs out-of-sequence is to be reordered by program name.

MERGE Parameter

This parameter defines an input device that contains a load library to be merged, copied, or updated onto the output tape.

RENAME Parameter

This parameter is used to change the name of an input program when the program is written to the output library.

DELETE Parameter

This parameter names a program on a merge input tape that is not to be included in the output.

EXTRACT Parameter

This parameter names a program to be copied to the output library when the program to be included does not appear on a merge input.

DSPLY Parameter

This parameter specifies the type of listings desired.

PATCH Parameter

This parameter is used to change program text, to alter load names and addresses, and to change the memory requirements portion of a program descriptor block.

END Parameter

This parameter signifies the end of input parameters.

Sequence of Parameters

◆ Although all parameters are optional, they must appear in the following order when supplied:

Creating a New Load Library Tape

COPY

REORDER

MERGE

RENAME

DELETE

EXTRACT

$\left\{\begin{array}{l}\text{DSPLY ALL}\\ \text{Specific DSPLY and PATCH}\end{array}\right\}$

END

Listing an Existing Load Library Tape

$\left\{\begin{array}{l}\text{DSPLY ALL}\\ \text{Specific DSPLY's}\end{array}\right\}$

END

Specific DSPLY and PATCH parameters must be supplied as a group. The sequence within this group is primarily by program name in alphanumeric order, and secondarily by load name in the sequence in which the loads are present within a program.

When the same program or load is to be both displayed and patched, the DSPLY parameter must appear first. For example:

```
COPY  SYSUT1
MERGE  1
DSPLYA  PROGA
PATCHP  PROGB
DSPLY  PROGC
PATCH  PROGC
PATCH  PROGD
DSPLY  PROGE
DSPLY  PROGF,LOADA
PATCH  PROGF,LOADA
DSPLY  PROGG
END
```

Any combination of parameters is valid as long as a COPY parameter is present; when a COPY is not used, only DSPLY parameters may be submitted.

Routine Parameters - Detailed

COPY Parameter

◆ This parameter designates the type of output library to be generated (SLL or PLL), the library name ,the creation date and version number, and the dispersion factor for Executive overlays. The COPY parameter is mandatory whenever any updating functions are performed.

Format:

$$\Delta\text{COPY}\Delta \left\{\begin{array}{l}\text{ssssss}\\ \text{NONE}\end{array}\right\} \text{,Nnnnnnn,} \left\{\begin{array}{l}\text{Dmm/dd/yy}\\ \text{Dyyddd}\end{array}\right\} \text{,Vvvv,Ffff}$$

Entry	Meaning
ΔCOPYΔ	Parameter identifier.
ssssss or NONE	ssssss = symbolic name of device from which Executive will be copied to generate a SLL. NONE = generate a PLL. If omitted, SYSLIB is assumed to contain the Executive and an SLL is generated.
,Nnnnnnn	nnnnnn = symbolic name for updated Load Library (up to six characters). If omitted, SYSLD is assigned for an SLL and PRGLD is assigned for a PLL.
,Dmm/dd/yy or ,Dyyddd	mm/dd/yy = creation date for SLL. yyddd = creation date for PLL. If omitted, the creation date is obtained from the standard date area.
,Vvvv	vvv = version number (001-999). If omitted, 001 is assumed.
,Ffff	fff = frequency of interspersed Executive overlays (000-999). If 000 is specified, the Executive is copied but overlays will not appear between programs. If 006 is specified, Executive overlays will be placed after every sixth program. If omitted, 001 is assumed and overlays will appear after <u>every</u> program.

COPY Parameter (Cont'd)

Notes:

1. All entries are optional. If an entry is omitted, a positional comma is not required.
2. If this parameter is omitted, a display only of the load library appearing on SYSUT1 is performed.

Examples of the COPY Parameter:

1. To produce a PLL, named PRGLD, with a version number of 001:

   ΔCOPYΔNONE

2. To produce a PLL, named PRLOAD, version 001, with a creation date of 1 January 1968:

   ΔCOPYΔNONE,NPRLOAD,D68001

3. To produce an SLL, named SYSLD, version 001, copying the Executive from SYSLIB and placing overlays after each program:

   ΔCOPY

4. To produce an SLL, named SYSLD, version 002, copying the Executive from SYSUT1 and placing overlays after every second program.

   ΔCOPYΔSYSUT1,V002,F002

5. To produce an SLL, named SYLOAD, version 001, creation date of 1 January 1968, copying the Executive from SYSLIB without interspersed overlays:

   ΔCOPYΔNSYLOAD,D01/01/68,F000

REORDER Parameter

◆ This parameter must be used for any input tape that contains programs that are not in ascending sequence. It informs the LLU that this tape is to be reordered into program-name sequence for LLU processing.

Format:

ΔREORDERΔSYSUTn,..., SYSUTn

Entry	Meaning
ΔREORDERΔ	Parameter identifier.
SYSUTn	n = 1,4,5,6,7 or 8 (LL device number to be reordered).

REORDER Parameter (Cont'd)

Notes:

1. When used, this parameter must immediately follow the COPY parameter.

2. When more than one symbolic device is specified, a comma must be used to separate each entry.

3. If an input not specified in this parameter is out of sequence, the routine terminates. (As all output must be in sequence by program name, the routine first scans the specified devices, building a table of program names, source devices, and relative position on the devices so that it can process these programs in sequence.)

Example:

ΔREORDERΔSYSUT1,SYSUT6,SYSUT8

MERGE Parameter

◆ This parameter must be used to inform the LLU routine that an input tape (or tapes) contains programs to be merged, updated, or copied onto the output tape.

Format:

ΔMERGEΔn

Entry	**Meaning**
ΔMERGEΔ	Parameter identifier.
n	n = 1 input LL is on SYSUT1. = 2 input LL's are on SYSUT1 and SYSUT4. = 3 input LL's are on SYSUT1, SYSUT4, and SYSUT5.

Notes:

1. If programs of the same name appear on different input tapes, only one version of the program is copied to the output tape. The version selected is determined by assigning SYSUT5 as the highest priority input, SYSUT4 as the next, and SYSUT1 as the lowest. Thus, if program XYZ appears on SYSUT1 and SYSUT5, only the version on SYSUT5 is placed on the output tape.

2. Only one MERGE parameter is permitted.

Example:

To merge programs from SYSUT1 and SYSUT4:

ΔMERGEΔ2

RENAME Parameter

◆ This parameter can be used to rename a program that appears on an input tape when that program is written to the output tape.

Format:

ΔRENAMEΔSYSUTn,aaaaaaaa,bbbbbbbb

Entry	Meaning
ΔRENAMEΔ	Parameter identifier.
SYSUTn	n = 1,4,5,6,7 or 8 (LL device containing program to be renamed.)
,aaaaaaaa	aaaaaa = old name (one to eight characters).
,bbbbbbbb	bbbbbb = new name (one to eight characters).

Note:

Multiple rename parameters may be used.

Examples:

ΔRENAMEΔSYSUT5,PROG8,PROGAA

ΔRENAMEΔSYSUT5,PROG9,PROGBB

DELETE Parameter

◆ This parameter can be used to specify that the program named is <u>not</u> to be placed on the output tape. It can only be used to delete programs that appear on merge tapes SYSUT1, SYSUT4, and SYSUT5.

Format:

ΔDELETEΔaaaaaa

Entry	Meaning
ΔDELETEΔ	Parameter identifier.
aaaaaa	aaaaaa = name of program to be deleted (one to six characters).

Notes:

1. This parameter can only be used in conjunction with the MERGE parameter.
2. This parameter overrides an EXTRACT parameter when the same program name is referenced.
3. Multiple Delete parameters may appear.

Examples:

ΔDELETEΔPROG8

ΔDELETEΔPROG22

EXTRACT Parameter

◆ This parameter is used to extract a particular program from an input tape for inclusion on the output tape. Programs to be extracted must appear on SYSUT6, SYSUT7, or SYSUT8.

Format:

ΔEXTRACTΔSYSUTn,aaaaaa

Entry	Meaning
ΔEXTRACTΔ	Parameter identifier.
SYSUTn	n = 6,7 or 8 (LL device from which program is to be extracted.)
,aaaaaa	aaaaaa = name of program to be extracted (one to six characters).

Notes:

1. If programs on the extract tape are not in sequence, a REORDER parameter must be used and precede all extract parameters that refer to this tape.
2. Multiple extract parameters may appear.

Example:

ΔEXTRACTΔSYSUT6,PROG5

DSPLY Parameter

◆ This parameter can be used to obtain a listing of all programs, selected programs, or individual program loads that appear on a load library.

When used with updating parameters (COPY, MERGE, etc.), the listing reflects the contents of the generated output tape. When used by itself, a listing of an input load library mounted on SYSUT1 is produced.

Two formats are provided: the DSPLY format provides a listing of the complete coding for a program or a program load, the DSPLYA format provides an abstract listing of program load and descriptor blocks.

Format for program listing:

$$\Delta\text{DSPLY}\Delta \begin{Bmatrix} \text{ALL} \\ \text{pppppp} \end{Bmatrix} \text{,llllll}$$

DSPLY Parameter (Cont'd)

Format for program abstract:

$$\Delta\text{DSPLYA}\Delta \left\{ \begin{matrix} \text{ALL} \\ \text{pppppp} \end{matrix} \right\} \text{,llllll}$$

Entry	Meaning
ΔDSPLYΔ or ΔDSPLYAΔ	Parameter identifier.
ALL or pppppp	ALL = display all programs on load library. pppppp = name (one to six characters) of specific program to be displayed.
,llllll	Name (one to six characters) of specific program load to be displayed. Note: If this entry is omitted, all program loads for the program specified are displayed.

Notes:

1. Program and load names may be one to six characters.
2. When an ALL entry is specified, other DSPLY or DSPLYA parameters cannot be used.
3. When this routine is used as a listing run only, SYSUT1 is assumed as the input device.
4. If a load entry is omitted, all loads within a specified program are displayed.

Examples:

ΔDSPLYΔALL

ΔDSPLYΔPROGA
ΔDSPLYΔPROGB,LOAD2
ΔDSPLYΔPROGB,LOAD4
ΔDSPLYΔPROGC

ΔDSPLYAΔALL

ΔDSPLYAΔPROGA
ΔDSPLYAΔPROGC

PATCH Parameters

◆ These parameters provide the facility to apply changes to a program or a program load before it is written to the output tape.

If a program being changed has been renamed, the patch must reference the new name; if a program load is being changed, the patch must reference the old name.

Three Patch parameters are provided:

PATCH	- used to apply changes to the text portion of a program load.
PATCHL	- used to change the name and loading address of a program load.
PATCHP	- used to change the starting address, version number, creation date, and memory requirements portions of a program descriptor block.

1. *PATCH Format:*

Card Columns	Content	Meaning
1-6	ΔPATCH	Parameter identifier.
7-12	pppppp	Name of program to be changed (one to six characters).
13-18	llllll	Name of load to be patched (one to six characters).
19-20	KM or KC	KM = patch information in columns 30-78 is in hexadecimal format; that is, two characters for each memory location to be changed. KC = patch information is in graphic format; that is, one character for each memory location to be changed.
21-26	hhhhhh	Left-hand end of program area to be patched. This must be a program-relative, hexadecimal address. Note: The first byte of a program is address 000000; the 11th byte is address 00000A.
27-28	nn	Number of columns in this card that contain patch information (01-49). If the KM format is used, nn must be an even number.
29		Blank.
30-78	x.....x	Patch information.

PATCH Parameters (Cont'd)

Notes:

1. Relocatable constants (DCA) in a program load cannot be modified by a patch card.
2. Multiple patches may be applied to the same load.
3. The maximum size of an area that can be patched by a single card is 49 locations (KC format) or 24 locations (KM format).
4. Patches are placed on the output tape following the last text block for the load. Each patch card creates a patch text block.

Example:

ΔPATCHSAMΔΔΔSECONDKC00055220NEWΔDATAΔFORΔPROGRAM

2. PATCHL Format:

ΔPATCHLΔpppppp,llllll,aaaaaa,bbbbbb

Entry	Meaning
ΔPATCHLΔ	Parameter identifier.
pppppp	Name of program to be changed (one to six characters).
,llllll	Name of program load to be changed (one to six characters).
,aaaaaa	Optional. New name of program load (one to six characters). Note: If this entry is omitted, a positional comma must be used.
,bbbbbb	Optional. New loading address of program load (six hexadecimal characters).

Examples:

ΔPATCHLΔJOE,(ROOT),,000700

ΔPATCHLΔJOE,OVLY03,OVLY08

PATCH Parameters (Cont'd)

3. *PATCHP Format:*

ΔPATCHPΔpppppp,ssssss,aaaaaa,bbbbbb,vvv,dddddd

Entry	**Meaning**
ΔPATCHPΔ	Parameter identifier.
pppppp	Name of program to be changed (one to six characters).
,ssssss	Optional. New starting address, or entry point, for program (expressed as a program-relative, six-character hexadecimal address).
,aaaaaa	Optional. New minimum memory requirements for program (six hexadecimal characters).
,bbbbbb	Optional. New maximum memory requirements for program (six hexadecimal characters).
,vvv	Optional. New version number of program (three characters, zero-filled).
,dddddd	Optional. New creation date of program (six characters).

Note:

A positional comma must be used to indicate the absence of an entry if a succeeding entry is used. See examples 1 and 3.

Examples:

ΔPATCHPΔJOE,,0040AC

ΔPATCHPΔSAM,000074,00238A,0026FF

ΔPATCHPΔHARRY,0004AA,,,001,121566

END Parameter

◆ This parameter is mandatory and denotes the end of input parameters for the LLU routine.

Format:

ΔEND

Considerations for Use

Structure and Organization of Routine

◆ The LLU operates under either Monitor or Executive control and is organized into two phases.

Phase I

The first phase constructs tables of information obtained while processing COPY, MERGE, REORDER, RENAME, EXTRACT, and DELETE parameters. These tables are used to determine the programs that are to be the output of Phase II.

Phase II

The second phase of the LLU processes the PATCH and DISPLAY parameters while copying programs from the input tapes and inserting the Executive overlays according to the information in the tables constructed by Phase I. This phase also produces the new load library along with any requested listings.

The LLU routine provides for a maximum of seven input tapes. Three of these may be designated as merge tapes; three may be designated as extract tapes. The remaining tape may be designated as a source from which to obtain Executive portions for an output SLL.

When the programmer designates an input as a merge tape, all of the programs on that tape (except those which are deleted by a DELETE parameter) will be transferred to the output tape. When the programmer designates an input as an extract tape, he must supply an EXTRACT parameter for each program that he wishes to transfer from that tape to his output load library.

The LLU routine assumes that the programs on all input library tapes are recorded in ascending alphanumeric sequence. If one or more of the inputs are not in order, the user must establish the proper sequence prior to processing by submitting a REORDER parameter.

Names of Programs and Program Loads

◆ The program name entry for an LLU parameter will normally be the name used in the PROG statement for the program when it was processed through the Linkage Editor. If a PROG statement was not present, the name used in the START card of the first module of the root segment is used.

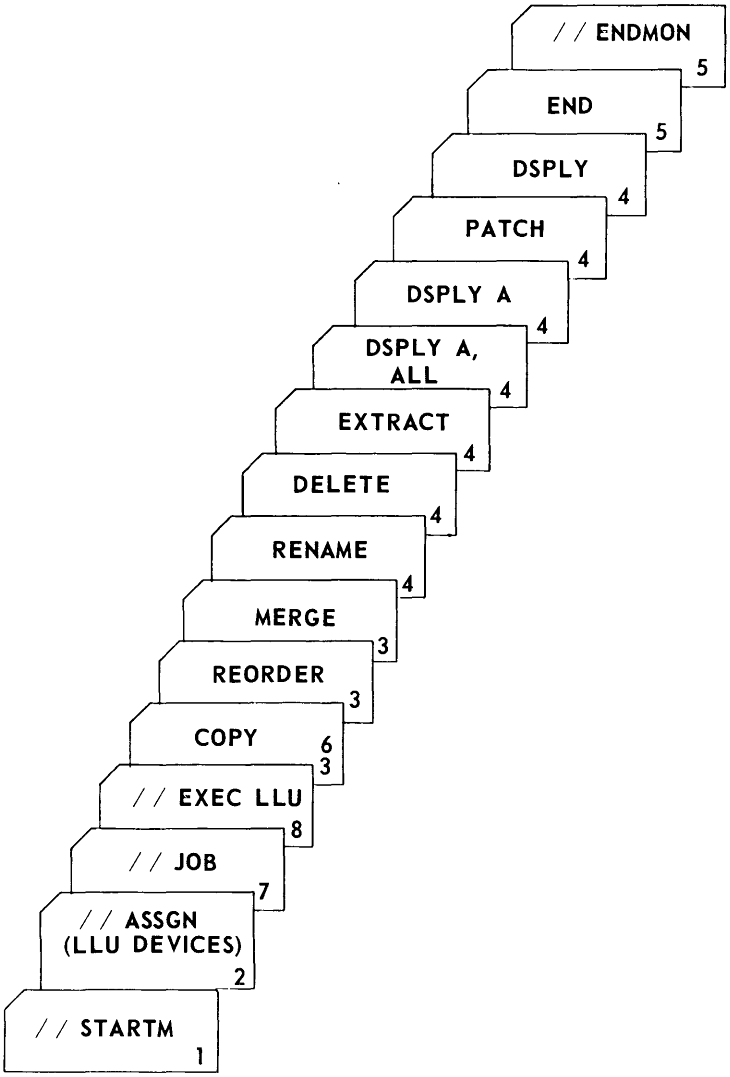
For program loads, the root load for all programs is referenced by the name "(ROOT)". The name of an overlay load depends on what name was assigned in the OVERLAY statement at Linkage Editor time.

The Linkage Editor program listings should always be consulted when in doubt.

Job Stream Sequence

◆ The parameters identified by 1-5 are present under the following conditions:

1. If the LLU is the first job in a Monitor stream.
2. If not already assigned.
3. Optional parameters; only one of type per program.
4. Optional parameters; may have many of each type per program.
5. If the LLU is the last job in a Monitor stream.
6. Must be present when any updating function is performed.
7. Replaced by //Δ END if not operating under Monitor.
8. Invalid if not operating under Monitor.



Parameter Examples

◆ 1. Copy a PLLT and obtain an abstract of the output.

```
// ASSGN SYSIPT, R1
// ASSGN SYSLST, L1
// ASSGN SYSUT1, 01
// ASSGN SYSUT2, 02
// END
ΔCOPY NONE
ΔMERGE 1
ΔDSPLYA ALL
Δ END
```

2. Merge two PLLT's and create an SLLT.

```
// ASSGN SYSIPT, R1
// ASSGN SYSUT1, 01
// ASSGN SYSUT2, 02
// ASSGN SYSUT4, 03
// ASSGN SYSLIB, 04
// END
ΔCOPY NTOSSLT, V001, F005
ΔMERGE 2
ΔEND
```

3. Add selected programs to, and delete selected programs from a PLLT.

```
// ASSGN SYSIPT, R1
// ASSGN SYSUT1, 01
// ASSGN SYSUT2, 02
// ASSGN SYSUT6, 03
// ASSGN SYSUT7, 04
// END
Δ COPY NONE
ΔMERGE 1
ΔDELETE PROGH
ΔDELETE PROGL
ΔEXTRACT SYSUT6, PROGD
ΔEXTRACT SYSUT6, PROGM
ΔEXTRACT SYSUT7, PROGQ
ΔEND
```

Device Assignments

SDN	Device Type	Remarks
SYSIPT	Card reader.*	Input parameters.
SYSLST	Printer.*	Output listings.
SYSLIB	Magnetic tape.	Contains Executive library when SLL is generated.
SYSUT1	Magnetic Tape.	Input tape required for any merge, update, or copy.
SYSUT2	Magnetic tape.	Output tape (SLL or PLL).
SYSUT3	Magnetic tape.	Work tape.
SYSUT4	Magnetic tape.	Merge input tape.
SYSUT5	Magnetic tape.	Merge input tape.
SYSUT6	Magnetic tape.	Extract input tape.
SYSUT7	Magnetic tape.	Extract input tape.
SYSUT8	Magnetic tape.	Extract input tape.

*The device type under Monitor can be magnetic tape.

Table 6-2 summarizes the required and optional devices used with this routine.

Table 6-2. LLU Routine Device Options

Options / Devices	Display Only	Generate Exec LL	General Program LL	Generate System LL
SYSIPT	X	X	X	X
SYSLST	X	X	X	X
SYSLIB (INPUT)		O		L
SYSUT1 (LL INPUT)	X	O	M1	M1
SYSUT2 (OUTPUT)		X	X	X
SYSUT3 (WORK)				W
SYSUT4 (LL INPUT)		O	M2	M2
SYSUT5 (LL INPUT)		O	M3	M3
SYSUT6 (LL INPUT)		O	E	E
SYSUT7 (LL INPUT)		O	E	E
SYSUT8 (LL INPUT)		O	E	E

X = Required.
O = Optional.
L = Required, unless the EXEC library appears on another input.
W = Required if the EXEC library appears on an input other than SYSLIB.
M1 = Required for one-way merge.
M2 = Required for two-way merge.
M3 = Required for three-way merge.
E = Optional. Only required if extracting programs from this device.

LINKAGE EDITOR (LNKEDT)

General Description

◆ The Linkage Editor (LNKEDT) routine establishes communication and linkage among modules of a program and converts the program from its language translator output card image format to the operating system's load library format. Thus, the use of this routine is an essential step in the progression from a source language program to an executable, machine-language program.

A program can consist of one or more object modules produced by the language translators. In the process of converting the program to load library format, the Linkage Editor also performs diagnostic checks on the structure and content of the program to ensure that it will be capable of execution under TOS.

Preset Functions

All output (object modules) from the language translators (Assembler, RPG, COBOL, FORTRAN, Sort/Merge Generator) must be converted by this routine to load library format. If a program consists of a single object module with no overlays, it can be processed by the Linkage Editor with no control statements required. In this case, the Linkage Editor simply produces a Program Load Library Tape in the format described in Appendix B. The bound program may then be loaded and executed by the Executive system.

Optional Functions

1. System Load Library Tape.

   The Linkage Editor may optionally be directed to produce a System Load Library Tape (SLLT) which contains a complete Executive System. This tape is self-initializing and can be used to load and execute the program without the need for an additional Executive System tape to be mounted.

2. Selection of Object Modules

   In addition to binding object module decks that appear in the input stream, modules may be extracted from a magnetic tape that contains Object Module Files (OMF's) or an Object Module Library (OML). The programmer may indicate by means of an INCLUDE statement that a specific object module (or an entire OMF) is to be included.

   Object modules that appear in the input stream, or that are selected from an OMF or OML as above, are referred to as explicity bound modules.

General Description *(Cont'd)*

3. Automatic Library Call Mechanism

If an external reference (EXTRN) in an object module being bound cannot be satisfied by an entry point in an explicitly-bound object module, the Linkage Editor assumes that an object module with this entry point exists on an OML. Based on this assumption, SYSLIB (and, optionally, additional OML's) is then searched for a module with a matching entry point. If such a module is found, it is bound into the root load. This type of module selection is known as implicit binding.

If desired, all implicit binding may be suppressed by use of the NCAL statement, or specific external references may be prevented from being implicitly bound by using the LIBRARY control statement. (See section on detailed parameters for a description of these control statements.)

4. Overlay Function

To use the overlay function, the programmer should understand the following terms:

a. Segment - the smallest functional unit that can be loaded as a logical entity. There are two types of segments: root and overlay. The root segment is part of every program and remains in memory throughout program execution. Overlay segments are brought into memory as needed and may or may not be required for program execution.

b. Path - consists of a segment and all segments in the same region between it and the root segment. The root segment is a part of every path in every region.

c. Region - a contiguous area of memory within which segments can be loaded independently of paths in other regions. An overlay program can use single or multiple regions.

The Linkage Editor can produce a single load program (root segment only) or a program utilizing overlays.

Either single region or multiple region overlays can be provided. A load in one region can communicate with a load in another region. See the OVERLAY parameter for a description of how overlay segments are defined.

5. Reservation of Common Storage Areas

For FORTRAN programs, the programmer may reserve common storage areas that do not contain data or instructions. Information about the control sections is taken from the ESD card output of a FORTRAN program. If two or more common areas have the same name, a single area is reserved which is equal to the size of the larger of the common areas.

Input

◆ There are two types of input to the Linkage Editor:

Object Modules

Control Statements

An object module is the output from a language translator run. It may be an entire program or a logical section of a program which was designed using the modular concept of programming. Object modules may also be submitted as magnetic tape files in either OMF or OML format.

Control statements are used to direct the Linkage Editor as to what functions to perform. They may be submitted through the Monitor job stream on SYSIPT, or from SYSUT1.

Primary Data Set

The primary data set is the main input to the Linkage Editor routine. Modules in this data set are processed sequentially and bound into the output program in the order in which they are submitted. (INCLUDE statements in the primary data set may cause a temporary switch to secondary input, but control always returns to the primary input.) Primary input terminates when the end of the data set is reached; however, if unresolved external references exist, and the automatic call library mechanism is <u>not</u> inhibited, an attempt will be made to resolve these references.

The Linkage Editor can accept only one primary data set, which is normally on SYSIPT, unless informed by the Monitor that SYSUT1 is the primary input source.

The primary data set may not contain INCLUDE statements to an object module in itself.

Secondary Data Set

A secondary data set contains an OMF or OML from which a module(s) is bound into the output program by use of INCLUDE statements found in the primary input. An entire secondary set data or any object module in it may be included. The secondary data set may contain INCLUDE statements to OML's but not to the same or any other data set.

For an example of the use of primary and secondary inputs, see pages 6-55 and 6-56.

Output

◆ The standard output of this routine is a Program Load Library Tape (PLLT) containing the programs bound within the session, and a map showing the bound loads and the modules within each load.

Optional output may be a System Load Library Tape (SLLT). This tape contains the Executive System and programs bound by the Linkage Editor.

A cross-reference listing can be obtained upon request. This is an edited listing of ENTRY and EXTRN points with their program-relative addresses and method of satisfaction.

Equipment Configuration

Required

◆ Processor (65K)

Console typewriter.

Magnetic tape devices (three required).

Card reader or Videoscan document reader with card read feature.

Printer.

Optional

◆ Magnetic tape devices containing OMF's and/or OML's may be used as additional input. Magnetic tape devices may also be substituted for the card reader and the printer. Seven-level magnetic tape devices, with the pack/ unpack feature, may be substituted for all files except SYSRES.

Routine Parameters - General

◆ There are twelve control statements used with the Linkage Editor, each of which is optional. If none are used, a single object module program with no overlays is produced. For other results, the desired functions can be selected by supplying appropriate control statements.

With the exception of the ACTION statement, all control statements are effective only for the program in which they are found. That is, to obtain a cross-reference listing for all programs in a multibind run, each program must contain an XREF statement.

Following is a functional description of each parameter:

PROG Parameter

This parameter names the program to be bound and signals the start of the primary input (control statements and/or object modules) for the program which it names. If omitted, the name of the first explicitly-bound object module is used as the program name.

This parameter also permits definition of memory size and the version number of the bound program.

ENTRY Parameter

This parameter indicates the end of the primary input and specifies the execution entry point for the bound program. If omitted, or if the entry point operand is not present in this statement, the program's starting address will be the address specified in the END card of the first explicitly-bound module in the root load.

OVERLAY Parameter

This parameter provides the symbolic name of the origin of an overlay load, which is known as the node point. It also can be used to define multiple region overlays.

INCLUDE Parameter

This parameter is used to specify the device which contains object modules to be bound into a program currently being constructed.

Routine Parameters - General *(Cont'd)*

LIBRARY Parameter

This parameter specifies the Object Module Library to be searched for the satisfaction of external references which are unresolved at the time of the search. It can also be used to specify external references which, if unresolved, are not to cause a search of SYSLIB.

ACTION Parameter

This parameter specifies the format of the output tape. It can also provide data for the output tape header label. If this parameter is omitted, a Program Load Library Tape (PLLT) is produced.

NOMAP Parameter

This parameter specifies that a program map is not desired. If omitted, a program map is produced which lists the program load names, module names and length, and relative addresses.

XREF Parameter

This parameter specifies that a cross-reference listing is to be produced. The cross-reference listing describes the modules of a bound program. It shows their entry points with program-relative addresses, and their external references with program-relative addresses and method of satisfaction.

NOCTL Parameter

This parameter specifies that an overlay control module is not to be produced for the bound program. (If this parameter is used for a program that contains overlay segments, the loading of these overlay segments at object time must be controlled by LPOV macros.)

NCAL Parameter

This parameter specifies that the automatic library call mechanism is to be inhibited. In this case, the Linkage Editor will not attempt to resolve unsatisfied external references.

ERREXIT Parameter

This parameter specifies a value to be substituted for any unsatisfied external reference. If this parameter is not supplied, unsatisfied EXTRNS are assigned a value of zero by the Linkage Editor.

LET Parameter

This parameter specifies that a program is to be bound even though errors have been detected that may prevent its execution. It allows unsatisfied EXTRNS to be bound. Ⓒ

Placement of Parameters

◆ Example of Sequence

ACTION

PROG

NOMAP

XREF

NOCTL

NCAL

LET

$[\text{ Object Module}] \begin{cases} \text{INCLUDE xxxxxx} \\ \text{INCLUDE yyyyyy} \end{cases}$

OVERLAY

[Object Module]

OVERLAY

INCLUDE

LIBRARY

ERREXIT

ENTRY

If used, the ACTION statement must be the first parameter.

A PROG statement must precede the control statement or object module of the program that it names. It cannot appear in secondary input.

NOMAP, XREF, NOCTL, NCAL, and LET statements, if used, must appear in primary input. Any PROG statement must precede these statements, and any ENTRY statement must follow them.

An OVERLAY statement must precede each overlay module appearing in the primary data set, and must also be used before each INCLUDE statement that calls for an overlay module.

LIBRARY, INCLUDE, and ERREXIT statements may appear at any point within primary or secondary input. They may be placed within object modules and control statements, or interspersed within object modules. If an INCLUDE statement appears within an object module, it is processed at the completion of the module that contains it.

An ENTRY statement should follow the last input item (object module or control statement) for each program to be bound.

Routine Parameters - Detailed

ACTION Parameter

◆ *Format:*

$$\Delta\text{ACTION}\Delta \begin{Bmatrix}\text{PRGLD}\\ \text{SYSLD}\end{Bmatrix} \text{,nnnnnnnn,vvv,} \begin{Bmatrix}\text{mm/dd/yy}\\ \text{yyddd}\end{Bmatrix} , \begin{Bmatrix}\text{yyddd}\\ \text{ssssss}\end{Bmatrix}$$

Entry	Meaning
ΔACTIONΔ	Parameter identifier.
PRGLD or SYSLD	Specifies format of the output tape: PRGLD = Program Load Library (PLLT) is to be produced. SYSLD = System Load Library (SLLT) is to be produced.
,nnnnnnnn	Name of library tape to be produced (one to eight characters). If no name is desired, this operand may be omitted.
,vvv	Version number of the library tape to be produced. If this entry is omitted, zeros are used for the version number.
,mm/dd/yy or ,yyddd	mm/dd/yy = creation date of library if SLLT is being produced. yyddd = creation date of library if PLLT is being produced. If omitted, the current date is used as the creation date.
,yyddd or ,ssssss	yyddd = purge date if PLLT is being produced. ssssss = Symbolic name of device containing Executive system if SLLT is being produced. If this operand is omitted for a PLLT, the current date is used as the purge date. If it is omitted for an SLLT, the Executive system is assumed to be on SYSLIB.

Notes:

1. If this statement is not present, the Linkage Editor will produce a PLLT.
2. If any operand is omitted, an intervening comma must be supplied.

Examples:

ΔACTIONΔPRGLD

ΔACTIONΔPRGLD,UTILITY,014,66135,66288

ΔACTIONΔSYSLD,,015,,05/15/66

ΔACTIONΔSYSLD,UTILITY,,05/15/66,SYS014

ΔACTIONΔSYSLD,,,,SYS014

PROG Parameter

◆ *Format:*

$$\Delta\text{PROG}\Delta\text{nnnnnnnn,}\left\{\begin{matrix}\text{MAX * nnnnnn}\\ \text{MAX + nnnnnn}\end{matrix}\right\}\text{,VERvvv}$$

Entry	Meaning
ΔPROGΔ	Parameter identifier.
nnnnnnnn	Name of program to be bound (one to eight characters). Note: Although provision is made for eight characters, only the first six are used.
,MAX * nnnnnn or ,MAX + nnnnnn	*nnnnnn = maximum amount of memory area available for the program to be bound. + nnnnnn = amount of memory to be added to the program length computed by the Linkage Editor. Notes: nnnnnn is a six-digit decimal number, zero-filled. If this operand is omitted, the maximum length of the program is that of its longest path.
,VERvvv	vvv = version number of the program being bound. If omitted, 000 is used as the version number.

Notes:

1. The PROG statement may only appear within primary input.
2. When more than one program is to be bound, the programmer should ensure that his programs appear in ascending sequence by program name and that each program contains a PROG statement.

Examples:

ΔPROGΔTEST,MAX + 000100,VER001

ΔPROGΔTEST,MAX * 001000

ΔPROGΔTEST,VER001

ENTRY Parameter

◆ *Format:*

ΛENTRYΛnnnnnnnn

Entry	**Meaning**
ΛENTRYΛ	Parameter identifier.
nnnnnnnn	Symbolic name of the starting execution address of the program being bound. If this operand is omitted, the starting address in the END card of the first explicitly-bound module is used. If no address appears in this END card, the starting address will be the first byte of the first module.

Notes:

1. If this parameter is omitted, the end of the primary input for the program is determined by the logical end of file on the primary input, or by the presence of the PROG statement for the next program.
2. The starting address specified by nnnnnnnn must correspond to a symbol defined by an assembly ENTRY, CSECT, or START statement in one of the object modules bound into the root load.
3. If a starting address is not specified, comments may not be used.

Examples:

ΛENTRY

ΛENTRYΛMODULE1

OVERLAY Parameter

◆ *Format:*

∧OVERLAY∧nnnnnnnn,REGION,llllllll

Entry	**Meaning**
∧OVERLAY∧	Parameter identifier.
nnnnnnnn	Symbolic origin, or node point, of the load being bound (one to eight characters). The first time a node name is used, it defines a node point whose location is equal to the address of the last byte of the preceding load plus one (double-word boundary). Subsequent use of the same node name refers to this address.
,REGION	Use of this operand causes the node point of the load to be set to the end of the longest preceding path plus one, adjusted to a double-word boundary. This operand can be used when two or more exclusive loads are required in memory at one time. This entry may be omitted if not required.
,llllllll	Symbolic name of the load. Although provision is made for an eight-character name, only the first six characters are used. Each load for a program must have a unique load name. If a load name is not specified, the name of the load will be the same as the first module in the load. If this name is all spaces, a name of AAAnnn is generated for the load. For each succeeding load, nnn is incremented by 1.

Note:

An OVERLAY parameter may not precede the first explicitly-bound object module, which is the root load. Also, it cannot appear within the bounds of an object module.

Examples:

∧OVERLAY∧ALPHA

∧OVERLAY∧ALPHA,JOBA

∧OVERLAY∧BETA,REGION,JOBX

OVERLAY Parameter (Cont'd)

Following is an example of a program that uses two regions of memory. The first region consists of the root load and four overlays. The second region consists of a special purpose load that is required to be in memory at the same time as the other overlay loads.

Description of Input Data

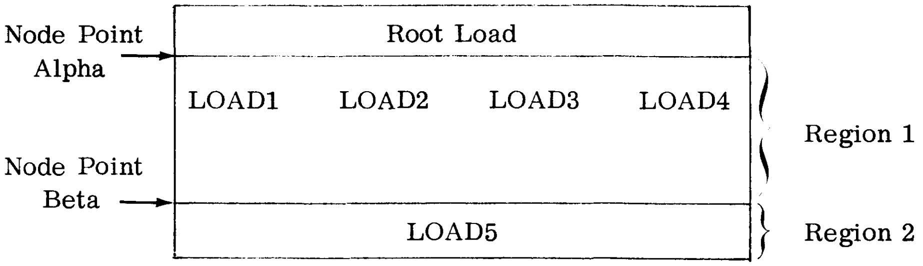
Object Module	Node Point Name	Load Name
Root load		
First overlay	ALPHA	LOAD1
Second overlay	ALPHA	LOAD2
Third overlay	ALPHA	LOAD3
Fourth overlay	ALPHA	LOAD4
Special Purpose Load	BETA	LOAD5

Arrangement of Input

```
ΔPROGΔSAMPLE
   [ Root load object module ]
ΔOVERLAYΔALPHA,LOAD1
   [ Overlay module ]
ΔOVERLAYΔALPHA,LOAD2
   [ Overlay module ]
ΔOVERLAYΔALPHA,LOAD3
   [ Overlay module ]
ΔOVERLAYΔALPHA,LOAD4
   [ Overlay module ]
ΔOVERLAYΔBETA,REGION,LOAD5
   [ Overlay module for region 2 ]
ΔENTRY
```

Output

Memory Allocation



INCLUDE Parameter

◆ *Format:*

ΔINCLUDEΔdddddd(nnnnnnnn,...)

Entry	Meaning
ΔINCLUDEΔ	Parameter identifier.
dddddd	Symbolic name of device on which there is an OMF or OML containing module(s) to be bound. If omitted, SYSLIB is assumed to contain the object module(s).
(nnnnnnnn,...)	Symbolic name of object module(s) to be bound. If this operand is not present, the entire OMF on the device specified will be bound. Commas must be used to separate all module names within parentheses.

Notes:

1. This parameter may only be used to extract modules from a secondary data set. (See Input Section for description of primary and secondary data sets.)
2. INCLUDE statements may be nested within object modules by placing an INCLUDE statement between the ESD card and the END card of the object module. (See example following notes.)
3. When INCLUDE statements are nested, they are satisfied in the following order:
    a. INCLUDE's found in an Object Module Library are bound first.
    b. INCLUDE's found in an Object Module File are bound next.
    c. After the above have been satisfied, INCLUDE's in the primary data set are processed.

    If an INCLUDE statement is not nested within the module, the module is bound into the program at that point.
4. An INCLUDE statement appearing in the secondary input may not reference an OMF, but it may reference an OML.

Examples:

ΔINCLUDEΔSYSUT1(MOD1)

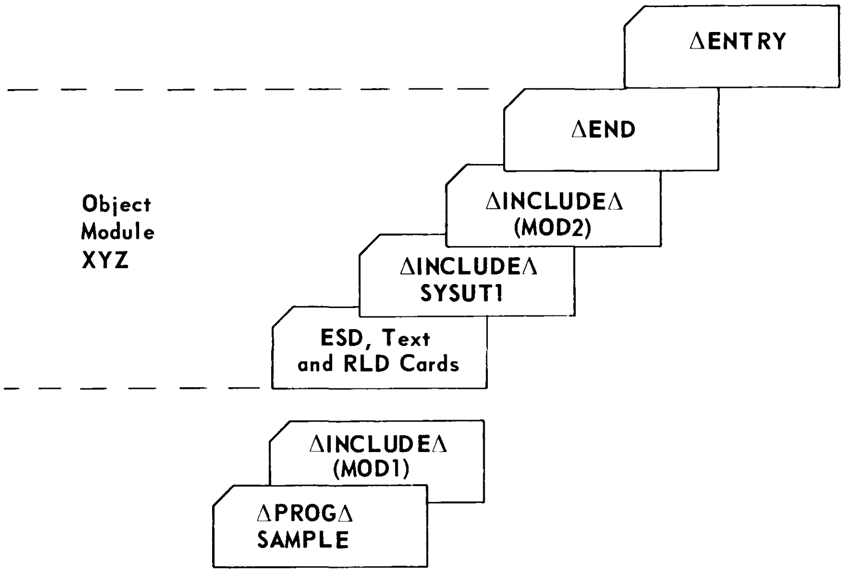
ΔINCLUDEΔ(MOD1,MOD5)

ΔINCLUDEΔSYSUT1

INCLUDE Parameter (Cont'd)

Following is an example of the use of multiple OVERLAY and INCLUDE statements in both primary and secondary input, and the resultant load structure.

Primary Input (Cards)



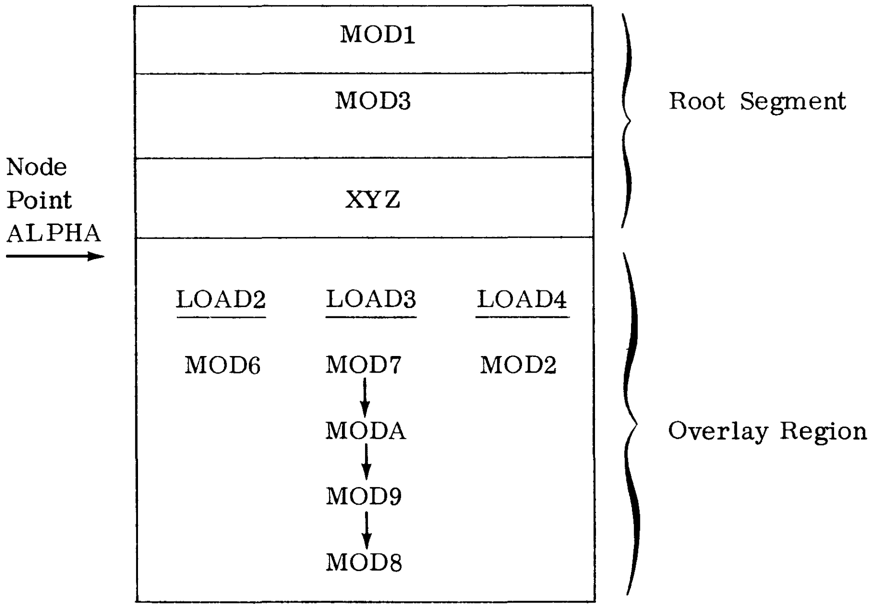
Secondary Inputs

OML (SYSLIB)
MOD1 (INCLUDE MOD3)
MOD2
MOD3
MOD4
MOD5 (INCLUDE MOD4)
MOD6
MOD7 (INCLUDE MODA)
MOD8
MOD9
MODA (INCLUDE MOD9)
MODB

OMF (SYSUT1)
LABEL
TM
OVERLAY ALPHA,LOAD2
INCLUDE (MOD6)
OVERLAY ALPHA,LOAD3
INCLUDE (MOD7,MOD8)
OVERLAY ALPHA,LOAD4
TM
TM

INCLUDE Parameter (Cont'd)

Resultant Load Structure



Explanation:

The first INCLUDE card in the primary input will read in MOD1. It will search for this on SYSLIB as no symbolic device name was supplied. MOD1 contains a nested INCLUDE MOD3 statement, so MOD3 is bound next. After this, control returns to the primary input and module XYZ is bound. This module contains a nested INCLUDE statement that instructs the Linkage Editor to refer to SYSUT1 and bind the complete OMF.

SYSUT1 contains an object module file which has no distinct modules of its own, but refers to several that are contained on SYSLIB. SYSUT1 also contains three OVERLAY statements. An overlay named LOAD2 is bound containing MOD6 from SYSLIB. Another overlay named LOAD3 is bound containing MOD7 and MOD8 from SYSLIB and their nested INCLUDES. When reading the third overlay statement for LOAD4, the Linkage Editor finds no further input on SYSUT1, so it returns to the primary input where it finds an INCLUDE statement for MOD2. This module is then bound into LOAD4.

The ENTRY statement is read next, ending the Linkage Editor processing for the program named "Sample".

Other programs may follow and will be bound on the same output PLLT.

LIBRARY Parameter

◆ *Format:*

ΔLIBRARYΔ dddddd(nnnnnnnn, . . .)

Entry	**Meaning**
ΔLIBRARYΔ	Parameter identifier.
dddddd	Symbolic name of device containing an OML which is to be searched for specified external references if they are unresolved by other input. This entry may be omitted. (See next entry for results.)
(nnnnnnnn, . . .)	Symbolic name of an external reference(s) which, if unresolved, should be searched for on the device specified above. If the previous entry (dddddd) is not given, then this operand specifies an external reference which, if unresolved, is not to cause a search of SYSLIB.

Notes:

1. All modules bound because of unresolved external references are bound into the root load.
2. Libraries specified in the LIBRARY statement are searched in the order specified, only once, in a forward direction.
3. SYSLIB is the last library searched.
4. Entry points contained within an object module bound because of a LIBRARY statement, or bound from SYSLIB, are used to satisfy any other unresolved external references.

Examples:

Directing Implicit Calling:

```
ΔLIBRARYΔSYSUT1(EXTERN1)
ΔLIBRARYΔSYSUT2(EXTERN5,EXTERN42)
```

Suppressing Implicit Calling:

```
ΔLIBRARYΔ(EXTERN1)
```

ERREXIT Parameter

◆ *Format:*

ΔERREXITΔ $\left\{\begin{array}{l}A = nnnnnn\\E = eeeeeeee\end{array}\right\}$

Entry	Meaning
ΔERREXITΔ	Parameter identifier.
A = nnnnnn or E = eeeeeeee	A = nnnnnn a six-digit decimal value to be substituted for all unsatisifed Externs. E = eeeeeeee a one- to eight-character symbolic address to be substituted for all unsatisfied Externs.

Notes:

1. If this parameter is not used, unsatisifed EXTRN's receive a zero address as their satisfying value.
2. If multiple ERREXIT statements are encountered, only the last one is effective.
3. When an entry point is specified by E =, it must correspond to a symbol defined by an Assembler ENTRY, CSECT, or START statement in an object module bound into the root load. Programs that violate this rule will not be bound.
4. The ERREXIT parameter affects all unsatisfied external references whether or not they are suppressed.
5. The LET parameter must also be used when the ERREXIT parameter is provided.
6. Absolute values are inserted in the object code in their hexadecimal representation.

Examples:

ΔERREXITΔA = 000001

ΔERREXITΔE = ADDRERR

ΔERREXITΔE = A

7. SYSTEM MAINTENANCE - RANDOM ACCESS

RANDOM ACCESS STORAGE ALLOCATOR (RAALLR)

General Description

◆ The Random Access Storage Allocator (RAALLR) reserves storage for a file on a random access volume by entering the name and limits of the file in the Volume Table of Contents (VTOC). This routine must be run before any file data can be loaded. Storage may be allocated for any file that is accessed by the serial, direct, or indexed-sequential method.

The information regarding the files to be allocated is supplied by parameter cards. Several files may be allocated in the same run.

Preset Functions

None.

Optional Functions

Two optional functions are provided:

1. Allocation - The programmer can assign a storage area for a file on a random access volume. This area can be allocated on a primary basis (the initial entry of a file), or by allocating additional areas (extents) for an existing file.

   The user can acquire space from the allocator by three methods. The first is by specifying the overall size of the area to be allocated and allowing the extent locations and sizes to be determined by the allocator. The second method is for the user to explicitly define each extent size and extent type (data, index, label, or overflow), leaving the allocator the task of determining if these extents are available and creating the necessary labels and extents lists. The third is an automatic allocation for indexed-sequential which uses *DTFIS parameters.

2. Deallocation & Purging - The programmer can deallocate a storage area for a specified file or a specified extent. Cards within an extent can be deallocated on mass storage except in indexed-sequential files.

   When a file or extent is deallocated, it also can be purged; that is, each track is erased and reestablished as an initialized track.

---

*For explanation of DTFIS, refer to Section 3, "Indexed Sequential FCP", of TOS/TDOS FCP and Executive Communications Macros Manual, (70-00-608).

Input

◆ Input to this routine consists of one or more random access volumes and one set of parameter cards for each file to be acted on by the allocator.

Output

◆ Output is composed of one or more random access volumes with storage allocated to one or more files. The routine also provides a listing of parameters and errors encountered.

Equipment Configuration

Required

◆ Processor (65K)

Console typerwiter

Card reader

Random access device(s) (70/564 Disc Storage Unit, 70/565 Drum Memory Unit, 70/567* Drum Memory Unit, or 70/568 Magazine)

Optional

◆ Printer

A paper tape reader or a magnetic tape device may be substituted for the card reader.

Routine Parameters - General

◆ There are seven parameters associated with this routine.

DLAB Parameter

Ⓒ This parameter specifies the file name, the file serial number, and the number of consecutive days from allocation date.

UNITS Parameter

This parameter defines, by serial number, the volumes of which the file is comprised.

ALLOC Parameter

This parameter specifies whether the operation will be a first-time allocation or an addition to an existing file; the type of file; the amount of storage needed for data, indexes, and overflow; and the bandwidth for a mass storage, indexed-sequential file.

XTENT Parameter

This parameter can be used to define extent boundaries by specific cylinders to be used for file allocation or deallocation. Any number of XTENT cards are permitted.

---

*70/567 is supported only as a 70/565

Routine Parameters - General *(Cont'd)*

DEPUR Parameter

This parameter can be used to indicate that an area previously allocated to a file is to be released for use. It also is used to specify whether or not the area is to be purged.

DTFIS Parameter

The DTFIS card provides a simplified way of declaring storage requirements for an indexed-sequential file by taking the DTFIS statement from the program that will load or update the file and inserting it intact into the allocation set following the DLAB and UNITS cards. DTFIS declarations peculiar to ISFCP are recognized by the Allocator, but ignored, and vice versa. The declarations unique to allocation are (1) the number of records in the file (FILSIZE); (2) the percentage of the space required for the file that is to be additionally allocated as a general overflow pool (OFOPCT); and (3) for mass storage files, the width of the band (BNDSIZE).

END Parameter

This parameter indicates the end of parameter input.

Routine Parameters - Detailed

DLAB Parameter

◆ This parameter contains the file name, the number of days the file will be active, and the file's serial number.

Format:

ΔΔDLABΔ'filename',ssssss,days

Entry	Meaning
ΔΔDLABΔ	Parameter identifier.
'	Apostrophe (8-5 punch).
filename	Actual file name (1-44 bytes) in the Format 1 file label.
'	Apostrophe (8-5 punch).
,ssssss	Six-character file serial number. Must be identical to the serial number of the first or only volume in the file.
,days	Number (1 to 8 digits) of consecutive days from date of allocation. (See note 1.)

Ⓒ

DLAB Parameter (Cont'd)

Notes:

1. If days entry is omitted, file will be allocated with today's date as expiration date.
2. The DLAB parameter must be the first parameter in a set for each file.

Examples:

```
ΔΔDLABΔ'FILEΔONE',VOL001,365
ΔΔDLABΔ'WORKΔFILE',000036
```

UNITS Parameter

◆ This parameter defines the volumes of which the file is comprised. It is a required parameter and always follows the DLAB.

Format:

```
ΔΔUNITSΔssssss,nnnn,.........ssssss,nnnn
```

Entry	**Meaning**
ΔΔUNITSΔ	Parameter identifier.
ssssss	Volume serial number as found in the DLAB card. If this is a multivolume allocation, succeeding serial numbers must be in the order the volumes are to be sequenced.
,nnnn	Volume sequence number for this volume of the file. If omitted for a multivolume allocation, volumes will be sequenced in the order in which they are listed in this parameter. (See example.)

Notes:

1. When extending a file, only the volumes which the allocator will process need to be put in the parameter. The sequence numbers for these volumes must also appear.
2. If relative track addressing is to be used to process the file, the first volume sequence number must be 0000.
3. When processing a mass storage indexed-sequential file, the disc or drum that contains the indexes must always be specified in the UNITS parameter as the first volume.

Examples:

```
ΔΔUNITSΔVOL001,,VOL002,,VOL003
ΔΔUNITSΔ000001
```

ALLOC Parameter

◆ This parameter specifies type of allocation, type of file, size of file, and space required for data, indexes, and overflow areas.

Format:

ΔΔALLOCΔT=n,A=n,D=c,I=c,O=c,B=bb

Entry	**Meaning**
ΔΔALLOCΔ	Parameter identifier.
T-n,	Type of file to be allocated: n=0 nonindexed-sequential. =1 indexed-sequential (disc/drum). =2 indexed-sequential (mass storage).
A=n,	Type of allocation to be performed: n=0 primary allocation. =1 addition to an existing file.
D=c,	Total cylinders or cards required for file data: c = number cylinders for disc/drum or nonindexed - sequential mass storage files. = number cards for mass storage (indexed sequential files only). See Note 5.
I=c,	c = number of cylinders to be allocated for indexed-sequential indexes (disc/drum only).
O=c,	Amount of space to be allocated for indexed-sequential overflow: c = number of cylinders for disc/drum. = number of cards for mass storage.
B=bb	Band size to be allocated per card for indexed-sequential mass storage data and overflow extents: bb = number (1-16) of cylinders per band. See Note 6.

Notes:

1. Omitted entries do <u>not</u> require positional commas.

2. When allocation is performed, all files whose expiration dates have passed are not automatically deallocated and their space are not made available for allocation. Ⓒ

ALLOC Parameter (Cont'd)

This applies to all types of files except mass storage indexed-sequential. A message will be printed out (3124) indicating that such a file has expired.

3. When an entry is omitted, the zero (0) value is assumed. For example, Type of file is assumed to be nonindexed-sequential if the T=entry is omitted.
4. Before allocating indexed-sequential files, the user should be thoroughly familiar with Section 3, Indexed Sequential FCP, of TOS/TDOS FCP and Executive Communication Macros Manual, (70-00-608).
5. If XTENT cards are to be used, the D=entry must not be supplied.
6. When extending a file, the B=entry must be given.

Examples:

To allocate a nonindexed-sequential file:

```
ΔΔALLOCΔT=0,A=0,D=125
```

To allocate an indexed-sequential file on mass storage:

```
ΔΔALLOCΔT=2,A=0,D=120,I=1,O=12,B=07
```

XTENT Parameter

◆ This parameter can be used to define extent boundaries for allocation and for deallocation.

A new XTENT parameter must be present for each volume to be processed, but one XTENT parameter can contain any number of extents for one volume.

Format:

```
ΔΔXTENTΔssssss,t,ccchhccccchh,.....t,ccchhccccchh
```

Entry	Meaning
ΔΔXTENTΔ	Parameter identifier
ssssss,	ssssss = volume serial number as found in Format 1 label
t,	Extent type: t = D Data = O Overflow* = I Index* = M Other than complete cylinder boundaries.**
ccchhccccchh	Left- and right-hand end cylinder and track limits: cccc = cylinder number hh = track number (Must be zero filled)

*For indexed-sequential files only **Non-indexed-sequential only.

DTFIS Parameter

◆ The DTFIS card provides a simplified way of declaring storage requirements for an indexed-sequential file by taking the DTFIS statement from the program that will load or update the file and inserting it intact into the allocation set following the DLAB and UNITS cards. DTFIS declarations peculiar to ISFCP are recognized by the Allocator, but ignored, and vice versa. The declarations unique to allocation are (1) the number of records in the file (FILSIZE); (2) the percentage of the space required for the file that is to be additionally allocated as a general overflow pool (OFOPCT); and (3) for mass storage files, the width of the band (BNDSIZE).

XTENT Parameter (Cont'd)

Note:

Multiple extents that are contiguous and are in their physical order on the XTENT card(s) are considered as one extent by RAALLR.

Example:

ΔΔXTENTΔ004539,D,005000019909

DEPUR Parameter

◆ This parameter can be used to indicate that all space or particular extents allocated to a file are to be released and/or purged.

Format:

ΔΔDEPURΔD=n,P=n

Entry	Meaning
ΔΔDEPURΔ	Parameter identifier
D=n	Type of deallocation: n = F deallocate entire file = X deallocate by extent
,P=n	Type of purge: n = F purge entire file = X purge by extent

Note:

1. If X entries are specified in the parameter, an XTENT parameter must follow the DEPUR parameter.
2. When purge is specified, the allocator reinitializes the Track Descriptor records (R0) in the specified file or extent. It also erases data from the tracks.
3. The DEPUR should always precede the ALLOC parameter, when deallocation of the previous file is preferred. Ⓐ

Examples:

ΔΔDEPURΔD=F,P=F	
ΔΔDEPURΔP=X	(XTENT card must follow)
ΔΔDEPURΔD=X	(XTENTcard must follow)

DTFIS Parameter (Cont'd)

Format: (See note 2)

Entry	Meaning
DTFISΔ	Parameter identifier
BNDSIZE=cc,	cc = number of cylinders that make up the band.
BLKSIZE=n,	Specifies the number of bytes the user wants as a maximum for blocking his data in each primary data sentence. If his parameter is omitted, it is assumed that only one sentence per track is desired. In this case, the maximum for disc is 3597 minus KEYLEN, and the maximum for drum is 3021 minus KEYLEN. For mass storage, 2190 is always assumed. When the user supplies a BLKSIZE specification, he determines the size of the upper index level sentences. If omitted, 2190 is assumed.
CYLOF=n,	This parameter specifies the number of tracks in each primary data cylinder that are to be reserved for overflow records. Primary sentences will not be written on these tracks. The value of n cannot exceed cylinder (or band) size minus 2. If this parameter is omitted, 0 is assumed.
FILSIZE=n,	Number of data records in the file. This is a required entry.
ISDEN=n,	This parameter specifies the percentage of each primary block to be filled while loading the file. If not specified, 100 is assumed. The block is written when a blocked record causes the percentage to be met, or when the next record would cause it to be exceeded.
KEYLEN=n,	This parameter specifies the number of bytes in the record key of each record in the file.
OFLOPCT=n,	Percentage of data storage to be additionally allocated as an overflow pool.
PRIDEVT=n	This optional parameter describes the device type. n = DISK - Device is 70/564 Disc = DRUM - Device is 70/565 Drum = MASS - Device is 70/568 Mass Storage Unit If omitted, DISK is assumed.

ⓒ

ⓒ

DTFIS Parameter (Cont'd)

Entry	Meaning
PRINDEX = $\left\{\begin{matrix}\text{MASS}\\ \text{n}\end{matrix}\right\}$	This parameter specifies the device type on which the primary track indexes are to be kept. If PRINDEX = MASS, each mass storage index sentence is written on the first good track in the band to which it refers. If PRINDEX = n, the primary track indexes are written on the same device as the upper index (UPINDEX) levels, and n + 12 is the size of the data field in the mass storage index sentence. Regardless of device type, only one mass storage index sentence is written per track, and the form of the sentence does not vary except in length. If this parameter is omitted, PRINDEX = MASS is assumed (70/568 only).
RECSIZE=n,	n = Average number of bytes per record. Record size always must be stated. n cannot be greater than n in the BLKSIZE parameter. If RECSIZE is given for variable records, the size of overflow sentences containing a single record is based on the record size field in the record or the RECSIZE specification, whichever is greater. (This allows the user a hedge against expanding records being relocated each time they are updated, with a corresponding loss of the original record space.) Ⓒ
UPINDEX=n	This parameter declares the device type on which the levels of indexing higher than the primary track index are kept. n = DISK - device is 70/564 disc = DRUM - device is 70/565 drum (If omitted, DISK is assumed)

Notes:

1. It is expected that the user will have generated his file's DTFIS before allocation so that he can use the same DTFIS parameters for allocation.

   With the exception of FILSIZE, BNDSIZE, and OFLOPCT, all entries are identical to those in the DTFIS macro.

DTFIS Parameter (Cont'd)

2. The format of the DTFIS parameter is different from the format of the other allocator parameters. It follows the macro instruction format as follows:

Card Column	1-9	10-15	16 71	72	73-80
	ISFILE	DTFISΔ	OPERAND1,OPERAND2, OPERANDn	C	

   Many of the DTFIS operands are optional, but it is not necessary for a comma to appear for every possible entry. Commas should appear only for those operands that are used.

   When the entries in the operand field require more than one card, a "C" must be punched in column 72 for all cards of the set except the last one.

3. DTFIS operands other than those explained in the preceding table are recognized by the allocator, but ignored.

4. Once an Indexed-Sequential file has been OPENed, file specification such as block size, and record size must not be changed in the user's program. If such changes are made, the file must be deallocated and then reallocated before processing.

END Parameter

◆ This parameter is required and signifies the end of parameter input.

Format:

ΔΔENDΔ

Considerations for use

◆ The following points should be considered in using the allocator:

1. The On-Line Catalog console routine must be run prior to running the allocator.

2. Parameter cards can contain comments as long as there are at least two spaces between the last entry in the card and the beginning of the comment.

3. All volumes to be allocated must have been initialized by the Random Access Volume Initializer (RAINIT).

4. Unless a file structure requires that specific absolute assignments be made on random access devices, the methods of allocation by total number of cylinders and (for indexed-sequential files only) by file definition (DTFIS) are simpler to use.

5. When an allocation is performed, the routine first checks every file label in the VTOC for an expiration date equal or previous to today's date. If any are found, the format 1 label is removed

Considerations for use *(Cont'd)*

from the VTOC, thus making this area available for allocation. This area is not purged, however, and any data previously written remains on the disc.

If the programmer plans to transcribe to the disc and use standard programming systems, he must make sure that the Track Descriptor records are reset to indicate that the tracks can be written to. This is done by following the first allocation for every file with a DEPUR parameter, in the format ΔΔDEPURΔP-F. See Examples.

If extents are added to a file after data has been transcribed, a run must be made using XTENT cards for the new extents only. If a ΔΔDEPURΔP=F is used following an extension, any data on previously allocated extents will be destroyed.

6. Allocation for a mass storage indexed-sequential file requires that at least the first sequential volume be either a disc or a drum in which index extents are to be allocated and mass storage extents are to be recorded.

7. Alternate track assignments are made in a mass storage flaw pool by RAINIT at initialization time. These assignments are reset to unassigned alternate tracks if the magazine is being allocated for an indexed-sequential file. In this case, the flaw pool is extended, if necessary, in both directions until card boundaries are reached.

8. Note carefully the assumptions that are made by the allocator upon encountering various errors in the parameters. In many cases, the invalid parameter is not rejected, and allocation will be performed based on assumptions made by the routine.

Examples

◆ 1. This example illustrates the allocation of a new nonindexed-sequential disc file to cylinders 9 through 200:

```
ΔΔDLABΔ'DISCΔFILE',DISC03,90
ΔΔUNITSΔDISC03
ΔΔALLOCΔT=0,A=0
ΔΔXTENTΔDISC03,D,000900020009
ΔΔEND
```

2. This example illustrates allocation for a new indexed-sequential mass storage file:

```
ΔΔDLABΔ'MASSΔSTORAGEΔFILE',DISC04,366
ΔΔUNITSΔDISC04,,MASS13
ΔΔALLOCΔA=0,T=2,B=07,D=100,I=1,O=8
ΔΔENDΔ
```

Examples *(Cont'd)*

3. This example illustrates the extension of data storage for the file in example 2 by adding 24 cards:

```
ΔΔ DLABΔ'MASSΔSTORAGEΔFILE', DISC04
ΔΔ UNITSΔDISC04,,MASS13
ΔΔ ALLOCΔA=1,T=2,D=24,B=04
ΔΔEND
```

4. This example illustrates allocation of a disc indexed-sequential file of 200,000 records with an overflow pool allowance of 20 percent:

©

```
ΔISFILE..ΔDTFISΔ PRIDEVT=DISK,Δ......C
                 RECSIZE=250,        C
                 BLKSIZE=1500,       C
                 KEYLEN=20,          C
                 FILESIZE=200000,    C
                 CYLOF=1,            C
                 ISDEN=90,           C
                 OFLOPCT=20
```

Device Assignments

◆ Under Monitor Control:

SDN	Device Type	Remarks
SYSIPT	Card Reader.	Parameter input.
SYSLST	Printer or magnetic tape.	Output listing.

Under Executive Control:

SDN	Device Type	Remarks
SYSRDR	Card Reader	Parameter input.
SYSPRT	Printer	*Output listing.

*If no logging of parameters or error messages is desired, the operator can respond with NO instead of a device assignment.

RANDOM ACCESS INDEX EDIT (RAINDX)

General Description

◆ The Random Access Index Edit (RAINDX) provides an index of the files and labels on a random access volume. This edit includes:

1. The serial number and owner identification code of the standard volume label for the volume.
2. Address limits of the VTOC.
3. Unused area remaining in the VTOC for additional file labels and extent entries.
4. Unused areas (extents) available on the volume.
5. Listing of the names of all files stored on the volume and the areas assigned to these files.

Preset Functions

The preset functions of this routine include steps 1-4 described above.

Optional Functions

A listing of the names and areas assigned to any or all files on the volume may be obtained.

More than one volume may be edited in a single run.

Input

◆ Input to this routine consists of the random access volume (or volumes) to be edited plus one or more parameter cards for each volume to be processed.

Output

◆ Output consists of an edited listing of the label area of the volume. This listing can be directed to the printer, the typewriter, or written onto magnetic tape.

Equipment Configuration

Required

◆ Processor (65K)

Console typewriter

Card reader or Videoscan document reader with card read feature

Printer

Random access device

Optional

◆ A magnetic tape device may be substituted for the printer as the output device.

The console typewriter may be used as input and/or output device.

Routine Parameters - General

◆ There are three parameters associated with this routine:

VOLM Parameter

This parameter is used to designate which volume is to be edited. It may also designate the typewriter as the output device.

FILE Parameter

This parameter may be used to designate which file labels are to be edited and listed.

END Parameter

This parameter signifies the end of parameter input.

Routine Parameters - Detailed

VOLM Parameter

◆ *Format:*

$$\Delta\text{VOLM}\Delta\left\{\begin{matrix}\text{ssssss}\\ \text{ASSIGN}\end{matrix}\right\}\Delta\text{T}$$

Entry	Meaning
ΔVOLMΔ	Parameter identifier.
ssssssΔ or ASSIGNΔ	ssssss = serial number of volume to be edited. (Serial number must be included for mass storage device). ASSIGN = edit volume on device assigned to RDMDVC.
T or blank	T = display output to console typewriter. blank = display output to SYSLST if running under Monitor control; to OUTDVC if running under Executive Control.

Note:

The output device specified in the first VOLM parameter will be the output device for all subsequent VOLM parameters.

Examples:

ΔVOLMΔ000777ΔT

ΔVOLMΔASSIGN

Considerations For Use

◆ 1. If magnetic tape is the output device, the preedit option of the Tape Edit routine must be used to print the output tape.

2. The volume serial number specified in the VOLM parameter <u>must</u> appear in the On-Line Catalog.

Examples

◆ 1. Running under Executive Control:

Edit all files on random access volume number 000777 with output directed to the typewriter.

(Device assignments will be requested by the Executive on the typewriter.)

```
ΔVOLMΔ000777ΔT
ΔFILEΔALLΔFILES
ΔEND
```

2. Running under Monitor Control:

Edit file ACCOUNTS PAYABLE of the volume assigned to RDMDVC with output directed to the printer.

```
//ΔSTARTM
//ΔASSGNΔSYSLST,L1
//ΔJOB
//ΔASSGNΔRDMDVC,A1
//ΔEXECΔRAINDX
ΔVOLMΔASSIGN
ΔFILEΔACCOUNTSΔPAYABLE
ΔEND
//ΔENDMON
```

```
                        RANDOM ACCESS INDEX EDIT          DATE: 021167          T02

STANDARD VOLUME LABEL                                                   ①  ②  ③④⑤

    SERIAL NUMBER: 000777    OWNER ID CODE: 0000000000    VTOC ADDRESS: 000:000/0:0/004

VTOC DESCRIPTION
                ⑥                                               ⑦
    VTOC EXTENT: 000:000/0:0 TO 000:000/0:9    RECORDS AVAILABLE: 0155    LAST FORMAT 1: 000:000/0:0 /013

FORMAT 5 LABEL(S)

    EXTENTS AVAILABLE IN VOLUME:  (RELATIVE TRACK/WHOLE CYLINDERS AVAILABLE)
     ⑧   ⑨
    0010/0010  0580/0142  2010/0002  0000/0000  0000/0000  0000/0000  0000/0000  0000/0000
      ⑭
FORMAT B LABEL(S)

    EXTENTS AVAILABLE IN VOLUME:  (LHE CARD-CYL/RHE CARD-CYL)

    0000/0000  0000/0000  0000/0000  0000/0000  0000/0000  0000/0000  0000/0000  0000/0000

DATA FILE DESCRIPTION
              ⑩
    FILE: KAY OCONNORS FILE ONE                 CREATION DATE: 66 360           EXPIRATION DATE: 68 365
                ⑪                  ⑫
Ⓒ   EXTENTS: D(001) 000:011/0:0 TO 000:021/0:9

    FILE: KAY OCONNORS FILE TWO                 CREATION DATE: 66 360
               ⑬
Ⓒ   EXTENTS: D(001) 000:022/0:0 TO 000:031/0:9

    FILE: KO FILE THREE                         CREATION DATE: 67 024

Ⓒ   EXTENTS: D(001) 000:032/0:0 TO 000:041/0:9

    FILE: KO FILE FOUR                          CREATION DATE: 67 038

Ⓒ   EXTENTS: D(001) 000:042/0:0 TO 000:043/0:9

    FILE: KO FILE FIVE
          ⑭
    EXTENTS: I(000)000:000/0:0 TO 000:000/0:0     SER #:000000
```

①	Card number if mass storage
②	Cylinder number
③	Zero
④	Track number
⑤	Record number
⑥	Start & End addresses of VTOC (see 1-5 above).
⑦	No. of file label records available.
⑧	First relative track no. available (#10)
⑨	Number of cylinders available (10).
⑩	File ID
⑪	Extent sequence number
⑫	Start & End addresses of Extent
⑬	Extent Identifier D = Data Extent I = Index Extent O = Overflow Extent L = Label Extent
⑭	Mass Storage Index Sequential only.

Output Examples

FILE Parameter

◆ *Format:*

$\Delta\text{FILE}\Delta\left\{\begin{array}{l}\text{filename}\\ \text{ALL}\Delta\text{FILES}\end{array}\right\}$

Entry	Meaning
ΔFILEΔ	Parameter identifier.
filename or ALLΔFILES	filename = name of file to be edited, 1-44 characters. (Must be identical to the filename field in the Format 1 file label that is to be edited.) ALLΔFILES = list all files stored in the VTOC.

Examples:

ΔFILEΔACCTSΔPAYABLE

ΔFILEΔALLΔFILES

END Parameter

◆ *Format:*

ΔEND

Device Assignments

◆ Under Executive Control:

SDN	Device	Remarks
PRMDVC	Card reader	Parameter input.
OUTDVC	Printer or magnetic tape	Output listing.
RDMDVC	Random access volume	Necessary when ASSIGN specified in VOLM parameter.

Under Monitor Control:

SDN	Device	Remarks
SYSIPT	Card reader	Parameter input.
SYSLST	Printer or magnetic tape	Output listing.
RDMDVC	Random access volume	Necessary when ASSIGN specified in VOLM parameter.

70/568 SERVICE PROGRAM (RAMSUP)

General Description

◆ The 70/568 Service Program (RAMSUP) provides a number of service functions designed for Spectra 70 systems that utilize the 70/568 Mass Storage Device. This Program may be employed to obtain magnetic tape backup for data stored on mass storage devices, to reconstruct data areas of a mass storage device, to duplicate magazines or portions thereof, to initialize or replace individual cards, and to obtain a magazine usage analysis report.

Any or all of the functions mentioned above can be selected by the use of routine parameters supplied at run time.

Preset Functions

None.

Optional Functions

The following optional functions may be elected:

1. Copying all or selected portions of a magazine to another magazine.
2. Copying all or selected portions of a magazine to magnetic tape.
3. Copying all or selected portions of a magazine to another magazine and also to magnetic tape.
4. Reloading a magazine with data stored on magnetic tape.
5. Initializing individual cards.
6. Analyzing selected card extract counters and indicating the number of defective tracks, if any, on all cards in the magazine.
7. To replace individual cards.

Input

◆ Input to this routine consists of a 70/568 magazine (or magnetic tape) and routine parameters that specify the service functions desired.

Output

◆ Output may be directed to a 70/568 magazine, to magnetic tape, and to the printer, depending on the functions requested by the input parameters.

Equipment Configuration

Required

◆ Processor (65K)

Console typewriter

Card reader or Videoscan document reader with card read feature

Mass Storage Unit (70/568)

Magnetic tape device

Printer

Equipment Configuration

Optional

◆ Magnetic tape may be substituted for the reader when operating under Monitor Control.

Additional magnetic tape devices may be assigned as alternate input and output devices.

Routine Parameters-General

Processing functions provided by RAMSUP are selected by the following routine parameters. Parameters may be used in any combination subject only to the equipment complement.

CPYM — Copy Magazine to Magazine

This parameter provides the ability to copy all or specified portions of one magazine to another magazine.

CPYT — Copy Magazine to Tape

This parameter provides the ability to copy all or specified portions of a magazine to tape.

CYMT — Copy Magazine to Magazine and Tape

This parameter provides the ability to copy all or specified portions of a magazine to another magazine and to tape.

RELD — Reload

This parameter provides the ability to reload all or specified portions of a magazine from a tape created by a function of this routine. It may be used in three ways: to reload specific cards, to reload an entire magazine or a number of consecutive cylinders, or to reload cylinder 0 of card 0 (VTOC) only.

CINT — Card Initialize

This parameter provides the ability to initialize cards that have been inserted into a magazine to replace damaged or worn cards.

ANAL — Magazine Usage Analysis

This parameter provides a check of the card extract counter of specified cards, and a listing of the number of defective tracks on each card.

RPLC — Card Replacement

This parameter provides the ability to replace a readable worn card by copying the data on the card to tape, allowing the operator to substitute a new card, and restoring the data to the new card from tape.

Routine Parameters-General *(Cont'd)*

TANK

This parameter can be used to inform the routine to read all parameters into memory at one time and to deallocate the reader for use by another routine.

END

This parameter denotes the end of parameter input.

Routine Parameters-Detailed

Copy Magazine to Magazine

◆ This parameter provides the ability to copy all or specified portions of one magazine to another magazine.

Format:

ΔCPYMΔM=(ssssss),O=(ssssss),S=(a,b),E=(c,d),B=(e,f)

Entry	Meaning
ΔCPYMΔ	Parameter identifier.
M=(ssssss)	ssssss = volume serial number (one to six characters) of the input magazine.
,O=(ssssss)	ssssss = volume serial number (one to six characters) of the output magazine.
,S=(a,b)	Optional. Starting cylinder of area to be copied: a = card number (0-255). b = cylinder number (0-15). See note 5.
,E=(c,d)	Optional. Ending cylinder of area to be copied: c = card number (0-255). d = cylinder number (0-15). See note 5.
,B=(e,f)	Optional. First cylinder of the output area. (Only used if the starting point of the output area is not the same as the input area.) e = card number (0-255). f = cylinder number (0-15). See note 5.

Notes:

1. The output magazine must be initialized prior to the copy.

2. Alternate tracks on input cards are not copied as alternate tracks on the output magazine. Alternate output tracks are only assigned if defective tracks are encountered when writing to the output magazine.

Copy Magazine to Magazine (Cont'd)

3. This function requires a work tape. The input cards are first transcribed to the work tape and then read reverse during which the data is copied to the output magazine.
4. When only input and output magazine serial numbers are supplied, an entire magazine is copied.
5. If the S = entry is used, the E = entry also must be used. If the B = entry is used, then S = and E = entries must be present

Examples:

1. To copy one magazine to another:

   ΔCPYMΔM=(2),O=(000307)

2. To copy three complete cards of one magazine to the same three cards on the output magazine:

   ΔCPYMΔM=(999999),O=(321), S=(201,0), E=(203,15)

3. To copy three complete cards of one magazine to another area on the output magazine:

   ΔCPYMΔM=(6382),O=(000521),S=(201,0),E=(203,15),B=(101,0)

Copy Magazine to Tape

◆ This parameter provides the ability to copy all or specified portions of a magazine to magnetic tape.

Format:

ΔCPYTΔM=(ssssss),S=(a,b),E=(c,d),MFV,NOALT

Entry	Meaning
ΔCPYTΔ	Parameter identifier.
M=(ssssss)	ssssss = volume serial number (one to six characters) of the input magazine.
,S=(a,b)	Optional. Starting cylinder of area to be copied: a = card number (0-255). b = cylinder number (0-15). See note 2.
,E=(c,d)	Optional. Ending cylinder of area to be copied: c = card number (0-255). d = cylinder number (0-15). See note 2.
,MFV	MFV = this file is an addition to the output tape which already contains one or more files. If not used, this file is to be the first (or only) file on the output tape.
,NOALT	NOALT = only one tape drive is available for output. If not used, an alternate tape drive is available for output tape swapping.

Copy Magazine to Tape (Cont'd).

Notes:

1. See Considerations for Use, page 7-29, for a description of how flaw tracks and alternate track areas are processed.
2. If the S = entry is used, the E = entry also must be used.

Examples:

1. To copy an entire magazine to tape:

```
ΔCPYTΔM=(78)
```

2. To copy three cards to an output tape that has existing files:

```
ΔCPYTΔM=(630),S=(133,0),E=(135,15),MFV
```

3. To copy 100 cards as the first file on an output tape with only one tape drive available for the output:

```
ΔCPYTΔM=(000019),S=(2,0),E=(101,15),NOALT
```

4. To copy partial cards to tape:

```
ΔCPYTΔM=(76),S=(120,7),E=(123,12)
```

Copy Magazine to Magazine and Tape

◆ This parameter provides the ability to copy all or specified portions of one magazine to another magazine and also to magnetic tape.

Format:

```
ΔCYMTΔM=(ssssss),O=(ssssss),S=(a,b),E=(c,d),B=(e,f),NOALT
```

Entry	Meaning
ΔCYMTΔ	Parameter identifier.
M=(ssssss)	ssssss = volume serial number (one to six characters) of the <u>input</u> magazine.
,O=(ssssss)	ssssss = volume serial number (one to six characters) of the <u>output</u> magazine.
,S=(a,b)	Optional. Starting cylinder of area to be copied: a = card number (0-255). b = cylinder number (0-15). See note 4.
,E=(c,d)	Optional. Ending cylinder of area to be copied: a = card number (0-255). d = cylinder number (0-15). See note 4.

Copy Magazine to Magazine and Tape (Cont'd)

Entry	Meaning
,B=(e,f)	Optional. First card of the output area. (Only used if the starting point of the output area is not the same as the input area.) e = card number (0-255). f = cylinder number (0-15). See note 4.
,NOALT	NOALT = only one tape drive is available for output. If not used, an alternate tape drive is available for output tape swapping.

Notes:

1. The output magazine must be initialized prior to the copy.
2. Alternate tracks on input cards are not copied as alternate tracks on the output magazine. Alternate output tracks are only assigned if defective tracks are encountered when writing to the output magazine.
3. When only input and output magazine serial numbers are supplied, an entire magazine is copied.
4. If the S = entry is used, the E = entry also must be used. If the B = entry is used, then S = and E = entries must be present.

Example:

To copy an entire magazine to another magazine and also to magnetic tape.

ΔCYMTΔM=(424272),O=(663392)

Reload

◆ This parameter provides the ability to reload all or specified portions of a magazine from a tape created by a function of this routine. There are three formats:

Format 1 — Reload Specific Cards:

ΔRELDΔM=(ssssss),C=(ccc,.....,ccc),O=(ssssss),MFV,NOALT Ⓐ

Entry	Meaning
ΔRELDΔ	Parameter identifier.
M=(ssssss)	ssssss = volume serial number (one to six characters) of the magazine image on the input tape.
,C=(ccc,..,ccc)	ccc = card number (0-255) to be loaded. A maximum of 15 cards may be specified.

Reload (Cont'd)

Entry	Meaning
,O=(ssssss)	ssssss = volume serial number (one to six characters) of the output magazine. (If omitted, the output volume serial number is assumed to be the same as the magazine serial number on the tape.)
,MFV	MFV = input tape contains one or more files. If not used, the input tape contains one file.
,NOALT	NOALT = only one tape drive is available for input. If not used, an alternate tape drive is available for input tape swapping.

Ⓐ

Note:

Any cards recorded as complete card images on tape may be reloaded by this format. Partially recorded cards are not valid input to this function.

Format 2 — Reload Complete Magazine or group of Consecutive Cylinders

Ⓒ

ΔRELDΔ M=(ssssss),S=(a,b),E=(c,d),O=(ssssss),B=(e,f),MFV,NOALT

Entry	Meaning
ΔRELDΔ	Parameter identifier.
M=(ssssss)	ssssss = volume serial number (one to six characters) of the magazine to be loaded from the input tape.
,S=(a,b)	Optional. Starting cylinder of input magazine to be loaded: a = card number (0-255). b = cylinder number (0-15). See note.
,E=(c,d)	Optional. Ending cylinder of input magazine to be loaded. a = card number (0-255). b = cylinder number (0-15). See note.
,O=(ssssss)	ssssss = volume serial number (one to six characters) of output magazine. (If omitted, the output volume serial number is assumed to be the same as the magazine serial number on the input tape.)

Reload (Cont'd)

Entry	Meaning
,B=(e,f)	Optional. First cylinder of the output area. (Only used if the starting point of the output area is not the same as the cylinder designated in the S=entry.) e = card number (0-255). f = cylinder number (0-15).
,MFV	MFV = input tape contains one or more files. If not used, the input tape contains one file.
,NOALT	NOALT = only one tape drive is available for input. If not used, an alternate tape drive is available for input tape swapping.

Ⓐ

Note:

If the S=entry is used, the E=entry also must be used. If the B= entry is used, then S= and E= entries must be present and the O=entry is required.

Format 3 - Reload Cylinder 0, Card 0 (VTOC) Only:

ΔRELDΔM=(ssssss),D=(dd,m),MFV,NOALT

Ⓒ

Entry	Meaning
ΔRELDΔ	Parameter identifier.
M=(ssssss)	ssssss = volume serial number (one to six characters) of the magazine recorded on tape.
,D=(dd,m)	Device to be reloaded: dd = symbolic physical device name as it appears in the device assignment list. m = magazine number (0-07).
,MFV	,MFV = input tape contains one or more files. If not used, the input tape contains one file.
,NOALT	NOALT = only one tape drive is available for input. If not used, an alternate tape drive is available for input tape swapping.

Ⓐ

Examples:

1. To reload three cards from tape to the original magazine:

   ΔRELDΔM=(666666),C=(100,101,102)

2. To reload 10 cards to a magazine other than the original magazine and on different cards from which the data was originally recorded:

   ΔRELDΔM=(666666),S=(5,0),E=(14,15),O=(888888),B=(25,0)

3. To reload the VTOC of magazine 682350 on device E0 from a tape with multiple files:

   ΔRELDΔM=(682350),D=(E0,06),MFV

Card Initialize (Cont'd)

Entry	Meaning
, D=(dd, m)	Optional. Used only to initialize card 000. dd = symbolic device name of magazine as it appears in the device assignment list. m = magazine number (0-07).

Notes:

1. The card initialize function does not flag defective tracks or assign alternate tracks; these are assigned only during card loading.

Reload (Cont'd)

Notes:

1. Input to the RELD function must have been created by a function of this routine.

2. See note 3, Considerations for Use section, for details regarding the loading of primary data track areas and alternate track cards.

3. The alternate track cards may not be reloaded with a parameter that is being used to reload data cards.

Card Initialize

◆ This parameter provides the ability to initialize individual cards that have been inserted into a magazine to replace worn or damaged cards. It also provides a console listing of any defective tracks encountered, purges any alternate tracks referenced by the old card, and initializes the card extract counter for the new card.

This function does <u>not</u> write a volume label or a VTOC, when initializing a card zero.

Format:

$$\Delta\text{CINT}\Delta\text{M=(ssssss)},\left\{\begin{array}{l}\text{S=a, E=b}\\ \quad\text{or}\\ \text{C=(ccc,. . ,ccc)}\\ \quad\text{or}\\ \text{D=(dd,m)}\end{array}\right\}$$

Entry	Meaning
ΔCINTΔ	Parameter Identifier.
M=(ssssss)	ssssss = volume serial number (one to six characters) of the magazine to be initialized.
,S=a	Optional. a = card number (0-255) of first card in a consecutive series to be initialized.
,E=b	Optional. b = card number (0-255) of last card in a consecutive series to be initialized.
,C=(ccc,. . . . . ,ccc)	Optional. Specific cards, up to a maximum of 15, to be initialized. This entry is not used to initialize consecutive cards; the S and E entries are used for this purpose. ccc = card number (0-255).

Card Initialize (Cont'd)

2. A complete magazine cannot be initialized by merely specifying the magazine serial number. One of the options must be specified.

3. When initializing a card zero, this function does not write a volume label or a VTOC. If a card zero is to be initialized, it must be copied to tape first, initialized, and then reloaded.

Examples:

1. To initialize cards 5 through 10:

   ΔCINTΔM=(601),S=5,E=10

2. To initialize four nonconsecutive cards:

   ΔCINTΔM=(000601),C=(3,21,201,233)

3. To initialize card 000 of magazine serial number 000601:

   ΔCINTΔM=(601), D=(E1,06)

Magazine Usage Analysis

◆ This analysis consists of 1) a comparison of the card extract counter in each card specified to determine if the maximum number of card extracts has been reached for the card, and 2) an examination of the alternate track area with a listing of the number of defective tracks on each card in the magazine.

The number of flaws encountered per card and the card number of any card whose extract counter has reached the maximum count are provided on the printer. Also printed is the number of unused tracks in the alternate track area.

Format:

ΔANALΔM=(ssssss),S=s, E=e

Entry	Meaning
ΔANALΔ	Parameter identifier.
M=(ssssss)	ssssss = volume serial number (one to six characters) of magazine.
S=s	s = first card (0-255) of the area to be analyzed.
E=e	e = last card (0-255) of the area to be analyzed.

Note:

Extract counters for card 000 and the alternate track cards are always analyzed.

Examples:

1. To analyze cards 1 through 10 of magazine 242604:

   ΔANALΔM=(242604), S=1, E=10

2. To analyze an entire magazine:

   ΔANALΔM=(682390)

Card Replacement

◆ This parameter provides the ability to replace a readable worn card by copying the data on the card to tape, allowing the operator to substitute a new card, and restoring the data to the new card from tape. All cards being replaced, although worn, must be readable.

Up to 15 cards can be replaced at one time.

Format:

ΔRPLCΔM=(ssssss),C=(ccc,.....,ccc)

Entry	Meaning
ΔRPLCΔ	Parameter identifier.
M=(ssssss)	ssssss = volume serial number (one to six characters) of magazine.
,C=(ccc,...,ccc)	ccc = card number (0-255) to be replaced. Up to 15 cards may be specified.

Notes:

1. Cards being used as replacement cards need not be initialized; this is accomplished automatically by this routine.
2. As the information on the worn card is written to tape, any data stored in an alternate track area for that card is also written to tape. When this data is reloaded to the replacement card, it is transcribed to a primary track. References to the worn card are deleted from the alternate track area.

Example:

To replace cards 1, 7, and 10 of magazine 187285:

ΔRPLCΔM=(187285), C=(1,7,10)

TANK

◆ This parameter can be used at run time to specify that all parameters which follow are to be read and stored in memory, after which the reader is to be deallocated. The RAMSUP routine will thus process parameters from this memory storage area rather than by issuing a card read instruction. This parameter is not applicable when running under Monitor.

Format:

ΔTANK

Note:

When this function is used, additional memory must be allocated for the parameter storage area. Allow 80 x X locations, where X is the number of parameters to be stored.

END

◆ This parameter denotes the end of parameter input and must be the last parameter supplied.

Format:

ΔEND

Considerations For Use

◆ Before using the RAMSUP routine, the following points should be reviewed for their possible effect on the type of processing desired.

1. The On-Line Catalog console routine must be run before this program is executed except when cylinder 0, card 0 is being initialized or reloaded.

2. Magnetic Tape Formats

   Tapes created by this routine are in standard Spectra 70 format and contain standard Volume and Header labels. In addition, the first and last data record of each file will contain information about the file and the magazine from which it was created. These records are identified as VSN records.

   a. Header labels

      If no label parameters (VOL and TPLAB) are supplied, RAMSUP generates a standard file Header label with the file identifier: RAMSΔTAPEΔFILEΔΔΔ . When run-time parameters are provided, the VOL parameter must be written as follows:

      //ΔVOLΔSYS001, ITUTAPE

      It is recommended that the user-supplied TPLAB cards should contain the Volume serial number of the magazine(s) being dumped.

   b. VSN Records

      These records identify the contents of the file. They are recorded at both the beginning and end of each file on the tape to facilitate the read reverse function of this routine.

      The format for the VSN record is as follows:

Bytes	Contents	Meaning
1-3	VSN	Record identifier.
4		Blank.
5-10	serial number	Volume serial number of magazine from which the data was copied.
11		Blank.

Considerations for Use *(Cont'd)*

Bytes	Contents	Meaning
12-26	see notes.	
27	F3 or F1 (hexadecimal)	Indicator that specifies the contents of the preceding field. F3 = field contains starting and ending card numbers. F1 = field contains a list of individual card numbers.
29-32	cchh (hexadecimal)	Left-hand end of alternate track area.
33-36	cchh (hexadecimal)	Right-hand end of alternate track area.

Note 1: If the S=(a,b) and E=(c,d) entries appear in the routine parameter card, the format of bytes 12-26 is as follows:

Bytes	Contents	Meaning
12	a	Starting card number.
13	b	Starting cylinder number.
14	c	Ending card number.
15	d	Ending cylinder number.
16-19	abcd	Same information as contained in bytes 12-15.
20-26	Hexadecimal 0's	Not used.

If the parameter also contains the B=(e,f) entry, bytes 16-17 above contain the output starting card and output starting cylinder numbers. Bytes 18-19 contain the ending card and cylinder numbers which are calculated by the program.

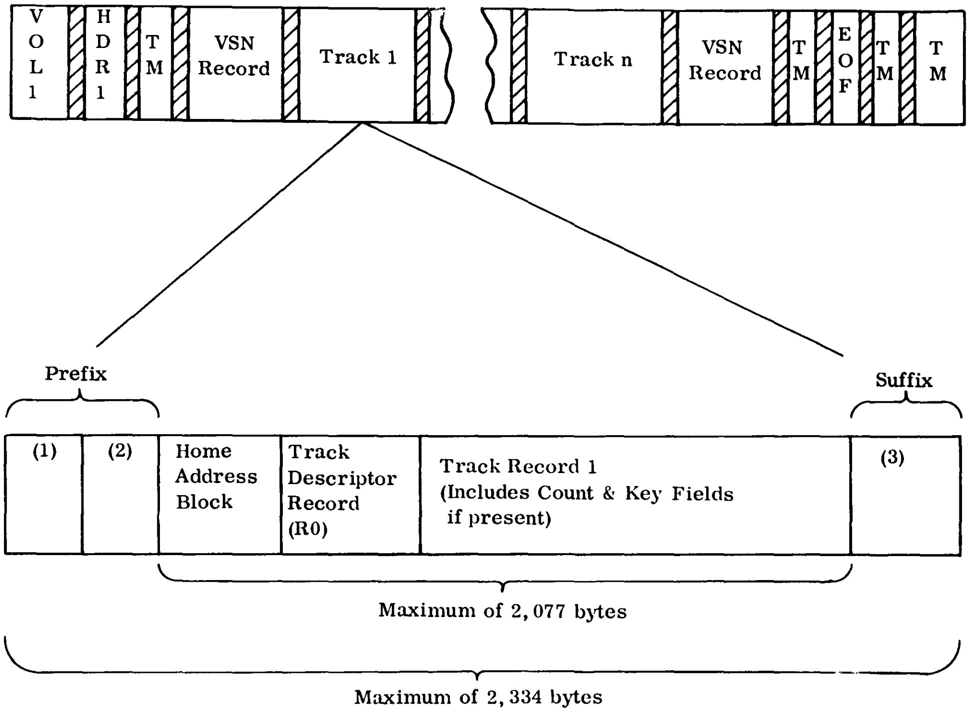
Note 2: If the parameter specifies individual cards (C=(ccc,ccc,ccc,)), bytes 12-26 contain a sorted list of the cards. The first byte contains the card number of the lowest-numbered card; succeeding bytes contain card numbers in the ascending sequence.

A maximum of 15 card numbers can be stored in this area. If less are specified, card numbers are left-justified and the remaining bytes are filled with hexadecimal zeros.

c. Data Records

The following illustrations depict the format of the output tape created by the RAMSUP routine.

Considerations For Use *(Cont'd)*



1) One-byte binary counter that specifies the length of the prefix fields. Preset by this routine to $01_{(16)}$.

2) Optional user field of up to 254 bytes. The RAMSUP routine does not generate this field when unloading 70/568 tracks to tape. The routine will, however, allow for this field when loading from tape to 70/568 cards.

3) If required, this routine generates a one- or two-byte appendage to guarantee that the track information on tape is an even multiple of three bytes (for reverse-reading purposes).

3. Processing of Defective and Alternate Tracks

a. Writing to Magnetic Tape:

If a defective track is encountered on an input card, an entry is made for that track in a table area maintained by the RAMSUP routine. The track is then skipped, and the routine continues to copy the remaining good tracks on the card to the output tape. After all readable tracks have been transcribed, the table area is accessed to locate the alternate track associated with each defective track. The data from the alternate track area is then transcribed to the output tape. The next card is processed.

Consideration for Use *(Cont'd)*

When a card containing an alternate track area is encountered, the tracks on the card are transcribed to tape exactly as they appear.

b. Writing to a 70/568 Magazine:

If a card being reloaded had originally contained defective tracks when it was unloaded to tape, the alternate track records are written to the output card as primary track records and not as alternate track records.

If a defective track is encountered on the output card, the Flag byte of the track is set to indicate that it is defective. The data is then written to the alternate track area and linkage made to the defective track.

c. When loading a card that contains the alternate track area, transcription is made on a track-for-track basis. However, if a defective track is sensed in the alternate track area on the output card, the routine will request assignment of tape SYS003. The alternate that could not be recorded is written to SYS003 and the routine then processes the next track. After the routine reloads all alternate tracks that can be recorded, any tracks written to SYS003 are reloaded to new locations in the alternate track area. A pointer is then placed in the original track to indicate where the data has been relocated.

d. Because of the manner in which defective tracks and alternate track cards are processed, this routine may not be used to load cards with defective tracks and alternate track cards with the same parameter.

Device Assignments

- Under Monitor or Executive:

SDN	Device Type	Remarks
SYSIPT	Card reader. (May be magnetic tape under Monitor only).	Parameter input.
SYSLST	Printer.	Output listings.
SYS001	Magnetic tape.	Primary input or output tape. Also required as a work tape for CPYM function.
SYS002	Magnetic tape.	Alternate input or output tape.
SYS003	Magnetic tape.	Input or output device used by RELD function. Assignment of this device is requested only if it is needed.

```
                                                  08/10/67

  USAGE ANALYSIS FOR MAGAZINE 000888 ①

  PART I: EXTRACT COUNTER EQUAL OR EXCEEDS LIMIT FOR CARDS-

②001 002 004 005 014 015 016 018


  PART II: CARDS CONTAINING FLAW TRACKS AND NUMBER OF FLAWS

  CRD  #F    CRD  #F    CRD  #F    CRD  #F    CRD  #F    CRD  #F    CRD  #F    CRD  #F    CRD  #F    CRD  #F    CRD  #F    CRD  #F    CRD  #F
③016④018    019 001    024 016    040 013    052 001    056 021    057 001    064 022    065 001    072 001    104 014    105 001    128 010
  175 001    187 003    203 006    208 013    248 007    249 002


⑤UNUSED TRACKS IN FLAW POOL:      148


  END OF REPORT
```

① Volume serial number of magazine being analyzed

② Card numbers of cards requested with exceeded extract counters

③ Card number

④ Number of defective tracks in this card

⑤ Number of available alternate tracks

DISC/DRUM DUMP AND RELOAD (DDRL)

General Description

Ⓐ ◆ The Disc/Drum Dump and Reload routine provides data backup of a disc or drum at the device or volume level by dumping the contents of either a disc or drum to magnetic tape and reloading from this tape. Unless otherwise specified, the entire device is dumped to magnetic tape. This routine also supports a testing environment by permitting the contents of a DDRL produced dump tape to be reloaded to a work device and to be tested immediately.

Preset functions.

None.

Optional functions.

Input or output devices may be specified as disc or drum.

Any number of cylinders may be dumped to tape or reloaded from tape. When reloading a disc or drum, any alternate track linkage that had been dumped to tape may be written back with the same linkage, or the routine can try to write to prime tracks only, supplying alternate track linkage where needed.

Ⓐ One Random Access Device may be copied directly to another Random Access Device.

Input

◆ Input to this routine consists of a disc or drum to be dumped to tape, or a magnetic tape to be reloaded to disc or drum. All tapes will have standard labels, either as generated by the routine or as indicated by VOL and TPLAB parameter cards. The record format is variable-length, unblocked.

Output

◆ Output of this routine may be a magnetic tape, or a disc or drum, depending on the function utilized.

Magnetic tape will be in the same format as described in input.

Equipment Configuration

Required

◆ Processor (65K).

Console typewriter.

Disc or drum.

Magnetic tape device.

Optional

◆ Parameters may be entered through the card reader.

Routine Parameters - General

◆ There are five parameters used with this routine:

Disc/Drum Dump to Tape Parameter

This parameter specifies the device to be dumped to tape, and the cylinders to be dumped from that device.

Disc/Drum Reload from Tape Parameter

This parameter specifies the device to be reloaded and the alternate track option to be followed. It also may specify that no read-after-write verification is to be performed.

Disc/Drum Copy to Disc/Drum Parameter

This parameter specifies the input and output devices to be copied, and the track and cylinder options to be followed. It also may specify that no read-after-write verification is to be performed.

No-Alternate Parameter

This parameter specifies that no alternate magnetic tape is available to the routine.

END Parameter

This parameter signifies the end of parameter input.

Routine Parameters - Detailed

Dump Parameter

◆ This parameter specifies the device (disc or drum) to be dumped to tape, and the cylinders to be dumped from that device.

Format:

$$\Delta\left\{\begin{matrix}\text{DSKD}\\ \text{DRMD}\end{matrix}\right\}\Delta\text{D}=\text{nnnnnn},\text{S}=\left\{\begin{matrix}\text{a}\\ \text{a/t}\end{matrix}\right\},\text{E}=\left\{\begin{matrix}\text{b}\\ \text{b/t}\end{matrix}\right\}$$

Entry	Meaning
$\Delta\left\{\begin{matrix}\text{DSKD}\\ \text{DRMD}\end{matrix}\right\}\Delta$	DSKD = disc dump to tape function. DRMD = drum dump to tape function.
D=nnnnnn	Symbolic device name of device being dumped. nnnnnn = alphanumeric field of one to six characters.
,S=a or ,S=a/t	Optional. Starting cylinder and track of area to be dumped. a = cylinder number (0-255). t = track number (0-19). See Note 1. Ⓒ
,E=b or ,E=b/t	Optional. Ending cylinder and track of area to be dumped. b = cylinder number (0-255). t = track number (0-19). See Note 1. Ⓒ

Dump Parameter (Cont'd)

Notes:

1. If the entire disc or drum is to be dumped, these entries should be Ⓐ omitted. If the S=entry is used, then the E=entry must also be used. If the 't' value is omitted, full cylinders are dumped.
2. The Volume Standard Label and VTOC are written to tape whether or not they are included in the parameter extents.

   All defective or flaw tracks are located and written in prime track sequence to tape, with the flaw track retaining the prime track home address.

   The alternate track area or any track flagged as an alternate will not be copied to tape if it is included in the parameter extents.

   Overflow records are placed on tape as separate records with an indicator in the program-generated flag byte which precedes each record. (See Considerations for Use, note 2, for detailed record format.)

Examples:

1. To dump an entire disc to tape:

   ΔDSKDΔD = SYSDSC
2. To dump ten cylinders of a disc to tape:

   ΔDSKDΔD = SYSRES, S = 1, E = 10
3. Ⓐ To dump cylinder 1, track 0 to cylinder 10, track 3 of a disc to tape:

   ΔDSKDΔD=SYSRES,S=1/0,E=10/3

Reload Parameters

◆ This parameter specifies the device to be reloaded and the alternate track option to be followed. It may also specify that no read-after-write verification is to be performed.

Format:

$\Delta\left\{\begin{matrix}\text{DSKR}\\ \text{DRMR}\end{matrix}\right\}\Delta\text{D = nnnnnn, SAVDEF, NRAW}$

Entry	Meaning
$\Delta\left\{\begin{matrix}\text{DSKR}\\ \text{DRMR}\end{matrix}\right\}\Delta$	DSKR = reload disc from tape function. DRMR = reload drum from tape function.
D=nnnnnn	Symbolic device name of device being reloaded.
Ⓒ ,SAVDEF	Optional. Specifies that all tracks read from tape whose indicators show that they were dumped from alternate tracks should be dynamically placed in the indicated alternate track without attempting to write them to the prime data track. See Note 1.
,NRAW	Optional. Specifies that the routine will not perform read-after-write verification.

Reload Parameters (Cont'd)

Notes:

1. The routine normally provides dynamic alternate or flaw track assignment so that the tape may be reloaded to a disc or drum other than the one used to create the tape. This means that the routine will attempt to write each track to its prime track, and if unsuccessful, write it to an alternate track, establishing appropriate linkage.

   If the option, SAVDEF, is specified in the reload parameter, the routine will not perform dynamic alternate track assignment.

2. If no alternate track area exists, or the alternate track area is filled, a request for one additional extent is made through the console typewriter. If an additional extent is assigned, subsequent prime tracks encountered on tape which fall within this extent will not be reloaded to the device.

   This alternate track area is initialized by the routine (Home Address and Track Descriptor Records are written with the flag byte set to indicate an alternate track) prior to reloading from tape.

Examples:

1. To reload an entire disc, attempting to write all alternate tracks to their prime locations, without read-after-write verification:

   ΔDSKRΔD=SYSDSC,NRAW

2. To reload a disc, writing all alternate tracks to the alternate track area without attempting to write them to the prime track and with read-after-write verification:

   ΔDSKRΔD=DSCDEV,SAVDEF

Copy Parameter

◆ This parameter specifies the input and output devices to be copied, and the track and cylinder option to be followed. It also can specify that no read-after-write verification is to be performed. Ⓐ

Format:

$$\Delta \begin{Bmatrix} \text{DKDK} \\ \text{DMDM} \end{Bmatrix} \Delta \ \text{DI=nnnnnn, DO-nnnnnn, S=} \begin{Bmatrix} \text{a} \\ \text{a/t} \end{Bmatrix} \text{, E=} \begin{Bmatrix} \text{b} \\ \text{b/t} \end{Bmatrix} ,$$

SAVDEF,NRAW Ⓐ

Copy Parameter (Cont'd)

Entry	Meaning
$\Delta \left\{ \begin{array}{l} \text{DKDK} \\ \text{DMDM} \end{array} \right\} \Delta$	DKDK = copy disc to disc function. DMDM = copy drum to drum function.
DI=nnnnnn	Symbolic device name of input device being copied.
DO=nnnnnn	Symbolic device name of output device.
,S=a or ,S=a/t	Optional. Starting cylinder and track of area being copied. a = cylinder number (0-255). t = track number (0-19). See note 1.
,E=b or E=b/t	Optional. Ending cylinder and track of area being copied. b = cylinder number (0-255). t = track number (0-19). See note 1.
,SAVDEV	Optional. Specifies that all tracks read, whose indicators show that they were dumped from alternate tracks, should be dynamically placed in the indicated alternate track without attempting to write them to the prime data track.
,NRAW	Optional. Specifies that the routine will not perform read-after-write verification.

Notes:

1. If the entire disc or drum is to be copied, these entries should be omitted. If the S=entry is used, then the E=entry also must be used. If the 't' value is omitted, full cylinders are copied.

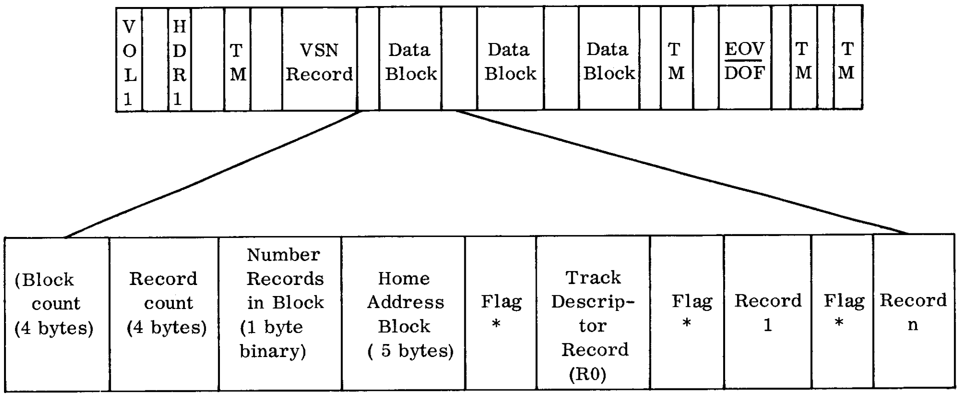
2. The Volume standard label and VTOC where they exist, are included in the copy, whether or not they are included in the parameter limits.

   All defective or flaw tracks are located and written in prime track sequence to random access with the flaw track retaining the prime track home address.

   The alternate track or any track that is flagged as an alternate will not be copied to random access if it is included in the parameter extents.

Considerations for Use *(Cont'd)*

2. The following illustration depicts the format of the output tape created by this routine:



*A one-byte flag placed in front of each record by the routine. If the record was read as an overflow record, the 2^0 bit is set in this byte. The remaining seven bits are unused.

3. When parameters are entered through the console typewriter, the leading space character is not used.

4. The routine supports the Model 70/564 Disc Storage Unit and the Model 70/567 Drum Memory. Ⓐ

Device Assignments

◆ Under Monitor or Executive:

SDN	Device Type	Remarks
SYSIPT	Card reader. May be magnetic tape under Monitor only.	Parameter Input.
SYSLST	Monitor device only.	For Monitor only.
SYS001	Magnetic tape.	Primary input or output.
SYS002	Magnetic tape.	Alternate input or output.
nnnnnn	Disc or drum. nnnnnn = name specified in parameter card.	Input or output.

Copy Parameter (Cont'd)

Examples:

1. To copy an entire disc to disc:

```
ΔDKDKΔDI=SYSRES,DO=SYSDSC
```

2. To copy 12 cylinders of a disc to a disc.

```
ΔDKDKΔDI=SYSRES,DO=SYSDSC,S=1,E=12
```

3. To copy cylinder 1, track 0 to cylinder 12, track 5 of a disc to a disc.

```
ΔDKDKΔDI=SYSRES,DO=SYSDSC,S=1/0,E=12/5
```

No-Alternate Parameter

◆ *Format:*

```
ΔNALT
```

Entry	Meaning
ΔNALT	Specifies that no alternate magnetic tape is available. See Note.

Note:

This optional parameter is processed the first time it appears. It sets SYS001 as the permanent symbolic tape assignment.

End Parameter

◆ This parameter is required and denotes the end of parameter input.

Format:

```
ΔEND
```

Considerations for Use

◆ 1. If no run time label parameters (VOL and TPLAB) are supplied, the routine generates a standard file header label with the file identifier: DDRLΔTAPEΔFILEΔΔ. The current date is used for both creation and expiration dates.

When run-time parameters are provided, the VOL parameter must be written as follows:

```
//ΔVOLΔSYS001,ITUTAPE
```

RANDOM ACCESS DUMP AND RELOAD (RADAR)

General Description

◆ The Random Access Dump and Reload Program routine (RADAR) provides dump and reload/reconstruct functions for disc, drum, and mass storage. These functions are performed on a file, extent, or volume basis. Volumes can contain indexed sequential or nonindexed sequential files.

The functions are selected by routine parameters supplied at run time.

Preset Functions

◆ None.

Optional Functions

◆ The following optional functions are provided:

Dump any number of extents from any number of random access volumes to one magnetic tape file.

Dump every extent allocated to a random access file to one magnetic tape file.

Reload dump or after-image logs from magnetic tape to a random access volume. A complete volume or portions of a volume may be reloaded.

Reload before-image records from magnetic tape to the appropriate random access volume(s), reloading the entire file or terminating at a specific checkpoint or at a specific point in time.

Input

◆ Input to this routine consists of a random access device or magnetic tape and routine parameters that specify the functions desired.

Output

◆ Output from this routine is to a random access device or magnetic tape depending upon the functions selected.

Equipment Configuration

Required

◆ Processor (65K)
Console typewriter
Card reader (not required if parameters are entered through the console typewriter).
Random access device
Magnetic tape device (nine-level or 7 -level with pack/unpack feature).

Optional

◆ Magnetic tape may be substituted for the card reader when operating under Monitor.

Additional magnetic tape devices may be assigned as alternate input or output devices or as a checkpoint device.

Routine Parameters
General

◆ Functions performed by RADAR are selected by the following routine parameters:

DUMPV - Dump Volume

The DUMPV parameter indicates that physical extents, without regard to file boundaries, are to be copied from a random access volume to a magnetic tape file. A single parameter can cause an entire random access volume or a specific range of cylinders to be copied to tape.

The DUMPV parameter for a 70/568 volume may specify dumping within a single magazine in either a card/cylinder or band/cylinder range.

DUMPF - Dump File

The DUMPF parameter indicates that a logical file is to be copied to magnetic tape. The file can consist of a single extent or of multiple extents and may be single or multivolume.

The only information required in the parameter is the volume serial number of the first or only volume of the file and the filename.

RELDA - Reload After Images

The RELDA parameter specifies that records are to be reloaded from tape to a random access volume. Input may consist of records produced by the DUMPV or DUMPF function, after-image logs produced by the LOGA function of DTFIS, and/or sorted after-image logs.

The records may be reloaded to the volume they originally came from or to a new volume. A complete volume or designated parts of a volume may be reloaded.

RELDB - Reload Before Images

The RELDB parameter indicates that before-image records up to a specified termination point, are to be reloaded from tape to one or more random access devices. The termination point specified in the parameter may be the end of the tape file, a designated checkpoint, or a certain before-image record.

Input may consist of before-image logs produced by the LOGB function of DTFIS or sorted before-image logs.

NOALT - No Alternate

The NOALT parameter specifies that no magnetic tape device is available as an alternate to the primary input/output device. If more than one volume is required during dump or reload the routine waits until the new volume is mounted on the primary device before continuing.

Routine Parameters General *(Cont'd)*

NDPSL - No Display

The NDPSL parameter specifies that parameters are not to be displayed on the console typewriter. It does not prevent parameters which are in error from being displayed.

This parameter may appear anywhere in the input stream and is effective only for subsequent parameters.

CKPT - Checkpoint

The CKPT parameter specifies that checkpoints are to be taken at each primary or alternate magnetic tape device change.

Checkpoints are initiated through the standard CKPT macro. The CKPT applies only to the function with which it is associated. If checkpoints are to be taken during a reload before-image function the CKPT parameter must precede the RELDB parameter.

SQEND - Sequence End

The SQEND parameter indicates that the routine is to execute the functions specified by the parameters already processed. After these functions have been completed, processing of parameters resumes.

The SQEND parameter may follow one or more DUMPV and/or DUMPF parameters, to permit more than one random access volume or file to be dumped to a single tape file. SQEND may follow one or more RELDA statements, permitting the reloading of a number of extents in a single operation and reducing tape passing time. RELDB parameters are executed when encountered in the input stream.

The SQEND statement does not effect NOALT and NDSPL options; these apply for the complete run. CKPT statement options are reset by a SQEND parameter.

END

The END parameter indicates that the routine is to execute the function specified in the preceding parameters and then terminate.

Routine Parameters Detailed

Dump Volume

◆ This parameter specifies that physical extents, without regard to file boundaries, are to be copied from a random access volume to a magnetic tape file.

Format:

```
ΔDUMPVΔ IPVSN=s,START=c/y,END=c/y,BAND=nn
```

Dump Volume (Cont'd)

Entry	Meaning
ΔDUMPVΔ	Parameter identifier.
IPVSN=s	s = serial number of the volume to be dumped (one to six characters).
,START=c/y	Optional. Start of a contiguous area to be dumped. c = card number (0-255; used for the 70/568 only. y = cylinder number: 0-202 for 70/564; 0-255 for 70/567; 0-15 for 70/568. **(See note 1.)**
,END=c/y	Optional. End of a contiguous area to be dumped. c = card number (0-255); used for the 70/568 only. y = cylinder number: 0-202 for 70/564; 0-255 for 70/567; 0-15 for 70/568. (See note 1.)
,BAND=nn	Optional. Number of cylinders in the upper band on the cards in the area to be dumped (1-15). This is required when dumping a 70/568 extent by band.

Note:

1. When dumping prime data tracks from any random access device, tracks are retrieved from the alternate track area and substituted for defective prime data tracks. The alternate track area and VTOC are not copied to tape.

Examples:

1. To dump the entire volume identified by volume serial number 000777 to tape:

ΔDUMPVΔIPVSN=000777

2. To dump cylinders 20 to 119 of volume ABC to tape:

ΔDUMPVΔIPVSN=ABC,START=20,END=119

Dump Volume (Cont'd)

3. To dump cylinders 10 to 15 of card 1; 0 to 15 of cards 2 to 5; and 0 to 5 of card 6 of the 70/568 volume T32.

```
ΔDUMPVΔIPVSN=T32,START=1/10,END=6/5
```

4. To dump the lower band of cards 10, 11, and 12 of 70/568 volume 555; the upper band consists of cylinders 0-3, the lower band consists of cylinders 4-15:

```
ΔDUMPVΔIPVSN=555,START=10/4,END=12/15,BAND=4
```

Dump File

◆ This parameter specifies that a logical file is to be copied to magnetic tape.

Format:

```
ΔDUMPFΔIPVSN=s,FNAME='fileid'
```

Entry	Meaning
ΔDUMPFΔ	Parameter identifier.
IPVSN=s	s = serial number of the file to be dumped (one to six characters).
,FNAME='fileid'	fileid = file identification as it appears in the VTOC format 1 record (1 to 44 characters).

Notes:

1. A VDC run-time parameter is required when dumping multivolume 70/564 or 70/567 files.
2. When dumping a multivolume 70/568 indexed sequential file the IPVSN entry must reflect the disc containing the file index and this volume must be on-line.
3. When dumping prime data tracks from any random access device, tracks are retrieved from the alternate track area and substituted for defective prime data tracks. The alternate track area and VTOC are not copied to tape.

Examples:

1. To dump the file UNSORTED TRANS contained on volume 000778 to tape:

```
ΔDUMPFΔIPVSN=000778,FNAME='UNSORTEDΔTRANS'
```

Reload After Images

◆ This parameter indicates that dump or after-image records are to be reloaded from tape to a random access volume.

Format:

ΔRELDAΔIPVSN=s,OPVSN=s $\left\{\begin{array}{l}\text{,CARDS=c/c/.....c}\\ \text{, START=c/y/t,END=c/y/t,BAND=nn}\end{array}\right\}$

,PDEV=mn/b,ALTND=c/y/t,ALTSZ=nnnn,SCALT= $\left\{\begin{array}{l}\text{YES}\\ \underline{\text{NO}}\end{array}\right\}$

,PURGE= $\left\{\begin{array}{l}\text{YES}\\ \underline{\text{NO}}\end{array}\right\}$

Entry	**Meaning**
ΔRELDAΔ	Parameter identifier.
IPVSN=s	s = volume serial number of random access device from which the input records originated (one to six characters).
,OPVSN=s	Optional. s = volume serial number of random access device to which the input records are to be reloaded (one to six characters). This is required only when the output volume differs from the input volume.
,CARDS=c/c/....c	Optional. c = card to be reloaded (0-255). A maximum of 10 cards may be specified in each RELDA parameter. Ⓓ
,START=c/y/t	Optional. This indicates the start of the area to be reloaded. c = card number (0-255); used for the 70/568 only. y = cylinder number: 0-202 for 70/564; 0-255 for 70/567; 0-15 for 70/568. t = track number: 0-9 for 70/564; 0-7 for 70/567 and 70/568. (See note 1.) Ⓒ

Reload After-Images (Cont'd)

Entry	Meaning
END=c/y/t	Optional. Indicates the end of area to be reloaded. c = card number (0-255); used for the 70/568 only. y = cylinder number: 0-202 for 70/564: 0-255 for 70/567; 0-15 for 70/568. t = track number: 0-9 for 70/564; 0-7 for 70/567 and 70/568. (See note 1.)
,BAND=nn	Optional. Indicates the number of cylinders in the upper band on the cards in the area to be reloaded (1-15). This is required when reloading a 70/568 by band.
,PDEV=mn/b	Optional. Gives the location of output 70/568 magazine. mn = installation mnemonic b = bin number (0-7) (See note 2.)
,ALTND=c/y/t	Optional. This indicates the right-hand end of the alternate track area for the 70/568. c = card number (0-255) y = cylinder number (0-15) t = track number (0-7) (See note 2.)
,ALTSZ=nnnn	Optional. Indicates the number of tracks in the alternate track area for the 70/568. nnnn = 1-9999 (See note 2.)
,SCALT= YES NO	Optional. Controls assignment of alternate tracks for the 70/568. YES = defective prime tracks are to be assigned alternates on the same card. NO = defective prime tracks are to be assigned alternates in the volume alternate track area. This is the preset option.

Reload After-Images (Cont'd)

Entry	Meaning
,PURGE= YES NO	Optional. Controls purging of the alternate track area for the 70/568. YES = clear the alternate track area of all references to the extent being reloaded. NO = do not clear the alternate track area. This is the preset option.

Notes:

1. If the START= entry is used, the END= entry also must be used.
2. The operands PDEV, ALTND, and ALTSZ are used to give the location of a 70/568 magazine and its alternate track area. They are used for reloading a magazine with a VTOC, that is not considered reliable. All three operands must be used.

Examples:

1. To reload the entire 70/568 volume 000556 back to 000556, clearing the alternate track area prior to reloading:

```
ΔRELDAΔIPVSN=000556,PURGE=YES
```

2. To reload the entire 70/568 volume M back to N assigning "same card alternates" for defective prime tracks:

```
ΔRELDAΔIPVSN=M,OPVSN=S,SCALT=YES
```

3. To reload cards 10, 51, and 66 to 70/568 volume MAG01:

```
ΔRELDAΔIPVSN=MAG01,CARDS=10/51/66
```

4. To reload, the 70/568 volume JDL on device E0, bin number 2; the alternate track area consists of 512 tracks ending at card 255, cylinder 15, track 7:

```
ΔRELDAΔIPVSN=JDL,PDEV=E0/2,ALTND=255/15/7,ALTSZ=512
```

5. To reload cylinders 21 thru 80 on the 70/564 volume 4546.

```
ΔRELDAΔIPVSN=4546,START=21/0,END=80/9
```

6. To reload, the following area in the lower band of 70/568 volume ABC; the lower band consists of cylinders 9-15;

cylinders 10-15 of card 100, all tracks;
cylinders 9-15 of card 101, all tracks;
cylinders 9-11 of card 102, all tracks;
cylinder 12 of card 102, tracks 0-5:

```
ΔRELDAΔIPVSN=ABC,START=100/10/0,END=102/12/5,BAND=9
```

Reload Before-Images

◆ This parameter specifies that before-image records up to a specified termination point, are to be reloaded from tape to one or more random access devices.

Format:

$$\Delta\text{RELDB}\Delta\begin{Bmatrix}\text{DATE=yyddd} & \begin{Bmatrix}\text{,TIME=hhmmss/rrr}\\ \text{,IDENT=uuu/bbbbbb}\end{Bmatrix}\\ \text{CPTNO=iiiii} & \end{Bmatrix}\text{,DIREC=}\begin{Bmatrix}\text{FWD}\\ \underline{\text{REV}}\end{Bmatrix}$$

Entry	**Meaning**
ΔRELDBΔ	Parameter identifier.
DATE=yyddd	Termination date for reloading. yyddd= Julian date; the first digit is the decade, the second the year, and the remaining three the day of the year. (See note 1.)
,TIME= hhmmss/rrr	Optional. Termination "time" for reloading. hhmmss = time of day in format: hours (00-23), minutes (00-59), seconds (00-59). rrr = record number (0-255). (See note 2.)
,IDENT= uuu/bbbbbbb	Optional. This indicates the termination "identifier" for reloading. uuu = user field (one to three characters). bbbbbbb = block number (0-9999999). (See note 2.)
,DIREC=$\begin{smallmatrix}\text{FWD}\\ \underline{\text{REV}}\end{smallmatrix}$	Optional. Indicates the direction that the tape is to be read. FWD = forward. REV = reverse. This is the preset option. (See note 3.)
,CPTNO=iiiii	Termination checkpoint id for reloading. iiiii = checkpoint identification (one to five characters).

Notes:

1. The DATE= entry refers to the "date written" field in the log tape record. (Refer to Dump and Log Record Format, page 7-53.) It indicates that all before-image logs containing a "date written" equal to or greater than specified are to be reloaded to the random access devices.

Reload Before-Images (Cont'd)

2. The TIME= and IDENT= operands refer to the "when written" field in the log tape record. (Refer to Dump and Log Record Format, page 7-53.) These entries in conjunction with DATE= indicate that all records with a date and time or date and identification equal to or greater than specified are to be reloaded to the random access device(s).

3. When reloading before-image logs the tape is normally read reverse, processing the latest logs first and ending up with the earliest log. If the tape has been sorted so that the latest logs appear at the beginning of the tape the DIREC=FWD option must be used.

4. When the CPTNO= operand is used all before-image logs back to the specified checkpoint are reloaded to the random access devices(s).

Examples:

1. To reload all before image logs created on or after Julian date 68034, processing the input tape in a forward direction:

```
ΔRELDBΔDATE=68034,DIREC=FWD
```

2. To reload all before-image logs created after checkpoint CP004.

```
ΔRELDBΔCPTNO=CP004
```

3. To reload all before-image logs created on or after the record written at 1700 hours, 21 minutes, and 39 seconds on Julian date 68079.

```
ΔRELDBΔDATE=68079,TIME=172139/0
```

4. To reload all before-image logs created on or after the record written on Julian date 68101, which contain a user field of 5 and a block number of 3027.

```
ΔRELDBΔDATE=68101,IDENT=5/3027
```

No Alternate

◆ This parameter specifies that no magnetic tape device is available as an alternate to the primary input/output device. If more than one tape is required during dump or reload, the program waits for the tape on the primary device to be changed before continuing. If the parameter is not used the program assumes the availability of an alternate device.

Format:

```
NOALT
```

No Display

◆ This parameter indicates that all subsequent parameters are not to be displayed on the console typewriter. It does not inhibit the display of parameters that contain errors.

Format:

```
NDSPL
```

Checkpoint

◆ This parameter indicates that checkpoints are to be taken at each primary/secondary tape change.

Format:

ΔCKPTΔ SDEV=ssssss

Entry	**Meaning**
ΔCKPTΔ	Parameter identifier.
SDEV=ssssss	Symbolic name of the tape device to which the checkpoints are to be written. ssssss = SYSBU1. This is the primary output device during a dump function. It is invalid during a reload function. If both SYSBU1 and SYSBU2 (the secondary device) are available the checkpoint is written to the current output device. = SYSBU3. This is the work device during certain reload functions. This device may be used though not otherwise required. = SYSBU4. This is the output device to which unreadable blocks are written. This device may be used though not otherwise required.

Note:

If checkpoints are to be taken during a RELDB function, the CKPT parameter must precede the RELDB parameters.

Sequence End

◆ This parameter specifies that the routine is to execute the functions specified by the parameters already processed. After these functions have been completed, processing of parameters continues.

Format:

ΔSQENDΔ

End

◆ This parameter specifies that the routine is to execute the functions specified in the preceding parameters and then terminate.

Format:

ΔENDΔ

Parameter Example

◆ The following example illustrates an input stream to the RADAR routine. It does not necessarily reflect a practical use of the program.

```
ΔNOALTΔ
ΔNDSPLΔ
ΔCKPTΔ SDEV=SYSBU1
ΔDUMPFΔ IPVSN=ALZ,FNAME='MASTADDR'
ΔDUMPVΔ IPVSN=R
ΔSQENDΔ
ΔDUMPVΔIPVSN=S
ΔDUMPFΔ IPVSN=T,FNAME='MASTORD'
ΔSQENDΔ
ΔRELDBΔCPTNO=21
ΔCKPTΔ SDEV=SYSBU3
ΔRELDAΔ IPVSN=J,OPVSN=K
ΔRELDAΔIPVSN=B,CARDS=20/176
ΔENDΔ
```

The first six parameters, through the first SQEND, are read and processed. The entire volume R and the file MASTADDR, which may be composed of a number of extents on a number of random access volumes, are dumped to SYSBU1. NOALT is the only parameter displayed. If ET is reached on SYSBU1, the second and subsequent volumes are also written to SYSBU1.

The next three parameters, through the second SQEND, are then read and processed. A new tape file is created from the data dumped from volume S and the file MASTORD. No parameters are displayed and no checkpoints written. All output is written to SYSBU1 regardless of the number of output volumes.

The next parameter triggers the execution of a reload before image fuction. The before-image logs are read in a reverse direction from SYSBU1 until a ckeckpoint with an ID of "21" is recognized. The NOALT and NDSPL options continue to apply and no checkpoints are taken even if more than one volume of input must be mounted on SYSBU1.

The final four parameters are then executed. All tape records from volume J are reloaded to volume K, while only those tape records from cards 20 and 176 from volume B are reloaded to B. Any number of input tapes containing dump or log records may be used as input to the reload. A checkpoint is written to SYSBU3 each time a new input volume is mounted. NOALT and NDSPL apply. When no more tapes are to be read, the program terminates.

Considerations For Use

◆ Before using the RADAR routine, the following points should be considered:

On-Line Catalog

◆ The On-Line Catalog console routine (EΔOLC) must be run before executing the program.

Parameter Format

◆ The basic parameter format is as follows:

a. space
b. parameter identifier
c. space
d. operands
e. space
f. comments if desired, or spaces

The parameter identifier specifies the function to be performed.

Operands are identified by a keyword followed by an equal sign (=). Operands are separated by a comma. Operands not pertinent to the desired function should not be present. Their absence must not be indicated by commas.

When parameters are entered on cards a space following an operand signifies that no further information is contained on that card. A nonblank in column 72 indicates that additional operand information is contained on a continuation card. Parameter information on continuation cards should not begin before column 16.

Parameters entered through the console typewriter are terminated by a space following an operand. A maximum of 70 characters may be entered at one time. If a nonspace appears in position 70 of the typein, a message requests any additional parameter information.

Run-Time Parameters

◆ Tapes created by this routine are in standard Spectra 70 format and contain standard volume and header labels. If no label parameter (VOL and TPLAB) are supplied, RADAR generates or checks for a standard file Header label with the file identification: RADARΔTAPEΔFILEΔn, where n is 1, 3 or 4 depending upon the symbolic device name (SUSBU1, BYSBU3, or SYSBU4,). When VOL and TPLAB parameters are supplied, the VOL parameters must be written as follows:

```
// VOL,SYSBU1,BU1FILE
// VOL,SYSBU3,BU3FILE
// VOL,SYSBU4,BU4FILE
```

When a multivolume 70/564 or 70/567 file is being dumped using the dump file (DUMPF) function, aVDC parameter must be supplied. The following rules must be observed:

a. The Filename operand must be RADARDF.

b. The Matrix operand may be omitted.

c. The Fileid operand must be the same as the FNAME= operand in the DUMPF parameter.

d. A volume serial number must be supplied for each device that contains a portion of the file to be dumped and is on line at the initiation of the DUMPF.

Run-Time Parameters (Cont'd)

Examples:

File ISDISC is contained on the 70/564 volumes A and B. The VDC parameter should be:

```
// VDC RADARDF,,ISDISC,A,B
```

Dump and Log Record Format

◆ Dump and log records are recorded on tape as variable-length, unblocked records in the format described below.

<table>
<tr><th>Bytes</th><th>Content</th><th colspan="2">Meaning</th></tr>
<tr><td>1-4</td><td>Block length (BL)</td><td colspan="2">This indicates the length of the block including this field.</td></tr>
<tr><td>5-8</td><td>Record length (RL)</td><td colspan="2">This indicates the length of the record including this field.</td></tr>
<tr><td>9</td><td>Before/After indicator (BAI)</td><td colspan="2">$B0_{16}$=before-image log.
$A0_{16}$=after-image log.
$A1_{16}$=sorted after-image log.
$A2_{16}$=dump record.</td></tr>
<tr><td>10-15</td><td>Volume Serial No. (VSN)</td><td colspan="2">This is the serial number of the random access device from which this dump or log record originated.</td></tr>
<tr><td>16-19</td><td>Home Address (HA)</td><td colspan="2">This is the prime track address (CCHH) of the dump or log record. When logging 70/568 tracks from the alternate track area for alternate area backup, this field contains the alternate track address.</td></tr>
<tr><td>20</td><td>Record Number (RN)</td><td colspan="2">This is the record number (R) of the dump or log record.</td></tr>
<tr><td>21</td><td>Device Type (DT)</td><td colspan="2">This is the device type as it appears in the Executive Device List.</td></tr>
<tr><td rowspan="3">22</td><td rowspan="3">How Written (HW)</td><th>Bit</th><th>Meaning</th></tr>
<tr><td>2^0-2^3</td><td>Not used.</td></tr>
<tr><td>2^4</td><td>If 1, this indicates that the block was written with a format write (if both 2^4 and 2^5 are 0, the existing block was updated.)</td></tr>
</table>

Dump and Log Record Format (Cont'd)

Bytes	**Content**	**Meaning**	
		2^5	If 1, this indicates that the block was written as an update but the key was missing.
		2^6	Not used.
		2^7	If 1, this indicates that both R0 and R1 are included in this record.
23-24	Unused Bytes (UB)	This indicates the number of unused bytes for this track as taken from bytes 6 and 7 of the R0 record. This field is always set to $(0000)_{16}$ except for 70/564 and 70/567 log records that involve physical record addition.	
25-29	Date Written (DW)	This date indicates when the dump or log was written (YYDDD).	
30-36	When Written (WW)	This is a seven-byte field applicable to log records only. If LOGTOD=YES was specified in the DTFIS, this field has the following format: HHMMSSN where HHMMSS=hours (00-23) minutes (00-59), and seconds (00-59). N=number of times a track is logged within a given HHMMSS clock value (binary 0-255). If LOGTOD=NO was specified in the DTFIS, this field has the following format: UUUNNNN where UUU = the user's id added before the record is logged. When not used, this is hex zeros. NNNN = the block number (1-9999999 packed decimal).	
37	Length of User's Field (LUF)	This is a binary value that is one greater than the number of bytes in the user's field. The length of the user's field is specified inthe DTFIS operand LOGLEN=n. LUF is equal to n+1 and may range from 1 to 219.	

Dump and Log Record Format (Cont'd)

Bytes	Content	Meaning
Variable	User Field (UF)	This is a variable-length field (0-218 bytes) containing user data. If LOGLEN= n was not specified in the DTFIS, this field is zero in length.
Variable	Count Key, data (CKD)	This is a variable-length field containing the count, key (if any), and data of the dump or log record. This field contains the standard 16 byte R0 and the R1 CKD when they both exist.

Processing Before - Image Records

◆ The RELDB function reads input from magnetic tape files. The tape files may contain various combinations of before- and after-image log records, checkpoint records, and user records.

The RELDB routine normally reads magnetic tape in the reverse direction, thus applying before images in the reverse sequence in which they were written. All before images encountered are applied until a parameter specified termination point is reached. Possible termination points are: the beginning of the file (the entire file is to be reloaded); a specific checkpoint number; or, a specific record. A specific record may be identified by the value of the "when written" field that contains either a block number or a time clock value.

When opening a tape for reverse reading (the standard use of the RELDB function). This routine rewinds one block and senses for the Beginning Tape. If not set, a tape is assumed to have been positioned for reloading prior to entrance to RELDB. (It is expected that this situation occurs when RELDB is executed immediately after system faliure.) The tape is then positioned forward one gap. If BT is sensed, the standard FCP OPEN is performed and the tape is positioned forward to the tape mark that follows the last data record. The tape is then positioned to the front of this tape mark.

Before-image logs are reloaded as encountered with no volume restrictions. When the prime track is coded defective, logs are written to the previously assigned alternate track. Newly defective prime tracks cause the RELDB function to dynamically locate an available track in the volume's alternate track area and to record the before-image record on that track (70/568 only). Same card alternate tracks are not assigned.

If the before images have been sorted so that the latest logs appear at the beginning of the tape,the DIREC=FWD option must be used in the RELDB statement.

Whenever before-image tapes are reloaded, the tapes must be given to the routine such that the earliest log is processed last.

Processing Dump and After-Image Records

◆ The RELDA function reads input from magnetic tape files. The tape files may contain various combinations of before- and after-image log records, checkpoint records, and user records. If the tape was produced by the DUMPV or DUMPF function of RADAR, it will contain dump records (in after-image log format) and optionally, checkpoint records. A tape file may also consist of sorted after-image logs. The reload process is based on the following assumptions:

1. A tape file containing dump or sorted logs is in volume serial number and random access address sequence.
2. Dump records (those produced by the dump function) from more than one tape file may be processed during a single reload function. However, dump records from only one tape file can satisfy reload requirements for a given physical (as opposed to file) random access extent.
3. If more than one sorted log file is processed in a single function, the files must be processed in the order created. Within each file, records must be in volume serial number and random access address sequence. Nevertheless, records need not be in "date written" and "when written" sequence. (The Reload function assumes that where duplicate after-image logs - equal VSN and CCHHR fields - exist they are sequenced such that the last log created is the last one to be reloaded.) However, the lowest values in the "date written" and "when written" fields of a given file must be greater than the highest values in the corresponding fields of any record of any file previously reloaded.
4. Unsorted after-image log records must be in sequence according to their "date written" and "when written" fields. These fields in the first record of an unsorted log file must be greater in value than the value contained in the corresponding fields of any record previously reloaded. Likewise, these fields in the last record of an unsorted after-image log file must contain a value that is less than that in any record of any file still to be loaded.

When reloading to a 70/565 disc or a 70/567 drum, dynamic alternate track assignment is not performed. Regardless of the type of records being reloaded (dump or log), records are written to the alternate tracks previously assigned by the Random Access Volume Initializer.

When reloading to the Model 70/568 Mass Storage Unit, a parameter may specify that any existing track in the alternate track area that references the extents to be reloaded be immediately "purged" (initialized and made available for subsequent assignment as alternate tracks). Reloading records created by the dump function causes dynamic assignment of alternate tracks when prime tracks are found to be defective. Reloading log records to tracks that have been coded defective, causes the log records to be written to the assigned alternate tracks.

Processing Dump and After-Image Records (Cont'd)

If the alternate track area is being reloaded or if the assignment of same card alternate tracks was requested (both 70/568-only functions), the RELDA function uses SYSBU3 as a work tape for temporary storage. This stores those tape records that cannot be written to prime tracks because the prime tracks are defective. When all records for all reload extents have been processed, the temporarily stored records are retrieved from SYSBU3 and written to available alternate tracks.

It should be noted that the process of reconstructing an extent and dynamically assigning alternates renders previous alternate area dumps and logs obsolete. A standard installation practice should be to dump affected alternate areas immediately after reloading.

The RELDA function terminates when the console operator indicates that all input tapes have been processed

When a full volume is copied to tape and subsequently reloaded, the VTOC of the input volume is included in the copy, thereby creating volumes with duplicate volume serial numbers. If this type of copy is not desired, the VTOC tracks must be eliminated through the use of START= and END=operands.

Use of SYSBU4

◆ SYSBU4 is an optional magnetic tape to which unreadable tape or random access blocks are written during the dump or reload functions. Blocks that are unreadable because of Read Parity errors or Transmission Parity errors are written to SYSBU4. Such blocks can be subsequently repaired and reloaded using the reload function of the RADAR routine.

Device Assignments

◆ Under Monitor or Executive;

SDN	Device Type	Remarks
SYSIPT	Card reader (may be magnetic tape under Monitor only).	Parameter input.
SYSBU1	Magnetic tape.	Primary input or output device. When used as output it may be used as a checkpoint device.
SYSBU2	Magnetic tape.	Alternate input or output device. When used as output it may be used as a checkpoint device.
SYSBU3	Magnetic tape.	Work tape used during certain reload functions. It also may be used as a checkpoint device.
SYSBU4	Magnetic tape.	Optional tape to which unreadable magnetic tape or random access blocks are written. It also may be used as a checkpoint device.

User Replacement of ITURTP and ITURTO Modules

◆ RADAR's standard tape input module (ITURTP) is required for reload functions, and its standard tape output module (ITURTO) required for dump function. Both can be replaced by user-coded modules. This replacement feature allows the user to maintain his log/dump records on a device other than a tape, and provides a means of handling log/dump records not in RADAR format.

Replacement of either (or both) of these modules demands satisfaction of certain interface criteria as the replacement module(s) must function with standard RADAR control modules.

The standard input/output tape devices SYSBU1 and SYSBU2 are not utilized by RADAR when either the standard ITURTP or ITURTO modules are replaced by user modules. This means:

1. The user replacement modules must open, close, and swap (as necessary) input/output device(s) which replace SYSBU1 and SYSBU2. The ITURTA module will process SYSBU3 and SYSBU4 as required.

2. The parameter CKPT SDEV=SYSBU1_ is illegal. Designation of SYSBU3 or SYSBU4 causes the specified tape to be opened, even if the device is not otherwise required by the function being executed.

3. The user is responsible for execution of the checkpoint macro; this will be accomplished in his replacement module(s). The "restart-name" and "error routine-name" operands of the CKPT macro must reflect routines incorporated in the user's replacement module(s).

4. If the user desires tape positioning and label checking at restart time, he may utilize the "DTF-address" operand of the CKPT macro (Fields ITURDT2 and ITURDT3, in the root segment, contain the first 112 bytes of DTFPH macro expansion for SYSBU3 and SYSBU4, respectively.) To position the user tape which replaces SYSBU1, the "DTF-address" operand names the user tape DTFPH. If SYSBU3 and SYSBU4 are to be repositioned, the user must define dummy DTFPH's following his user tape DTFPH, and move the information from ITURDT2 and ITURDT3 to the left-most-end of these dummy DTFPH's. The dummy DTFPH's must each be 192 bytes in length.

INDEXED SEQUENTIAL ANALYSIS/ REORGANIZATION (ISAR)

General Description

◆ The Indexed Sequential Analysis/Reorganization Program (ISAR) is designed to assist the user in identifying his Model 70/568 mass storage unit file organization problem areas and in removing the associated inefficiencies. Generally, ISAR analyzes a Model 70/568 indexed sequential file and identifies those areas which are potentially troublesome, and at the user's request, restructures the problem areas to remove the inefficiencies.

Preset Functions

◆ None.

Optional Functions

◆ This routine provides the following options:

1. Analyzes overflow extents and alternate track areas to indicate the area available for future requirements.
2. Identifies Model 70/568 cards that have contributed records to the overflow extents and alternate track areas.
3. Analyzes individual prime data cards and determines the number of unassigned tracks, the number of tracks assigned to overflow records, and the number of tracks assigned as alternates.
4. Determines the number of cylinders and tracks available for file expansion, that is, for additional primary data and associated indexes.
5. Reorganize an entire Model 70/568 indexed sequential file, a specified extent, or contiguous named extents.

Input

◆ An indexed sequential Model 70/568 file.

Output

◆ ISAR produces three types of output:

1. Indexed Sequential Model 70/568 Files.
2. Printer (Console) Analysis Information.
3. Console Message and Error Typeouts.

Equipment Configuration

Required

◆ Processor (65K).

Console Typewriter.

Mass Storage Unit (Model 70/568).

Optional

◆ The card reader is used for parameter input, but can be replaced by the Console, Model 70/97.

A magnetic tape device, 9-level or 7-level, with pack/unpack feature is required for reorganization.

A disk storage unit (Model 70/564) or drum storage unit (Model 70/565) is required for reorganization.

Routine Parameters - General

◆ Except for the END parameter, all routine parameters are optional and need only be supplied when various functions are desired.

ANALF Parameter

ANALF supplies statistical data on designated Model 70/568 indexed sequential files whose file labels appear on specified discs or drums.

ANALV Parameter

ANALV supplies alternate track statistics for designated Model 70/568 volumes (magazines).

ANALD Parameter

The ANALD statement directs ISAR to perform a detailed analysis of the designated prime data area.

REORG Parameter

The REORG statement directs ISAR to reorganize an entire Model 70/568 I/S file, a specified extent, or contiguous named extents.

Sequence of Parameters

◆ The following sequence of parameters must be maintained regardless of which options are selected:

NOALT
NDSPL
ANALF
ANALV
ANALD
REORG
END

Routine Parameters - Detailed

NOALT

◆ This parameter indicates that a second magnetic tape device is not available as an alternate to the primary magnetic tape input/output device.

Format:

ΔNOALTΔ

Notes:

1. If more than one tape is required during the execution of ISAR, the program will wait for the operator to change the tape on the primary device (SYSBU1) before accessing the second volume.
2. If this parameter is omitted, the program assumes the availability of another magnetic tape (SYSBU2) and automatically alternates between SYSBU1 and SYSBU2.

NDSPL

◆ This parameter inhibits the recording of parameters having no errors on the console typewriter.

Format:

ΔNDSPLΔ

ANALF

◆ The File Information Analysis (ANALF) statement indicates that the file labels, indexes, and overflow areas associated with specified Model 70/568 indexed sequential files are to be analyzed to forewarn of any impending file overflow condition that could disable the system.

Format:

ΔANALFΔFNAME = 'file-id',IPVSN = VVVVVV,LIST = $\left\{ \frac{\text{NO}}{\text{YES}} \right\}$

Entry	**Meaning**
ΔANALFΔ	Parameter identifier.
FNAME='file-id',	Specifies the name of the Model 70/568 indexed sequential file that is to be analyzed. The name can be from 1 to 44 characters in length and must be enclosed within apostrophes.
IPVSN=VVVVVV	Specifies the Volume Serial Number of disc or Drum containing the file definition. If this entry consists of less than 6 characters, it will be right justified and zero (X'F0') filled to the left.
,LIST= $\left\{ \frac{\text{NO}}{\text{YES}} \right\}$	Optional. Indicates whether a printer listing is to be performed for the Model 70/568 prime data cards that have contributed records to overflow extents. If this operand is not specified, NO is assumed, and output is written to either the printer or to the console typewriter.

Note:

The 'file-id' must match the filename in the Format 1 record of the VTOC.

ANALV

◆ The ANALV statement directs ISAR to scan the alternate track areas on the specified Model 50/568 volume(s) to determine the area available for subsequent track assignments.

Format:

$$\text{ANALV IPVSN=vvvvvv}\left[/\text{vvvvvv}/\ldots\ldots\ \text{v}\right],\text{LIST}=\begin{Bmatrix}\underline{\text{NO}}\\ \text{YES}\end{Bmatrix}$$

Entry	**Meaning**
ANALV	Parameter identifier.
IPVSN = vvvvvv/ v	Specifies the volume serial number of the Model 70/568 volume(s) whose alternate areas are to be analyzed. The v entry is the 1 to 6 character alphanumeric volume serial number and is separated from the next volume serial number by a virgule ("/"). Up to eight volume serial numbers can be specified.
$,\text{LIST}=\begin{Bmatrix}\underline{\text{NO}}\\ \text{YES}\end{Bmatrix}$	Optional. Indicates whether the listing of prime cards which have contributed to the alternate track area is required. If the LIST operand is omitted, NO is assumed.

Note:

ISAR indicates the remaining space in the alternate track area.

Example:

ΔANALVΔIPUSN = A1234/A1235/A1236,LIST = YES

ANALD

◆ The ANALD statement directs ISAR to perform a detailed analysis of the designated prime data area.

Format:

$$\Delta\text{ANALD}\Delta\text{IPVSN = vvvvvv,TRKAN = }\begin{Bmatrix}\underline{\text{YES}}\\ \text{NO}\end{Bmatrix}\left[\begin{Bmatrix}\text{CARDS=c [/c/ . . . c]}\\ \text{START=X, END=Y}\end{Bmatrix}\right]$$

ANALD (Cont'd)

Entry	Meaning
ANALD	Parameter identifier.
IPVSN = vvvvvv	Identifies the Model 70/568 volume to be analyzed. The entry v is the (1 to 6 alphanumeric characters) volume serial number.
TRKAN= $\left\{\begin{matrix}\underline{\text{YES}}\\ \text{NO}\end{matrix}\right\}$	Optional. When the NO value is supplied, it indicates that the ANALD function is restricted to a check of card extract counters. Individual tracks are not analyzed. If the TRKAN operand is omitted, or if TRKAN = YES is specified, the entire detailed analysis, including the card extract analysis, is performed.
CARDS = c [/c/ . . . c]	Indicates that each of the specified Model 70/568 cards is to be analyzed. Each c represents a specific card number (0-255), and is separated by a virgule ("/") from the next card number. A maximum of 10 cards may be specified in a single ANALD statement.
START = x	Indicates that a series of consecutive cards beginning with card number x (0-255) is to be analyzed. The entry x is a one to three digit decimal number.
END = y	Indicates that card number y (0-255) is the last in the series of consecutive cards to be analyzed. The entry y may not be less than the x value expressed in the START=operand. The entry y is a one to three digit decimal number.

Notes:

1. When neither "CARD=" nor "START=,END=" is specified, this function is preset to "START=0,END-255."

2. If card 0 is being analyzed, the 8 tracks in cylinder 0 are considered available tracks.

3. Overflow descriptor tracks are considered overflow tracks and "CYLOF" tracks (that is, H1 of CCH equals $(14)_{16}$) are considered available CYLOF tracks.

ANALD (Cont'd)

Examples:

1. ΔANALDΔIPVSN=A1234,START=50,END=100

   ISAR will analyze cards 50 through 100 on volume A1234.

2. ΔANALDΔIPVSN=A1234,CARDS=10/20/23/85

   ISAR will analyze cards 10, 20, 23, and 85 on volume A1234.

3. ΔANALDΔIPVSN=A1234

   ISAR will analyze cards 0 through 255 on volume A1234.

4. ΔANALDΔIPVSN=A1234,TRKAN=NO,START=50,END=100

   ISAR will analyze the extract counter only on cards 50 through 100. The output listing reflects only those cards whose extract counters have been exhausted.

REORG

◆ The REORG statement directs ISAR to reorganize an entire Model 70/568 indexed sequential file, a specified extent, or contiguous named extents. When less than an entire file is reorganized, the extents which are to receive the reorganized data must be explicitly specified.

Format:

ΔREORGΔIPVSN=vvvvvv,FNAME='file-id',CYLOF=n,ISDEN=n
,IPEXT=vic [/vic .../vic] ,OPEXT=vic [/vic .../vic]

Entry	Meaning
REORG	Parameter identifier.
IPVSN=vvvvvv	Specifies the volume serial number of the disc or drum that contains the Format 1 VTOC of the Model 70/568 file which is to be reorganized. The entry v is the volume serial number (1 to 6 alphanumeric characters).
FNAME='file-id'	Specifies the name of the Model 70/568 indexed sequential file to be reorganized. The name can consist of from 1 to 44 alphanumeric characters and must be enclosed within apostrophes. When less than 44 characters are specified, trailing spaces are assumed.

REORG
(Cont'd)

Entry	Meaning
CYLOF = n	Specifies an absolute integer which is the number of tracks in each prime data band that are to be reserved for overflow and alternates. Prime data sentences are not written to these tracks. The value of n may range from 0 to the number of tracks in the band, minus 1 (PTI on disc/drum), or to the number of tracks in the band minus 2 (PTI on Model 70/568). If this operand is omitted, a value of 0 is assumed. This operand corresponds directly to the DTFIS macro operand of the same name.
ISDEN=n	Specifies an absolute integer which is the percentage of each primary data track to be filled while loading the reorganized file. If not specified, 100 is assumed. This operand corresponds directly to the DTFIS macro operand of the same name.
IPEXT=vic [/vic.../vic]	Specifies the extent or series of contiguous extents. When a series of extents is to be reorganized, each extent in the series must be specified in the order established within the Format C VTOC. Each v:c pair describes an extent. The entry v is the volume serial number (1 to 6 alphanumeric characters) of the volume containing the extent, and the entry c is the card number (0-255) of the first card in the extent.
OPEXT=vic [/vic... /vic]	Specifies the extents that are to receive the reorganized data. The number of output extents can exceed or be exceeded by the number of input extents. They can encompass the same physical areas or can be entirely different areas; however, in the latter case, the input extents should subsequently be purged from the file by the random access allocator. Each extent in the output area must be described by a v:c pair. The output extents should be described in the logical order in which they are to be used. ISAR modifies the extent list within the Format C-VTOC to correspond to the order established within the OPEXT operand. The entry v is the volume serial number (1 to 6 alphanumeric characters) of the volume which contains the extent, and the entry c is the card number (0-255) of the first card in the extent.

REORG
(Cont'd)

Notes:

1. Reorganization of two or more non-contiguous extents requires preparation of a REORG statement for each break in continuity.

2. If the IPEXT= and OPEXT= operands are not specified, the entire file is reorganized. If reorganization by extent is desired, both the IPEXT= and OPEXT= operands must be specified.

Examples:

1.
   ```
   ΔREORGΔIPVSN=CL,FNAME='FICA',IPEXT=M1:10/M2:20,
   OPEXT=M1:10/M2:20/K1:30
   ```

   The file FICA, defined on volume CL, requires partial reorganization. The extent that begins with card 10 on volume M2 is to be reorganized. The reorganized data is to be returned to the extents that begins with cards 10 on volume M1, and 20 on volume M2. In addition, a new extent that begins with card 30 on volume K1, has been assigned to provide space for the many overflow records in the input extents which were responsible for the reorganization. The reorganized prime data wlll "spill over" onto the new extent that begins with card 30 on volume K1.

2.
   ```
   ΔREORGΔIPVSN=INDSEQ,FNAME='MASTERACCT',CYLOF=12
   ```

   The file MASTERACCT, defined on volume INDSEQ, is to be fully reorganized. Additional prime data extents necessary to accommodate the reorganized file were allocated prior to the execution of ISAR.

   Twelve tracks of each prime data band are to be reserved for overflow and alternate tracks. The original file may have been loaded with a CYLOF value other than 12.

END

◆ The END statement indicates that ISAR has processed all input parameters and should terminate processing.

Device Assignment

◆ Input/output random access volumes are specified by parameter (Volume Serial Number) or obtained from appropriate file labels. The Update On-Line Catalog routine must be run prior to ISAR.

Devices	ANALD	ANALF	ANALV	REORG	Remarks
SYSBU1	*	*	*	X	Primary input/ output for REORG.
SYSBU2	*	*	*	*	Alternate to SYSBU1 for REORG.
SYSBU3	*	*	*	X	Storage for indexes, checkpoint, and Format C.
SYSLST	*	*	*	X	Required for ANALF when NAALF when LIST=YES.
SYSIPT	X	X	X	X	Cards, tape, or disc.

* = Optional.   X = Required.

Notes:

1. The Console option is selected by the console reply of "NO" to the Executive request for SYSIPT assignment. ISAR then requests console input.
2. For operation under the Monitor, SYSLST must be assigned either to the printer, a magnetic tape unit or a disc storage unit. Program parameters are always written to SYSLST under Monitor unless inhibited by the NDSPL option.
3. A parameter input device is optional under the Executive. If parameters are not entered at the console, SYSIPT must be assigned to a card reader. Under the Monitor, SYSIPT may be assigned to the card reader, a magnetic tape unit or to a disc storage unit.

Printer Output

◆ The primary display device for analysis information is the printer. The limited information produced by the ANALF and ANALV functions, when analysis details are not desired, can be optionally displayed on the console typewriter.

ANALD Listing

◆ The detailed analysis listing consists of 1 print line for each card analyzed. The line contains 13 columns:

MAG VSN - The magazine volume serial number.

CD# - The Model 70/568 card number.

PRIME - The number of prime data tracks.

ALTER - The number of alternate tracks of all types.

DEFEC - The number of defective tracks. This value is not necessarily equal to the ALTER value, since some alternate tracks can be in the CYLOF area and some in the volume's alternate track area.

OVFLW - The number of overflow tracks, including the overflow descriptor track.

ANAIL - The number of unused tracks in the prime and data areas, as well as unallocated tracks and the 8 tracks in cylinder 0 when card 0 is analyzed.

PTI - The number of prime track index tracks.

EOF - The number of end-of-file tracks (1 track per file).

PD ALT - The number of alternate prime data tracks.

OVF ALT - The number of overflow alternates.

PTI ALT - The number of prime track index alternates.

EOF ALT - The number of end of file alternates.

The last 4 columns are a breakdown of the fourth (ALTER) column. The third through ninth columns must total 128.

Figure 7-1 was produced by the parameter:

ΔANALDΔIPVSN=2,START=0,END=4

```
                           I.S. 70/568 DETAILED ANALYSIS

MAG VSN   CD #   PRIME   ALTER   DEFEC   OVFLW   AVAIL   PTI   EOF   PD ALT   OVF ALT   PTI ALT   EOF ALT

000002    000    48                        2       72    6

000002    001    53                                68    7

000002    002    53                                68    7

000002    003    53                                68    7

000002    004    53                                68    7
```

Figure 7-1. Example of Indexed Sequential Model 70/568 Detail Analysis

ANALF Listing

◆ The file analysis listing consists of:

1. Three file-level print lines, which contain:
    a. The number of prime data and overflow records as appears in the file's Format 2 label.
    b. The available prime tracks, as calculated from the end of file address in the Format 2 label. All available tracks in the end of file extent, plus all tracks in other unused allocated prime data extents, are included.
    c. The available index tracks, as calculated from the first track of the cylinder following the highest level index. This index address is also extracted from the Format 2 label.
2. Three volume-level print lines that appear whether LIST=YES or LIST=NO. All 3 lines reflect track totals, which are:
    a. The available index area.
    b. The available prime area.
    c. The available overflow area. This value is obtained by physically examining each track in the overflow extent(s).

The remaining lines appear only if LIST=YES and indicate the prime data cards that have contributed records to the overflow area and the number of records each card had contributed.

Figure 7-2 was produced by the ANALF function with LIST=YES specified.

```
                        I.S. 70/568 FILE ANALYSIS

FILE TOTALS: PRIME DATA RCDS 00000700 OVFLW DATA RCDS 00008

             AVAIL PRIME TKS 00003353

             AVAIL INDEX TKS 00040

AVAIL INDEX AREA IN VSN 00000A : 00040

AVAIL PRIME AREA IN VSN 000002 : 03353

AVAIL OVFLW AREA IN VSN 000002 : 00763

CARDS IN VOLUME 000002 CONTAINING OVERFLOW RECORDS AND NUMBER OF OVERFLOWS:


CRD #R  CRD #R  CRD #R  CRD #R  CRD #R  CRD #R  CRD #R  CRD #R  CRD #R  CRD #R  CRD #R

CRD #R

200 0001    201  0006   210 0001

END OF REPORT
```

Figure 7-2. Example of Indexed Sequential Model 70/568 File Analysis

ANALV Listing

◆ The volume analysis listing (ANALV) consists of a line containing the number of unused tracks in a volume's alternate track area that is always printed. Additional information which specifies the cards that have contributed to the Alternate Track Area and the number of tracks on each card is produced when LIST=YES.

Figure 7-3 was produced by the action of ISAR on the parameter:

```
ΔANALVΔIPVSN=2,LIST=YES
```

The Reorganization listing is produced to show the number and order of prime data extents before and after a file reorganization. If an entire file is reorganized, the extents are unchanged. The information is obtained from Format C labels as they appeared before the REORG function, and after they were adjusted during the reorganization process.

Figure 7-4 is a listing obtained during the partial reorganization of a file. Six prime and three overflow extents are allocated to the file, in the sequence shown at the left side of the listing. The second prime extent, beginning at card 11 of VSN 2 and extending through card 20, was reorganized into the third and fourth extents, beginning at cards 25 and 35. The right side of the list shows the file extents as adjusted during reorganization. The file has been altered to occupy cards 0-9 (unchanged by the REORG), continues at card 25, and then follows in the extent beginning at card 35. The records that were in the second extent (cards 11-20) are now in the original third (cards 25-29) and fourth (cards 35-39) extents. The original second extent is now last and can be reallocated if the area is required for some other file.

The parameter which generated the reorganization indicated in Figure 7-4 is:

```
ΔREORGΔIPVSN=A,FNAME='REORG TEST',IPEXT=
2:11,OPEXT=2:25/2:35
```

```
                              I.S. 70/568 VOLUME ANALYSIS

0374 UNUSED TRACKS IN ALTERNATE AREA OF VOLUME 000002

CARDS IN VOLUME 000002 CONTAINING FLAW TRACKS AND NUMBER OF FLAWS:

CRD #F  CRD #F  CRD #F  CRD #F  CRD #F  CRD #F  CRD #F  CRD #F  CRD #F  CRD #F  CRD #F  CRD #F  CRD #F  CRD #F

033 006 051 004
```

Figure 7-3. Example of Indexed Sequential Model 70/568 Volume Analysis

```
                                  I.S. 70/568 FILE REORGANIZATION

          FORMAT C EXTENTS BEFORE ADJUSTMENT                                FORMAT C (EXTENTS AFTER ADJ)

VOL-SER-NO  TYPE  LHE-CD/CYL RHE-CD/CYL ALT-TK-AREA IPEXT   VOL-SER-NO  TYPE  LHE-CD/CYL RHE-CD/CYL ALT-TK-AREA IOEXT

  000002   PRIME     0   0       9   7    253 - 255          000002   PRIME     0   0       9   7    253 - 255

  000002   PRIME    11   0      20   7    253 - 255    1     000002   PRIME    25   0      29   7    253 - 255    1

  000002   PRIME    25   0      29   7    253 - 255          000002   PRIME    35   0      39   7    253 - 255    2

  000002   PRIME    35   0      39   7    253 - 255          000002   PRIME    41   0      45   7    253 - 255

  000002   PRIME    41   0      45   7    253 - 255          000002   PRIME    51   0      55   7    253 - 255

  000002   PRIME    51   0      55   7    253 - 255          000002   OVFLW    10   0      10   7    253 - 255

  000002   OVFLW    10   0      10   7    253 - 255          000002   OVFLW    21   0      24   7    253 - 255

  000002   OVFLW    21   0      24   7    253 - 255          000002   OVFLW    30   0      34   7    253 - 255

  000002   OVFLW    30   0      34   7    253 - 255          000002   PRIME    11   0      20   7    253 - 255
```

Figure 7-4. Example of Indexed Sequential Model 70/568 File Reorganization

Run Time Parameters

VDC

◆ A VDC parameter is mandatory for normal indexed sequential processing of the file being reorganized. The matrix size, which is expressed in the VDC parameter, is:

$$12(1+e) + 17(1+n) +2$$

where:

e = the number extents in the file.

n = the highest volume sequence number in the file -1.

The Filename operand must contain ITURDIL. If an ANALF is run, the following VDC specifications must be observed:

1. The Filename operand can contain any legitimate value.
2. The Matrix operand may be omitted.
3. The Fileid operand value must correspond to the FNAME operand of the ANALF statement.
4. One volume serial number must be supplied for each disc/drum volume containing index extents.

VOL/TPLAB

◆ VOL/TPLAB parameters may be supplied by SYSBU1 (SYSBU2) during a reorganization function. If supplied, the File Name field of the VOL card must be ITURBU1.

Linkage Editor Parameters

◆ The following are the linkage edition parameters needed to combine the routines object modules:

```
ΔPROGΔPRISM,VERvvvΔ
ΔNCALΔ
ΔNOCTLΔ
ΔXREFΔ
ΔINCLUDEΔSYSUT1(ITURCM)Δ
ΔOVERLAYΔNODE1,ITURVAΔ
ΔINCLUDEΔSYSUT1(ITURVA)Δ
ΔOVERLAY,NODE1,ITURAD
ΔINCLUDEΔSYSUT1(ITURAD)Δ
ΔOVERLAYΔNODE1,ITURAF
ΔINCLUDEΔSYSUT1(ITURAF)Δ
ΔOVERLAYΔNODE1,ITURAVΔ
```

Linkage Editor Parameters *(Cont'd)*

```
ΔINCLUDEΔSYSUT1(ITURAV)Δ
ΔOVERLAYΔNODE1,ITURDIΔ
ΔINCLUDEΔSYSUT1(ITURDI)Δ
ΔOVERLAYΔNODE2,ITURDUΔ
ΔINCLUDEΔSYSUT1(ITURDU)Δ
ΔOVERLAYΔNODE3,ITURGXΔ
ΔINCLUDEΔSYSUT1(ITURGX)Δ
ΔOVERLAYΔNODE4,ITURGIΔ
ΔINCLUDEΔSYSUT1(ITURGI)Δ
ΔOVERLAYΔNODE4,ITUROTΔ
ΔINCLUDEΔSYSUT1(ITUROT)Δ
ΔOVERLAYΔNODE3,ITUROCΔ
ΔINCLUDEΔSYSUT1(ITUROC)Δ
ΔOVERLAYΔNODE3,ITURCEΔ
ΔINCLUDEΔSYSUT1(ITURCE)Δ
ΔOVERLAYΔNODE2,ITURDIΔ
ΔINCLUDEΔSYSUT1(ITURDL)Δ
ΔOVERLAYΔNODE5,ITURITΔ
ΔINCLUDEΔSYSUT1(ITURIT)Δ
ΔOVERLAYΔNODE5,ITURJTΔ
ΔINCLUDEΔSYSUTL(ITURJT)Δ
ΔOVERLAYΔNODE5,ITURKTΔ
ΔINCLUDEΔSYSUT1(ITURKT)Δ
ΔENTRYΔITURCMODΔ
```

General Description
(Cont'd)

Module		Peripheral Routine				
Section	**Type**	**CDRA**	**RAPR**	**RARA**	**RATP**	**TPRA**
Root Segment	Control Block	ITUTPR1	ITUTPR4	ITUTPR5	ITUTPR3	ITUTPR2
	Own Code	*	*	*	*	*
	Control Routine	ITUTP7	ITUTP7	ITUTP7	ITUTP7	ITUTP7
Overlay 1	Scan-Validate	ITUTP8	ITUTP8	ITUTP8	ITUTP8	ITUTP8
Overlay 2	F-S Generation	ITUTP9	ITUTP9	ITUTP9	ITUTP9	ITUTP9
Overlay 3	Specific Val	ITUTPC4	ITUTPC8	ITUTPC7	ITUTPC6	ITUTPC5
	Buffer Comp & Logging	ITUTPE	ITUTPE	ITUTPE	ITUTPE	ITUTPE
Overlay 4	Input	ITUTP0	ITUTP4	ITUTP4	ITUTP4	ITUTP2
	Output	ITUTP5	ITUTP6	ITUTP5	ITUTP3	ITUTP5
Region 2	I/O Buffers	ITUTPB	ITUTPB	ITUTPB	ITUTPB	ITUTPB
	F-S Storage	ITUTPF	ITUTPF	ITUTPF	ITUTPF	ITUTPF

Section	Type	Peripheral Routine		
		CDRAM	**TPRAM**	**RARAM**
Root segment	Control Block	ITUTPR6	ITUTPR7	ITUTPR8
	Own Code	*	*	*
	Control Routine	ITUTP7	ITUTP7	ITUTP7
Overlay 1	Scan-Validate	ITUTP8	ITUTP8	ITUTP8
Overlay 2	F-S Generation	ITUTP9	ITUTP9	ITUTP9
Overlay 3	Specific Val	ITUTPC4	ITUTPC5	ITUTPC7
	Buffer Comp & Logging	ITUTPE	ITUTPE	ITUTPE
Overlay 4	Input	ITUTP0	ITUTP2	ITUTP4
	Output	ITUTP5	ITUTP5	ITUTP5
Region 2	I/O Buffers	ITUTPB	ITUTPB	ITUTPB
	F-S Storage	ITUTPF	ITUTPF	ITUTPF

*Own-coding modules are named by the user. Refer to Page A-8 for own-coding options.

APPENDIX A

MODIFICATION OF PERIPHERAL ROUTINES

PERIPHERAL ROUTINE STRUCTURE

General Description

◆ The peripheral routines have been developed as a series of object modules. Each module performs a unique function although some modules are common to more than one routine.

Each program basically requires two modules: (1) a control block module that defines the actual functions to be performed by the program; and (2) a specific validation module to validate the parameters for each routine. The programmer has three options when using these routines:

1. To use the routine without modification;
2. To modify the routine by supplying parameters at run time; or
3. To tailor his own routine by binding assembled own-coding modules together with the supplied object modules.

The object modules forming the peripheral routines are listed below in the order they are to be bound by the Linkage Editor. The call name for each module used to construct a routine is also shown:

Module		Peripheral Routine			
Location	Type	CDTP	CDPR	TPTP	TPPR
Root segment	Control Block	ITUTPAC	ITUTPA	ITUTPAD	ITUTPAB
	Own Code	*	*	*	*
	Control Routine	ITUTP7	ITUTP7	ITUTP7	ITUTP7
Overlay 1	Scan-Validate	ITUTP8	ITUTP8	ITUTP8	ITUTP8
Overlay 2	Field Select Generation	ITUTP9	ITUTP9	ITUTP9	ITUTP9
Overlay 3	Specific Validation	ITUTPC2	ITUTPC	ITUTPC3	ITUTPC1
	Buffer Computation and Logging	ITUTPE	ITUTPE	ITUTPE	ITUTPE
Overlay 4	Input	ITUTP0	ITUTP0	ITUTP2	ITUTP2
	Output	ITUTP3	ITUTP6	ITUTP3	ITUTP6
Region 2	I/O Buffers	ITUTPB	ITUTPB	ITUTPB	ITUTPB
	Field Select Storage	ITUTPF	ITUTPF	ITUTPF	ITUTPF

*Own-coding modules are named by the user. Refer to page A-8 for own-coding options.

Example

◆ To bind object modules for the Card to Tape (CDTP) routine with an own-coding module for user processing after read:

The parameter cards required for the Linkage Editor routine are listed below. The own-coding module (ITUOC9) is on punched cards.

```
//ΔEXECΔLNKEDT (Monitor Control card)

ΔPROGΔCDTP

ΔNCAL

ΔXREF

ΔLET

ΔINCLUDEΔSYSLIB(ITUTPAC)

Δ INCLUDE ΔSYSUT1 (OWNC)

ΔINCLUDEΔSYSLIB(ITUTP7)

ΔOVERLAYΔALPHA,LOAD1

ΔINCLUDEΔSYSLIB(ITUTP8)

ΔOVERLAYΔALPHA,LOAD2

ΔINCLUDEΔSYSLIB(ITUTP9)

ΔOVERLAYΔALPHA,LOAD3

ΔINCLUDEΔSYSLIB(ITUTPC2,ITUTPE)

ΔOVERLAYΔALPHA,LOAD4

ΔINCLUDEΔSYSLIB(ITUTP0,ITUTP3)

ΔOVERLAYΔBETA,REGION,LOAD5

ΔINCLUDEΔSYSLIB(ITUTPB,ITUTPF)

ΔENTRYΔITUTP701
```

SUPPLIED MODULES

Root Segment

◆ *Control Block Modules*

Each peripheral routine has a unique control block containing all the processing options required by the routine.

Module Name	Routine
ITUTPAC	CDTP Control Block
ITUTPA	CDPR Control Block
ITUTPAD	TPTP Control Block
ITUTPAB	TPPR Control Block
ITUTPR1	CDRA Control Block
ITUTPR4	RAPR Control Block
ITUTPR5	RARA Control Block
ITUTPR3	RATP Control Block
ITUTPR2	TPRA Control Block
ITUTPR6	CDRAM Control Block
ITUTPR7	TPRAM Control Block
ITUTPR8	RARAM Control Block

Specific data in a control block may be referenced by the following tags:

Tag	Data
ITUTPACB	Control Block
ITUTPA1	Print Header Storage
ITUTPA3	Print/Punch Translate Table
ITUTP2F	Input Tape DTF (PRIPT1)
ITUTP3F	Output Tape DTF (PROPT1)
ITUTP4F	Input Random Access DTF
ITUTP5F	Output Random Access DTF

Control Block data may be modified by supplying Utility Modifier, Print, Punch or Page Heading parameters at run time.

This page deleted by revision: November, 1967

Root Segment *(Cont'd)*

Control Routine Module (ITUTP7)

This module consists of storage work areas and coding common to all the peripheral routines. It controls the initial entry into the routines; requests a parameter device assignment; reads supplied parameters; controls parameter processing; establishes the right-hand address of the field select storage area; and deallocates the parameter device.

After a peripheral program has been loaded, it branches to ITUTP701 to begin processing.

Overlay 1

◆ *Scan-Validate Module (ITUTP8)*

This module validates and stores the Utility Modifier parameters in the control block. Parameters for control block fields preset to binary ones will not be accepted or stored.

Overlay 2

◆ *Field Select Generation Module (ITUTP9)*

This module consists of coding required to generate and store the instructions for the field select operation specified. Generated instructions are stored beginning at the right-hand address of Region 2.

Overlay 3

◆ *Specific Validation Modules*

Each peripheral routine's control block is validated by a corresponding validation module.

Module Name	Routine
ITUTPC2	CDTP
ITUTPC	CDPR
ITUTPC3	TPTP
ITUTPC1	TPPR
ITUTPC4	CDRA/CDRAM
ITUTPC8	RAPR
ITUTPC7	RARA/RARAM
ITUTPC6	RATP
ITUTPC5	TPRA/TPRAM

Record requirements for the routines vary according to the peripheral device, as follows:

1. Card records must be fixed-length and unblocked. The maximum record and block size is 160 characters.

2. Magnetic tape records may be fixed-length with the block size equal to or a multiple of the record length. Or tape records may be variable or undefined with a record length of 0. The minimum record or block size is 12 characters.

Overlay 3
(Cont'd)

Each module also validates other control block parameters as shown in the following table.

Table A-1. Parameter Values (Non-Random Access Routines)

Parameter Identification	Parameter Values Allowed			
	CDTP	CDPR	TPTP	TPPR
Tx	C,R,F,RF	B,BF,C,F, DL,LF	C,R,F,RF	B,BF,C,F,D,L, LF,MB,JC,JL
Ix	1	1	R,U,N,M	R,U,N,M
Ox	R,U,N	C,1,X	R,U,N	C,1,X

Table A-2. Parameter Values (Random Access Routines)

Parameter Identification	CDRA/ CDRAM	RAPR	RARA/ RARAM	RATP	TPRA/ TPRAM
Tx	C,R,F,RF	B,C,D, F, L,BF, LF	C,R,F,RF	C,R,F,RF	C,R,F,RF
Ix	1	*	*	*	R,N,U,M
Ox	Y,N	1,X,C	Y,N	R,N,U	Y,N

*Not applicable.

Buffer Computation and Logging Module (ITUTPE)

This module computes the memory requirements for I/O buffers and controls all logging messages for each routine.

Double buffers are always allocated for punched card input or output. The printer is also double buffered and 160 bytes are allocated to each buffer. Depending on available memory, single or double buffers are computed for tape devices. See Appendix D for additional information on memory requirements and use of available memory.

Logging messages are preceded by a five-character code, xx51Δ.

xx = 22 for CDTP
24 for CDPR
20 for TPTP
26 for TPPR
38 for CDRA/CDRAM
39 for TPRA/TPRAM
41 for RAPR
42 for RATP
43 for RARA/RARAM

Overlay 4

◆ *Input Modules*

1. Card Read/Paper Tape Read Module (ITUTP0)

   This module reads punched card input into alternate 80-byte buffers.

2. Tape Read Module (ITUTP2)

   This module reads standard tape records from magnetic tape into one or two input buffers. It does not unblock variable-length, blocked records. FCP OPEN and CLOSE logic is provided.

   Note:

   When unlabeled tape input is specified:

   1. The tape is positioned before the first data record.
   2. Single volume input is assumed.
   3. The routine terminates at end of reel condition.

3. Random Access Read Module (ITUTP4)

   This module reads standard records from a random access volume using FCP DTFSR.

   Standard labels only are processed. FCP OPEN and CLOSE logic is provided.

Output Modules (ITUTP6, ITUTP3)

1. Printer/Punch Write Module (ITUTP6)

   This module provides printer and/or punched card output. Punched card output is one 80-column card in EBCDIC per input record and an END card with /* in the first two columns. Printer output is in the mode, format, etc. as specified by the parameters in the control block.

2. Tape Write Module (ITUTP3).

   This module writes standard records to magnetic tape from one or two output buffers in the format specified by the control block parameters. It does not block variable-length records. FCP OPEN and CLOSE logic is provided.

   Note:

   Data records on an unlabeled output tape may or may not be preceded by a tape mark and followed by a double tape mark.

3. Random Access Write Module (ITUTP5)

   This module writes standard records to a random access volume using FCP DTFSR.

   Standard labels only may be written. FCP OPEN and CLOSE logic is provided.

Region 2

◆ *I/O Buffer Module (ITUTPB)*

This module defines the buffer areas that are required as a result of buffer computation.

Field Select Storage Module (ITUTPF)

This module consists of the generated instructions for the field-select operations specified in the control block module.

OWN CODING

Own Coding - General

◆ Eleven own-coding options are available to the user. Each own-coding option corresponds to an external symbol defined as an EXTRN in the control routine module. When the control routine module is assembled, a zero address constant is generated for each EXTRN. The actual address of the corresponding own coding ENTRY, if present, is supplied at Linkage Editor time.

At run time each EXTRN is tested for a nonzero address and when satisfied, control is given to the own-coding module at that address. The own-coding options and corresponding EXTRN operands are as follows:

EXTRN	Own-Coding Option
ITUOC1	Check Standard Input UHL's.
ITUOC2	Check Standard Input UTL's.
ITUOC3	Create Standard Output UHL's.
ITUOC4	Create Standard Output UTL's.
ITUOC5	Process Nonstandard Input Header Labels.
ITUOC6	Process Nonstandard Input Trailer Labels.
ITUOC7	Create Nonstandard Output Header Labels.
ITUOC8	Create Nonstandard Output Trailer Labels.
ITUOC9	User Processing after Read.
ITUOC10	User Processing prior to Write.
ITUOC11	EOJ.

Note:

ITUOC5 through ITUOC8 are available for magnetic tape only.

Register Usage

Registers 5 and 6 have been allocated as base registers for the own-coding modules.

Register 14 contains the return address to the utility routine upon entering own-coding.

Own Coding - General *(Cont'd)*

Register 1 contains either the left-hand address of the label area upon entering ITUOC1 through ITUOC4 or the operand of the LBRET macro used to exit from own-coding.

Register 0 contains either P (punch or L (print) constants upon entering ITUOC10 in the CDPR, RAPR, or TPPR routine or EV or RF constants when exiting from ITUOC6 and ITUOC8.

Register 12 contains the left-hand address of the logical record to be processed upon entering ITUOC9.

Register 13 contains the left-hand address of the logical record to be processed upon entering ITUOC10.

Note:

Any register other than 0, 1, 5, or 6 that is used by own-coding must be stored upon entering own-coding and restored prior to exiting from own-coding.

Linkage

Register 14 is always used by the utility program to enter own-coding modules ITUOC1 through ITUOC11.

The LBRET macro must be used to exit from own-coding modules ITUOC1 through ITUOC4 after each label has been processed. All labels are read and written by the utility program. If eight labels are to be processed, only LBRETΔ2 is required since label processing is automatically terminated after the eighth label. If fewer than eight labels are to be processed, both LBRETΔ2 and LBRETΔ1 are required. LBRETΔ2 indicates that another label is to be processed. LBRETΔ1 forces end of label processing.

Either Register 14 or LBRETΔ1 must be used to exit from own-coding modules ITUOC5 through ITUOC11.

```
USER     START
*OWN CODING FOR PROCESS AFTER READ
         ENTRY    ITUOC9
         USING    *,5
ITUOC9   BALR     5,0
         B        START
         .
         .
         .
         .
START    (User Processing)
         .
         .
         .
         .
RETURN   BR       14
         END
```

Own-Coding - General *(Cont'd)*

Label Processing - Tape

When tape labels are specified in the control block, own-coding options ITUOC1 through ITUOC8 are available to the programmer. In the case of standard labels, the programmer may chose to do all label processing or only UHL and UTL processing. However, in the case of nonstandard labels, the user must do all label processing.

Standard labels on tapes defined as unlabeled are recognized but not transcribed.

Label Processing - Random Access

Files on random access devices to be processed by the peripherals must be written with standard labels. The Random Access Volume Initializer and Storage Allocator routines write these labels.

If standard user labels are desired, own-coding options ITUOC1 through ITUOC4 are available to the programmer. The first track of the first extent of a file is reserved for user labels regardless of whether labels are written or not.

Nonstandard labels are not supported on random access devices.

Additional Processing of Data Records

Any processing of data records that does not alter the record size may be handled by the user through own-coding options ITUOC9 and ITUOC10.

EOJ

Special end of job processing may be provided by the user through own-coding option ITUOC11. In the case of a successor program call, however, control is not returned to the utility program.

Own Coding - Detailed

◆ *Tape Labels*

1. Input Header Labels

   After positioning the input tape according to supplied parameters, the utility program tests ITUOC5 EXTRN for satisfaction.

   If ITUOC5 is satisfied, the utility program enters own-coding. The programmer must read and check all labels and position the tape at the first data record before returning to the utility program by way of Register 14 or LBRETΔ1.

   If not satisfied, standard label processing is assumed. Labels VOL1 and HDR1 are processed while VOL2-8 and HDR2-8, if present, are bypassed. When a non-HDR record is finally read, ITUOC1 EXTRN is tested for satisfaction.

Own-Coding - Detailed *(Cont'd)*

If ITUOC1 is satisfied, the address of the UHL record is stored in Register 1 and own-coding is entered. After checking each UHL record, the programmer must return to the utility program by way of the LBRET macro.

If not satisfied, the OPEN logic is completed when a tape mark is recognized. UHL records, if present, are bypassed.

2. Input Trailer Labels

When an end of volume condition is sensed, the utility program tests ITUOC6 EXTRN.

If ITUOC6 is satisfied, the utility program enters own-coding. The programmer must read and check all labels and store an EF (End of File) or EV (End of Volume) in the low-order bytes of Register 1 before returning to the utility routine by way of register 14 or LBRETΔ1.

Note:

The address of a four-byte blockcount (packed decimal) may be obtained by including the following EXTRN and address constant.

```
EXTRN      ITUTP7S4
DC         A(ITUTP7S4 + 4)
```

If not satisfied, standard label processing is assumed. EOV1 or EOF1 is processed and EOV2-8 or EOF2-8, if present, are bypassed. When a non-EOV or EOF label is finally read, ITUOC2 EXTRN is tested for satisfaction.

If ITUOC2 is satisfied, the utility program stores the address of the UTL record in Register 1 and enters own-coding. After checking each UTL record, the user must return to the utility program by way of the LBRET macro.

If not satisfied, the CLOSE logic is completed when a tape mark is recognized. UTL records, if present, are bypassed.

3. Output Header Labels

After positioning the output tape according to supplied parameters, the utility program tests ITUOC7 EXTRN for satisfaction.

If ITUOC7 is satisfied, the utility program enters own-coding. The programmer must create and write all header labels before returning control to the tape write module by branching to the address in Register 14 or LBRETΔ1.

If not satisfied, standard label processing is assumed. VOL1 and HDR1 are processed and VOL2-8 and HDR2-8 are bypassed, if present. ITUOC3 EXTRN is then tested for satisfaction.

If ITUOC3 is satisfied, the left-hand address of the label area is stored in Register 1 and own-coding is entered. After creating each 80-byte UHL record, the programmer must return to the utility program by way of the LBRET macro. The programmer must create at least one UHL but may not create more than eight.

If not satisfied, a tape mark is written.

Own-Coding - Detailed *(Cont'd)*

4. Output Trailer Labels

After sensing an end of volume condition and writing a tape mark, the utility program tests ITUOC8 EXTRN for satisfaction.

If ITUOC8 is satisfied, the utility program stores EF (End of File) or EV (End of Volume) in the low-order bytes of Register 0 and enters own-coding. The programmer must create and write all labels before returning control to the tape write module by way of Register 14 or LBRETΔ1.

Note:

The address of a four-byte block count (packed decimal) may be obtained by including the following EXTRN and address constant:

```
EXTRN       ITUTP7S4
DC          A(ITUTP7S4 + 12)
```

If not satisfied, standard trailer labels are assumed and the appropriate label (EOF or EOV) is written. ITUOC4 is then tested for satisfaction.

If ITUOC4 is satisfied, Register 1 is loaded with the left-hand address of the label area and own-code is entered. After creating each UTL record the programmer must return to the utility program by way of the LBRET macro. The programmer must create at least one UTL, but may not create more than eight.

If not satisfied, a tape mark is written.

Random Access Labels

1. Input Header Labels

After building the extent matrix, the utility program tests ITUOC1 EXTRN for satisfaction.

If ITUOC1 is satisfied, the address of the UHL record is stored in Register 1 and own-coding is entered. After checking each UHL record, the programmer must return to the utility program by way of the LBRET macro.

If ITUOC1 is not satisfied, UHL records, if present, are bypassed.

2. Input Trailer Labels

When an EOF record is sensed, the utility program tests ITUOC2 EXTRN.

If ITUOC2 is satisfied, the utility program stores the address of the UTL record in Register 1 and enters own-coding. After checking each UTL record, the user must return to the utility program by way of the LBRET macro.

If ITUOC2 is not satisfied, UTL records, if present, are bypassed.

Own Coding - Detailed
(Cont'd)

3. Output Header Labels

After building the extent matrix, the utility program tests ITUOC3 EXTRN for satisfaction.

If ITUOC3 is satisfied, the left-hand address of the label area is stored in Register 1 and own-coding is entered. After creating each 80-byte UHL record, the programmer must return to the utility program by way of the LBRET macro. The programmer must create at least one UHL but may not create more than eight.

If ITUOC3 is not satisfied, a UTL EOF record is written on the first track of the extent and processing continues.

4. Output Trailer Labels

At the end of processing, the ITUOC4 EXTRN is tested by the utility program. If satisfied, Register 1 is loaded with the left-hand address of the label area and own-code is entered. After creating each UTL record, the programmer must return to the utility program by way of the LBRET macro. The programmer must create at least one UTL, but may not create more than eight.

If ITUOC4 is not satisfied, a UTL EOF record is written following the UHL EOF and processing continues.

Process After Read

After a logical record or block is read into the input buffer, ITUOC9 EXTRN is tested for satisfaction.

If ITUOC9 is satisfied, the utility program stores the left-hand address of the logical record to be processed in Register 12 and enters own-code. The programmer may not alter the record's size. The programmer may move the record to another area if the left-hand address of the area is stored in Register 12. After processing the record, the programmer must return to the utility program by way of Register 14 or LBRETΔ1.

If not satisfied, the utility program will process the data record according to the parameters specified.

Process prior to Write

After the utility program has processed a logical record and moved it to the output buffer, it tests ITUOC10 EXTRN for satisfaction.

Own Coding - Detailed *(Cont'd)*

If ITUOC10 is satisfied, the utility program stores the left-hand address of the logical record processed in Register 13 and enters own-coding. The programmer may not change the record's size or move it to another area for processing. After processing the record, the programmer must return to the utility program by way of Register 14 or LBRETΔ1.

> Note:
>
> If ITUOC10 is satisfied and the function specifies both punch and printer output, own-coding is entered twice. The low-order byte of Register 0 contains a P when own-coding is entered prior to punching or an L when own-coding is entered prior to printing.

If not satisfied, the data record is written out by the utility program, in the format specified by the program parameters.

End of Job

After all processing has been completed and CLOSE logic initiated, ITUOC11 EXTRN is checked for satisfaction.

If ITUOC11 is satisfied, the utility routine enters own-coding. The programmer must handle any additional processing and return to the utility program by way of Register 14 or LBRETΔ1. However, if the programmer initiates a successor program call, control must not be returned to the utility program.

If not satisfied, normal termination of the utility routine is assumed.

Special Considerations

◆ When own-coding options are exercised, the following considerations are the user's responsibility:

1. Creation of each own-coding routine to be included in the utility program. The following must be included in the own coding routine: An ENTRY statement in each own-coding routine that corresponds to the EXTRN of the option exercised and the return linkage using the LBRET macro or Register 14.

2. Assembly of each own-coding routine to be linked.

   If more than one own-coding routine is to be included in the same utility program each routine may be assembled as a separate object module or all the routines may be assembled as a single object module.

3. Binding of all object modules required by the utility program using the Linkage Editor.

   Own-coding object modules must be included before the Control routine. Multiple own-coding modules of the same type may not be included in the same utility program.

APPENDIX B

TOS LIBRARY FORMATS

LOAD LIBRARIES

General

◆ In the Tape Operating System, programs are loaded and executed from load library tapes, of which there are three types:

1. System Load Library (SLL): contains operating system components and object programs.
2. Program Load Library (PLL): contains object programs only.
3. Executive Load Library (ELL): contains operating system components only.

All programs, Executive and object, are preceded and followed by program descriptor blocks which are used by the Executive for program searching.

Each program is composed of one or more segments, called loads. Each load is preceded by a load descriptor block which is used by the Executive for overlay searching. The portion of the load containing instructions, constants, and their relative locations within the load are called text blocks. In addition, modifier blocks may be present when address constants appear within the load.

Programs are separated from each other by a tape mark; the final program is followed by two tape marks.

System Load Library (SLL)

◆ This tape contains the following system components:

1. Bootstrap
2. Resident Executive Loader
3. Resident Executive
4. Executive overlays
5. Interspersed Executive overlays
6. Monitor
7. Object programs.

The overall organization of an SLL containing two object programs is illustrated in figure B-1.

System Load Library (SLL)
(Cont'd)

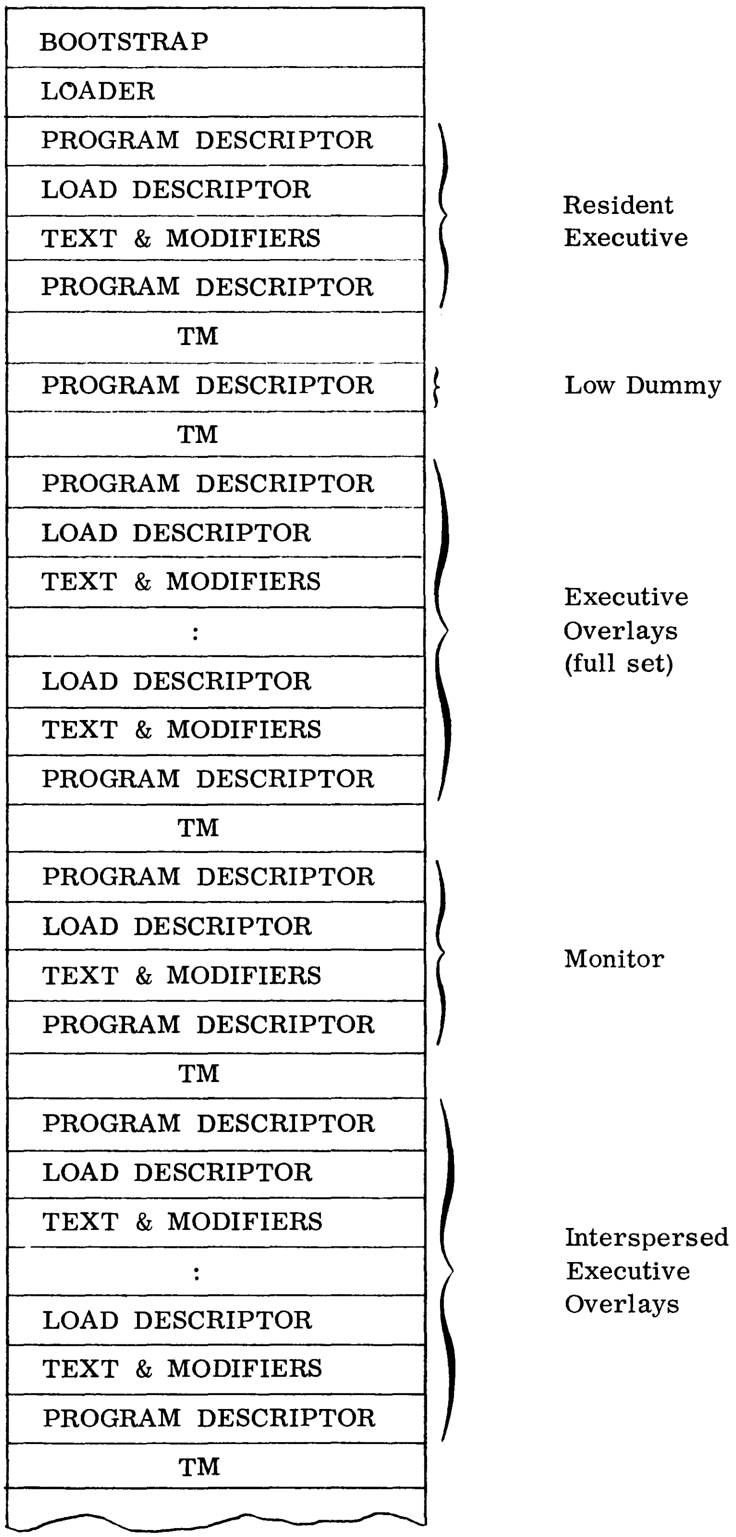


Figure B-1. System Load Library

System Load Library (SLL)
(Cont'd)

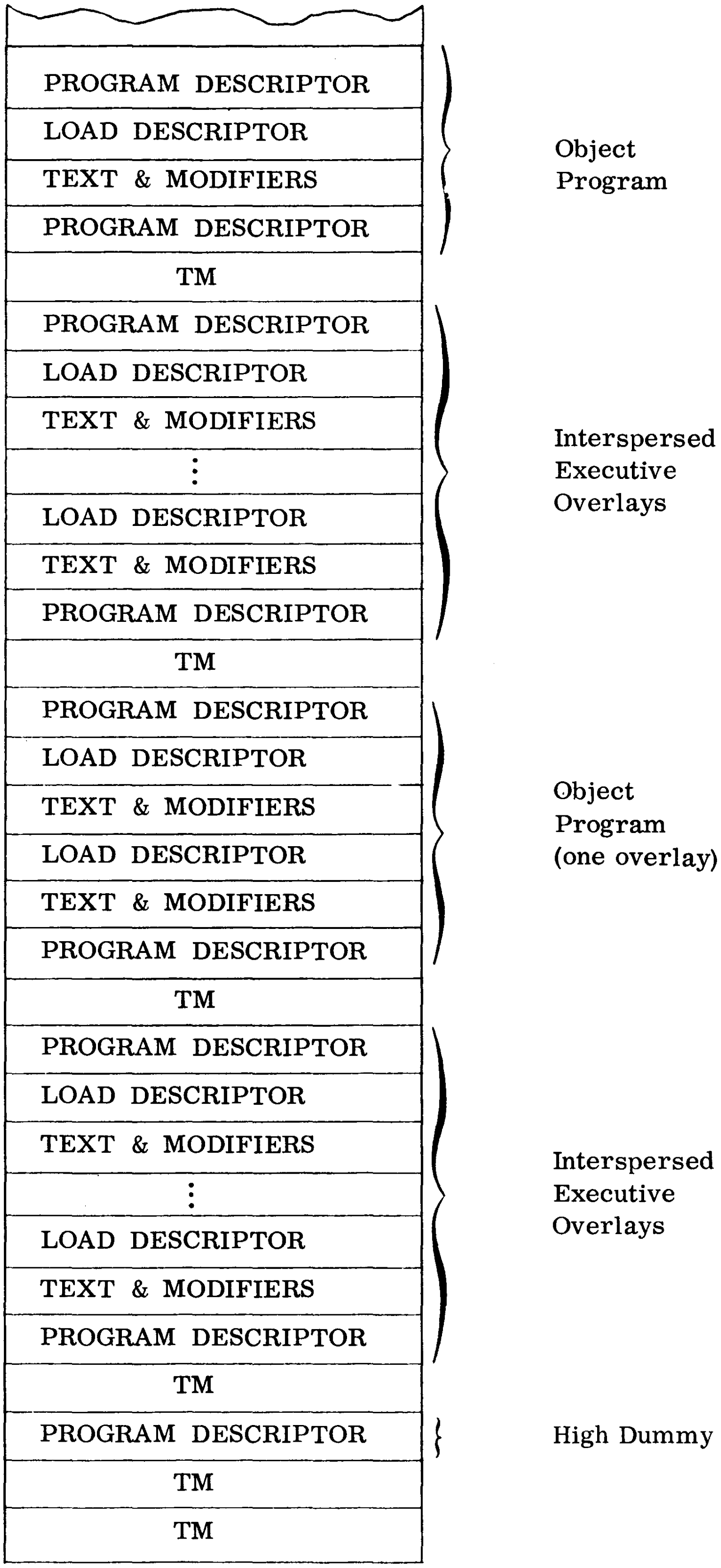


Figure B-1. System Load Library (Cont'd)

Program Load Library (PLL)

◆ This tape contains object programs only, and thus requires another library tape (SLL or ELL) to be mounted to execute any program contained thereon. The overall organization of a PLL containing three object programs is illustrated in figure B-2.

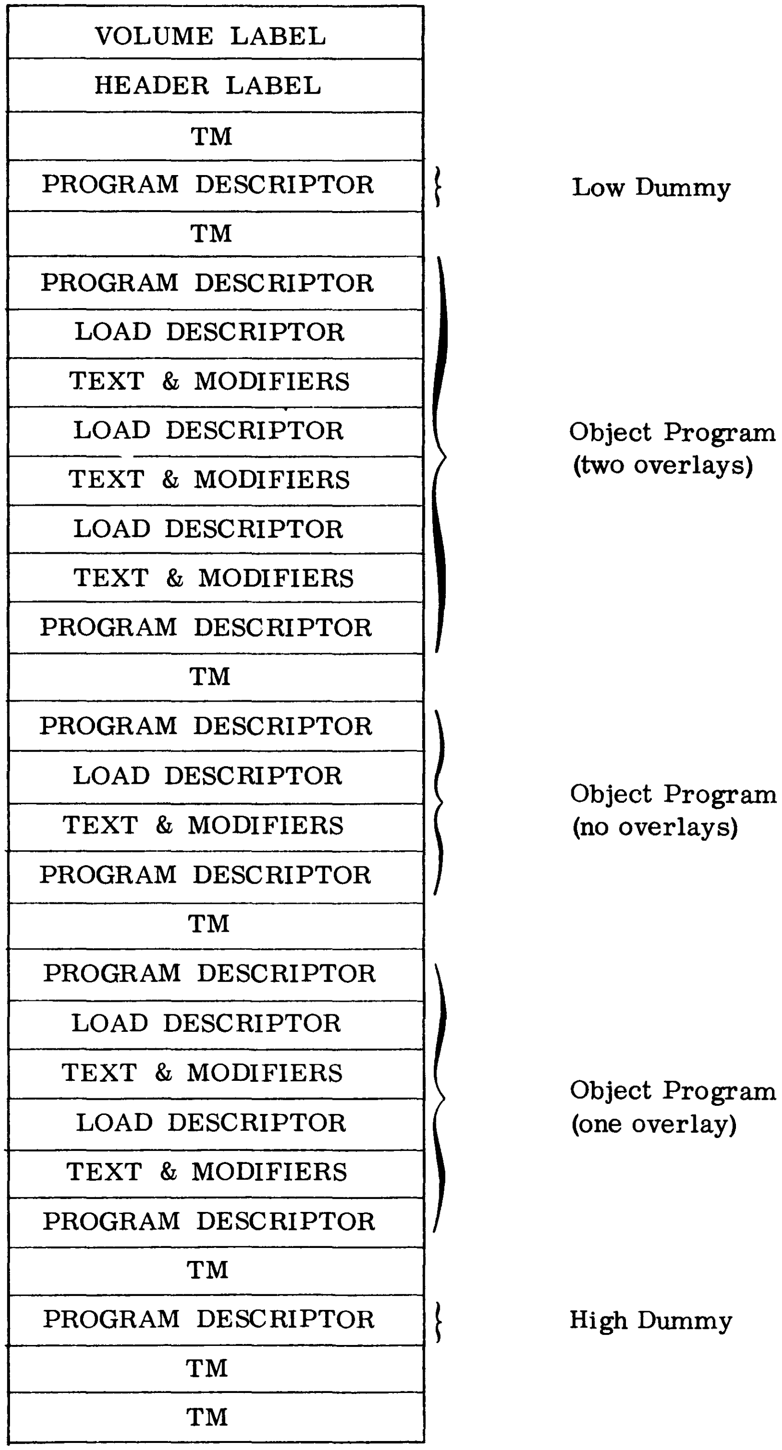


Figure B-2. Program Load Library

Executive Load Library (ELL)

◆ This tape contains the following operating system components only:

1. Bootstrap
2. Resident Executive Loader
3. Resident Executive
4. Executive Overlays
5. Monitor

The organization of this tape is the same as that illustrated in figure B-1, except that object programs are not included.

Load Library Record Formats

◆ 1. PROGRAM DESCRIPTOR BLOCK

Format:

Bytes	Contents
0	Block code: $D1_{(16)}$ = Dummy $E1_{(16)}$ = Executive overlay $01_{(16)}$ = Object
1-7	Reserved
8-11	Start address, in binary, of entry point to root load.
12-19	Program name.
20-23	Minimum memory requirement, in binary, for program.
24-27	Maximum memory requirement, in binary, for program.
28-35	Root load name.
36-41	Date created by Linkage Editor.
42-44	Version number, 000-999.
45-49	Reserved.
50	Block code, same as byte 0.

Note:

The Low Dummy Program Descriptor block contains $00_{(16)}$ in bytes 1 through 49. The High Dummy contains $FF_{(16)}$ in bytes 1 through 49.

Load Library Record Formats *(Cont'd)*

2. LOAD DESCRIPTOR BLOCK

Format:

Byte	Contents
0	Block code, always $02_{(16)}$.
1	Block subcode: $00_{(16)}$ = this load contains text. $01_{(16)}$ = this load does not contain text.
2-3	Reserved.
4-7	Program-relative loading address, in binary, of first text block in load.
8-11	Program-relative address, in binary, of load node point.
12-19	Load name.
20	Block code, same as byte 0.

3. TEXT BLOCK

Format:

Byte	Contents
0	Block code, always $04_{(16)}$.
1	Block sub-code: $00_{(16)}$ = a modifier block follows. $01_{(16)}$ = a text block follows. $02_{(16)}$ = this is last block in load.
2-3	Number of bytes of text, in binary, in this text block.
4-7	Program-relative loading address, in binary, of next text block (if byte 1 contains $01_{(16)}$) in load.
8-19	Reserved.
20-1043	Text. This field has a variable length; its size is determined by bytes 2-3. It may contain from 1 to 1,024 bytes of text.

Load Library Record Formats *(Cont'd)*

4. MODIFIER BLOCK

Format:

Byte	Contents
0	Block code, always $05_{(16)}$.
1	Block subcode: $00_{(16)}$ = a modifier block follows. $01_{(16)}$ = a text block follows. $02_{(16)}$ = this is last block in load.
2-3	Number of modifiers, in binary, in this block.
4-7	Program-relative loading address, in binary, of next text block (if byte 1 contains $01_{(16)}$) in load.
8-11	Reserved.
12-19 or 12-123	Modifiers. This field is variable in size and contains from 1 to 27 modifiers. Its size is determined by bytes 2-3.

Note:

The first modifier (bytes 12-19) is always eight bytes and has the following format:

Format of a Modifier in Modifier Block

Bytes	Bit	Meaning
0-3		Program-relative float factor, in binary, by which the address constant is modified.
4	0-3	Zeros.
	4-5	Length of address constant: 00 = one byte 01 = two bytes 10 = three bytes 11 = four bytes
	6	Action flag: 0 = add to address constant. 1 = subtract from address constant.
	7	Length of next modifier: 0 = next modifier is eight bytes (a new float factor is to be applied). 1 = next modifier is four bytes (same float factor is to be applied).
5-7		Program-relative location, in binary, of address constant.

OBJECT MODULE LIBRARY

◆ The Object Module Library (OML) is a component of the TOS Call Library. The Call Library Tape contains the data required by the Assembler, COBOL, and Linkage Editor to produce executable programs. The tape itself is in nonexecutable form. The other components which may be present on the Call Library Tape are:

Assembly Macro Library

COBOL Library

System Executive

The Object Module Library consists of a series of object modules produced by a language translator. These modules are sequenced on the OML alphabetically by module name.

The OML is composed of three sections:

1. Directory of module names and relative position on the library.

   The OML directory facilitates searching for specific object modules stored in the library. This directory contains the names of the modules and their relative positions in the library. Object module names are arranged alphabetically.

2. Indices of entries, extrns, and common areas in each module.

   The index blocks contain information pertaining to ENTRY, EXTRN, and COMMON items. Index blocks for all object modules appear immediately following the directory block. They are arranged in the same order as the modules to which they pertain. Within each index block, all entries will appear first (in alphabetical order), followed by all extrns and V-type constants, followed by all defined common statements. If an object module contains an INCLUDE statement, the name of the module to be included is stored in the initial index block.

3. Object modules

   There are four types of blocks for an object module. These are as follows:

   a. Descriptor Block

      One descriptor block always precedes each module in the library. It contains information concerning the types of blocks in the module and the load address for the first text block.

   b. Extrn Blocks

      These blocks contain the names and ESID numbers of all the external references in the module. There can be up to 26 EXTRNs in a block.

OBJECT MODULE LIBRARY (Cont'd)

c. Text Blocks

Each text block contains up to 1,024 bytes of text.

d. Text Modifier Blocks

One or more modifier blocks follow each text block if the preceding text block contains address constants that must be modified when the module is relocated.

Figure B-3 shows an example of a two-module OML and the detailed formats for each type of block mentioned above.

TAPE MARK	
IDENTIFICATION BLOCK	
00	DIRECTORY BLOCK
01	MODULE A INDEX
01	MODULE A INDEX
01	MODULE A INDEX
01	MODULE B INDEX
01	MODULE B INDEX
02	MODULE A DESCRIPTOR
03	MODULE A EXTRNS
04	MODULE A TEXT
05	MODULE A MODIFIERS
05	MODULE A MODIFIERS
04	MODULE A TEXT
05	MODULE A MODIFIERS
04	MODULE A TEXT
02	MODULE B DESCRIPTOR
03	MODULE B EXTRNS
04	MODULE B TEXT
05	MODULE B MODIFIERS
TAPE MARK	
TAPE MARK	

Figure B-3. Example of a Two-Module Object Module Library

Object Module Directory Block (00)

Item	Bytes	LHE	RHE	Name	Format	Contents
1	1	0	0	Block code.	Hexadecimal	Contains 00.
2	1	1	1	Block subcode.	Hexadecimal	00 = first and last. 01 = first. 02 = intermediate. 03 = last.
3	7	2	8	(Not used.)	Variable	Reserved for future use.
4	8	9	16	Module name.	EBCDIC	Name of first module.
5	2	17	18	Position of module.	Hexadecimal	= 0001 for first module.
6	8	19	26	Module name.	EBCDIC	
7	2	27	28	Position of module.	Hexadecimal	
Repeat items 6 and 7 up to 27 additional times.						
		29	298			
n	1	299	299	Block code.	Hexadecimal	Contains 00.

Fixed-length block of 300 bytes.

Index Block (01)

Item	Bytes	LHE	RHE	Name	Format	Contents
1	1	0	0	Block code.	Hexadecimal	$(01)_{16}$.
2	1	1	1	Block code.	Hexadecimal	$(00)_{16}$ = first and last. $(01)_{16}$ = first. $(02)_{16}$ = intermediate. $(03)_{16}$ = last.
3	7	2	8	Reserved.	Hexadecimal	Binary zero.
4	2	9	10	Sequence.	Binary	Relative number of modules.
5	8	11	18	Module name.	EBCDIC	
6	1	19	19	Revision number.	Hexadecimal	
7	5	20	24	Revision date.	EBCDIC	
8	3	25	27	Module length.	Hexadecimal	
9	1	28	28	Number of entries.	Hexadecimal	
10	1	29	29	Number of EXTRNS.	Hexadecimal	
11	1	30	30	Number of common labels.	Hexadecimal	
12	8	31	38	DDNAME (for Include).	EBCDIC	Or binary zero (one per module permitted).
13	8	39	46	OMNAME (for Include).	EBCDIC	Or binary zero (one per module permitted).
14	8	47	54	EXTRN name.	EBCDIC	Entrance point for execution.
15	4	55	58	Starting address.	Hexadecimal	Entrance point addend to EXTRN name, or actual address.
16	12	59	70	Data item.	Variable	Entries, EXTRNS, Common.
Repeat item 16 up to 19 additional times.						
		71	298			
n	1	299	299	Block code.	Hexadecimal	$(01)_{16}$.

Fixed-length block of 300 bytes (binary, zero-filled).

Format of Data Items (12 bytes per item)

Type of Item	Bytes	Content
Entry	8	Name
	1	$(F1)_{16}$
	3	Relative position in module.
Extrn	8	Name
	1	$(F2)_{16}$
	3	$(000000)_{16}$
Common	8	Name
	1	$(F3)_{16}$
	3	Length
V-Type	8	Name
	1	$(F4)_{16}$
	3	$(000000)_{16}$

Object Module Descriptor Block (02)

Item	Bytes	LHE	RHE	Name	Format	Contents
1	1	0	0	Block code.	Hexadecimal	Contains $(02)_{16}$.
2	1	1	1	Type of blocks that follow.	Hexadecimal	(See Note.)
3	4	2	5	(Not used.)	Variable	Reserved for future use.
4	8	6	13	Module name.	EBCDIC	
5	1	14	14	Revision number.	Hexadecimal	
6	5	15	19	Revision date.	EBCDIC	
7	3	20	22	Starting address.	Binary	Load address for first text block.
8	1	23	23	Block code.	Hexadecimal	Contains $(02)_{16}$.

Fixed Block Size - 24 Bytes

Note:

$(00)_{16}$ = TEXT only.

$(01)_{16}$ = EXTRN and TEXT.

$(02)_{16}$ = Object Module Descriptor for next Object Module.

$(03)_{16}$ = EXTRN only.

EXTRN Block (03)

Item	Bytes	LHE	RHE	Name	Format	Contents
1	1	0	0	Block code.	Hexadecimal	Contains $(03)_{16}$.
2	1	1	1	Block subcode.	Hexadecimal	00 = intermediate EXTRN block following. 01 = last.
3	10	2	11	(Not used.)	Variable	Reserved for future use.
4	8	12	19	EXTRN.	EBCDIC	
5	1	20	20	Type code.	Hexadecimal	(See Index Block.)
6	2	21	22	ESID.	Hexadecimal	
Repeat items 4, 5, and 6 up to 25 additional times.						
		23	298			
n	1	299	299	Block code.	Hexadecimal	Contains $(03)_{16}$.

Fixed-length block of 300 bytes (binary, zero-filled, unused items).

Text Block (04)

Item	Bytes	LHE	RHE	Name	Format	Contents
1	1	0	0	Block code.	Hexadecimal	$(04)_{16}$.
2	1	1	1	Block subcode.	Hexadecimal	Following block descriptor: $(00)_{16}$ = Modifier follows. $(01)_{16}$ = text follows. $(02)_{16}$ = current block is last. (End of Module.)
3	2	2	3	Block byte count.	Binary	Number of bytes this block.
4	4	4	7	Load address next block.	Binary	Null if subcode $\neq (01)_{16}$.
5	12	8	19	Reserved.	Variable	Future requirements.
6	1024	20	1042	Text.	EBCDIC	Instructions, data.

Minimum block size = 21.
Maximum block size = 1044.

Modifier Block (05)

Item	Bytes	LHE	RHE	Name	Format	Contents
1	1	0	0	Block code.	Hexadecimal	$(05)_{16}$.
2	1	1	1	Block subcode.	Hexadecimal	Following block descriptor: $(00)_{16}$ = modifier follows. $(01)_{16}$ = text follows. $(02)_{16}$ = current block is last. (End of Module.)
3	2	2	3	Modifier count.	Binary	Number of modifiers this block.
4	4	4	7	Load address next block.	Binary	Null if subcode $\neq (01)_{16}$.
5	4	8	11	Reserved.	Variable	For future requirements.
6	10	12	21	Modifier	Variable	(See note below.)

Minimum block size = 21.
Maximum block size = 117.

Note:

Format of Modifier:

The first modifier is always 10 bytes. Succeeding modifiers must be only four bytes if the continuation bit of the previous modifier is one.

Bytes	Content
4	Module relative float factor.
2	ESID number.
1	Flag - format = 0000LLMC where: LL is length of constant to be modified. LL = 00 one byte. = 01 two bytes. = 10 three bytes. = 11 four bytes. M = 0 add value of float factor. = 1 subtract value of float factor. C = 0 last modifier in block or, next modifier has new float factor or ESID number. = 1 a modifier follows using previous float factor and ESID.
3	Module relative pointer to address constant.

CALL LIBRARY

◆ The Call Library tape is a composite library tape that contains the various libraries used in TOS. Library sections (when present) appear on this tape in the following order:

ASSEMBLY MACRO LIBRARY
COBOL LIBRARY
OBJECT MODULE LIBRARY
EXECUTIVE LIBRARY

Figure B-4 illustrates the composition of a complete Call Library tape.

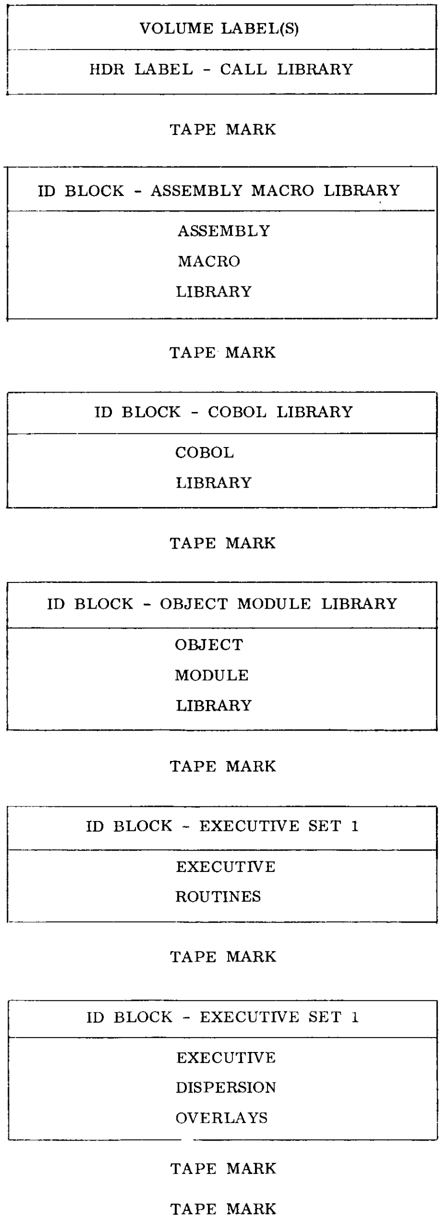
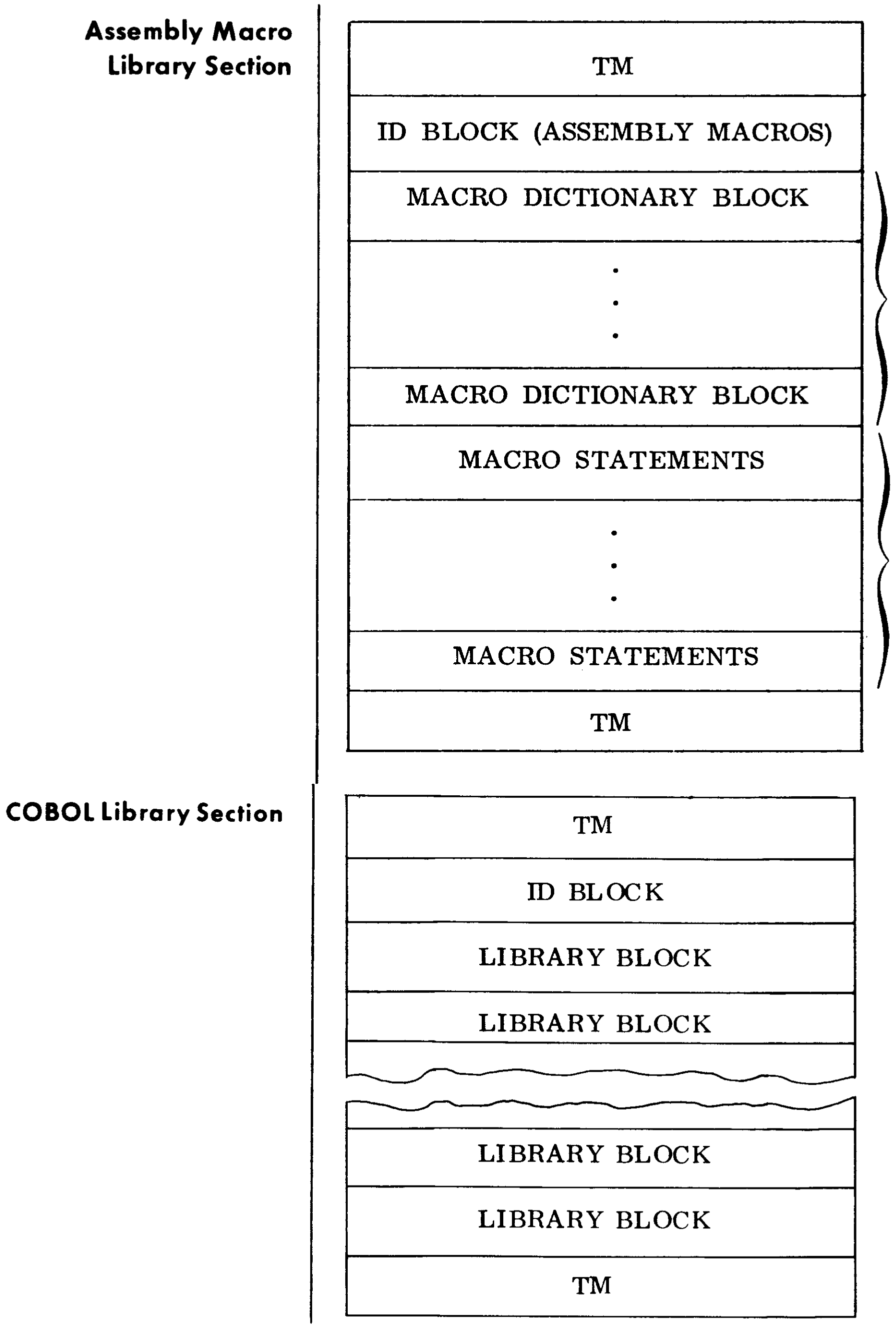


Figure B-4. TOS Call Library



Arranged in priority
number order (1-4) and
by alphanumeric sequence
within priority.

Arranged in priority
number order (1-4) and
by alphanumeric sequence
within priority.

Arranged in library section order (1-4) with entry names within each section listed in alphanumeric order.

COBOL Library Section
(Cont'd)

1. *Identifier Block*

Byte	Contents
0- 2	Block Identifier, always IDN.
3	Position of COBOL Library in relation to other libraries on tape. May be 1, 2, 3, or 4.
4- 20	File Identification, COBOL-SOURCE-LIBR.
21- 38	Not used.
39- 40	Version number (01-99).
41	Not used.
42- 46	Creation Date, in format YYDDD. YY = Year DDD = Julian Date
47- 80	Not used.

2. *Library Block*

Byte	Contents
0	Block Code: $02_{(16)}$ = First block of entry. $03_{(16)}$ = Not first block of entry.
1	Section Number: $01_{(16)}$ = Environment Division $02_{(16)}$ = Data Division $03_{(16)}$ = Procedure Division $04_{(16)}$ = Complete source programs
2	Block Subcode: $02_{(16)}$ = Not last block of entry. $03_{(16)}$ = Last block of entry.
3- 10	Entry Name, left-justified and space filled. (Repeated in all blocks of entry.)
11- 16	Creation date or last date modified, in format MMDDYY.
17-488	Source program statements in compressed format.

Note:

Library blocks are variable in length and may contain up to 489 bytes. Block length is always a multiple of three bytes, with binary zeros used to pad the block to a multiple of three if required.

COBOL Library Section *(Cont'd)*

3. Format of a Source Program Statement in Library Block

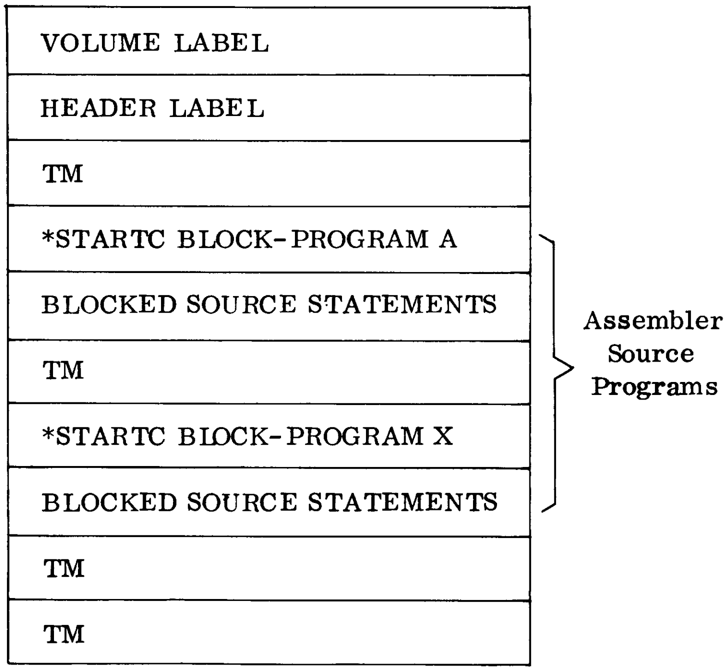
Byte	Contents
0	Length of compressed source program statement, including this byte.
1- 6	Sequence number.
7- 80	Variable-length COBOL statement with redundant spaces compressed.

Note:

Source cards are compressed by replacing each field of two or more spaces by a one-byte counter. Two counter bytes are required if the source card contains more than 63 contigous spaces.

SOURCE LIBRARY FORMATS

Source Library (SYSUT5)



A source library tape may be generated by the TOS Assembler or the TOS Source Library Update routine. Either tape may be supplied as input to the SLU routine.

Source Library (SYSUT5)
(Cont'd)

Standard Volume labels, Header labels, and tape marks are generated by the SLU routine or Assembly system.

The *STARTC block generated by the SLU contains the following data:

Position	Value
1-8	*STARTCΔ
9-16	Eight-character program name (unused positions are space-filled).
17-22	Spaces.
23-25	VER
26-28	Three-digit version number specified in the *STARTCΔ parameter.
29	Source statement format identifier. FF identifies compressed format. Any other value indicates fixed format.
30-35	Version date (MMDDYY).
36-80	Spaces.

The *STARTC block generated by the Assembler is the card image of the Assembler *STARTC parameter.

Source statements are always blocked with the blocking factor depending upon the record format. Fixed-length records are blocked by five;compressed records are variably blocked up to a maximum of 480 bytes per block. When specified, records are compressed by replacing each field of two or more spaces by a one-byte counter.

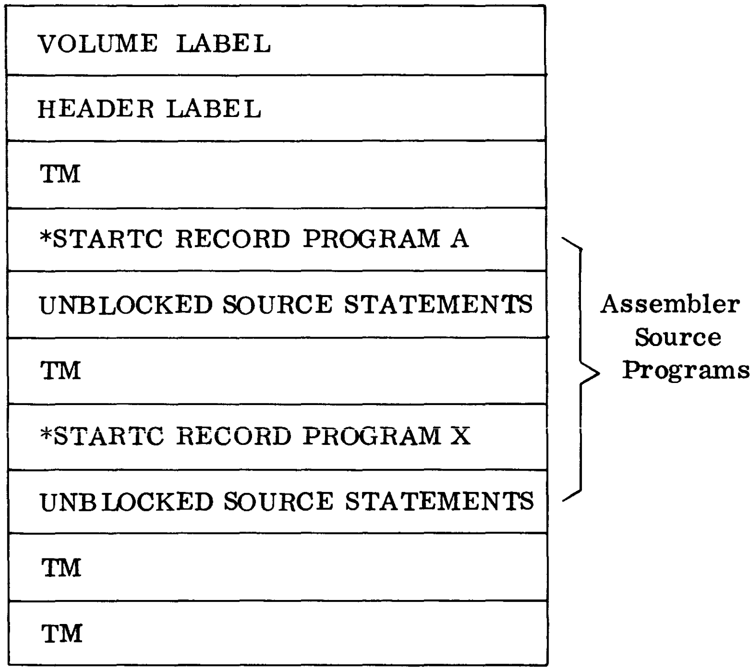
Note:

When a source statement contains a hexadecimal value less than 40, it is not compressed and is prefixed by hexadecimal 51.

The *STARTC block is always one, unblocked 80-character record.

A tape mark separates each program; a double tape mark follows the last program.

Optional Output Tape (OPTOUT)



This tape may be generated as an option of the SLU routine. Any program may be selected for this tape. Merge/extract logic is not required.

A standard Volume label, Header label, and tape marks are automatically generated. The header label contains the following file identifier:

SOURCEΔLIBΔOPTOUT.

The *STARTC record is the same as the output library tape.

Source statements are unblocked, 80-character records. A tape mark separates each program; a double tape mark follows the last program.

APPENDIX C

RANDOM ACCESS LABEL FORMATS

STANDARD VOLUME LABEL

◆ The standard volume label for a random access volume is written by the RAINIT routine with a standard count field, a Key field which contains the constant VOL1, and an 80-character data field formatted as follows:

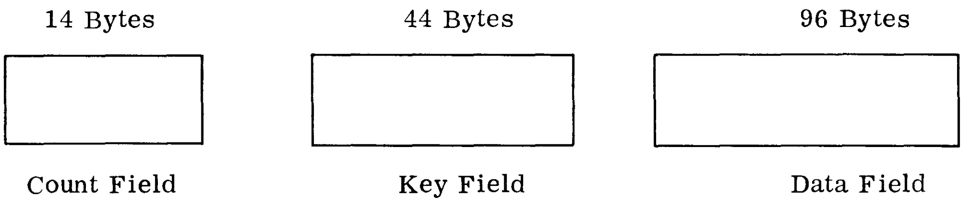
1 4	5 10	11	12 21	22 41	42 51	52 80
Label ID	Volume Serial Number		Data File Directory	Reserved	Owner ID Code	Reserved

Volume Security

Bytes	Content	Meaning
1-4	VOL1	Label identifier.
5-10	Volume serial number	Unique identification code assigned to a volume when it enters an installation. Normally 000001 to 999999, but may contain alphabetic characters.
11	Volume security	Indicates security status of volume: 0 = no security protection. 1 = volume has security protection.
12-21	Data File Directory	The first five bytes contain the starting address (CCHHR) of the VTOC. The last five bytes are blank.
22-41		Blank; reserved for future use.
42-51	Owner identification code	Identifies the owner of this volume.
52-80		Blank; reserved for future use.

VOLUME TABLE OF CONTENTS (VTOC)

◆ The Volume Table of Contents (VTOC) is a table area used to describe the contents of a volume. Each entry in the VTOC consists of the following fields.



Seven types of entries (labels) are used in the VTOC:

1.	Format 4 label:	This record describes the size of the VTOC and the location and size of the alternate track area. This is always the first entry in the VTOC.
2.	Format 5 label:	This contains the addresses of available (not-allocated) extents on the volume. This is always the second entry in the VTOC.
3.	Format 1 label:	This record is the file header label. It contains creation and expiration dates, addresses of file extents, and other information about a file.
4.	Format 3 label:	These are used as continuations of Format 1 records when room is needed to describe additional file extents. This record is not created unless a file has more than three extents.
5.	Format 2 label:	This record contains statistical, index, and overflow information for each indexed-sequential file.
6.	Format B label:	This is used to record available extents in 70/568 mass storage magazines (for indexed-sequential files only).
7.	Format C label:	This is a catalog in extent sequence order of all extents in all magazines containing this file. It is written in the VTOC of a disc or drum that contains the indexes for a mass storage indexed-sequential file.

Format 4 and format 5 labels are created by the RAINIT routine. The other formats are created by the RAALLR routine. In addition, the format 4 and format 5 labels are updated by the Allocator whenever files are allocated or deallocated.

The maximum number of VTOC records that can be stored on one track is:

15 records on one 70/564 track

12 records on one 70/565 track

1 record on one 70/568 track

VOLUME TABLE OF CONTENTS (VTOC)
(Cont'd)

When a full cylinder is assigned to the VTOC, the maximum number of VTOC records is:

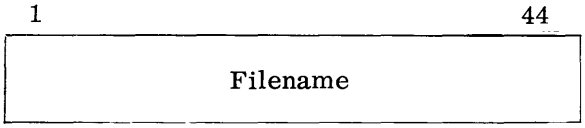
150 records on the 70/564
96 records on the 70/565
7 records on the 70/568

VTOC RECORD FORMATS

Format 1 Label

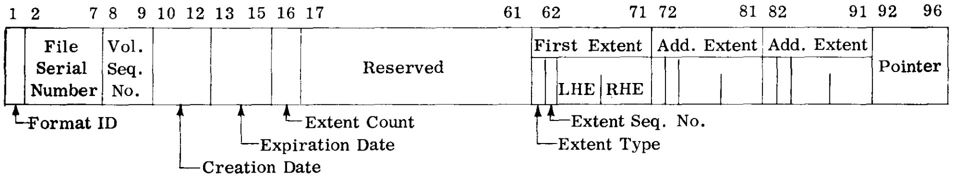
◆ This record is the standard file label for all files on direct access storage devices.

1. Key Field



Bytes	Content	Meaning
1-44	Filename	Name assigned to identify the file.

2. Data Field



Bytes	Content	Meaning
1	1	Label format identifier.
2-7	File Serial No.	The serial number of the first or only volume on which this file resides.
8-9	Volume Sequence No. (binary)	In multivolume files, the sequence number of the volume within the file, otherwise, 0000.
10-12	Creation Date (binary)	Year and day the file was created. YDD where Y=year (0-99) and DD=day (1-366).
13-15	Expiration Date (binary)	Year and day the file may be deleted. (Same form as above.)

Format 1 Label
(Cont'd)

<table>
<tr><th>Bytes</th><th>Content</th><th colspan="2">Meaning</th></tr>
<tr><td>16</td><td>Extent Count (binary)</td><td colspan="2">Number of extents for this file on this volume.</td></tr>
<tr><td>17-38</td><td></td><td colspan="2">Blank; reserved for future use.</td></tr>
<tr><td>39-40</td><td>File Type</td><td colspan="2">Indicates the physical organization of the data file:
HEX 4000 = consecutive
8000 = indexed-sequential incomplete
8080 = indexed-sequential completed</td></tr>
<tr><td rowspan="7">41</td><td rowspan="7">Record Format</td><td colspan="2">Indicates the type of records contained in the file:</td></tr>
<tr><td>Bit</td><td>Meaning</td></tr>
<tr><td>0-1</td><td>01 - variable-length record.
10 - fixed-length record.</td></tr>
<tr><td>2</td><td>Unused</td></tr>
<tr><td>3</td><td>1 - block records</td></tr>
<tr><td>4</td><td>1 - truncated records in file</td></tr>
<tr><td>5-6</td><td>01 - control character ASA code.
10 - control character machine code.
00 - control character not stated.</td></tr>
<tr><td>42</td><td></td><td colspan="2">Reserved for future use.</td></tr>
<tr><td>43-44</td><td>Block length (binary)</td><td colspan="2">Indicates block length for fixed-lemgth records or maximum block size for variable-length records.</td></tr>
<tr><td>45-46</td><td>Record length (binary)</td><td colspan="2">Indicates record length for fixed-length records or maximum record length for variable-length records.</td></tr>
<tr><td>47</td><td>Key length (binary)</td><td colspan="2">Indicates length of key portion of the data records in the file.</td></tr>
<tr><td>48-49</td><td>Key location (binary)</td><td colspan="2">Indicates high-order position of data record.</td></tr>
</table>

Format 1 Label *(Cont'd)*

<table>
<tr><th>Bytes</th><th>Content</th><th colspan="2">Meaning</th></tr>
<tr><td rowspan="4">50</td><td rowspan="4">Data Set Indicators</td><td>Bit</td><td>Meaning</td></tr>
<tr><td>0</td><td>If 1, this is last volume in file.</td></tr>
<tr><td>1</td><td>If 1, this file must remain in the same absolute location on the direct access device.</td></tr>
<tr><td>2-7</td><td>Unused.</td></tr>
<tr><td rowspan="4">51-54</td><td rowspan="4">Secondary Allocation (binary)</td><td colspan="2">Indicates the amount of storage to be requested for this data file when initial primary area is exhausted.</td></tr>
<tr><td>Byte 1</td><td>Type of allocation</td></tr>
<tr><td>Hex-E3
Hex-C3</td><td>=tracks
=cylinders</td></tr>
<tr><td>Bytes 2-4</td><td>=binary number indicating number of tracks or cylinder requested.</td></tr>
<tr><td>55-59</td><td>Last Record Pointer (binary)</td><td colspan="2">Points to last record written in a sequential or partition-organization data set. The format is TTRLL, where
TT = relative track address
R = ID of last record
LL = number of bytes remaining on the track following the last record
If this field does not apply, it will contain binary zeros.</td></tr>
<tr><td>60-61</td><td></td><td colspan="2">Blank; reserved for future use.</td></tr>
<tr><td>62</td><td>Extent Type Indicator</td><td colspan="2">Indicates the type of extent with which the following three fields are associated:
Hex code Meaning
00 = next three fields do not contain an extent.
01 = the extent containing user's data records.
02 = general overflow area of an indexed-sequential file.</td></tr>
</table>

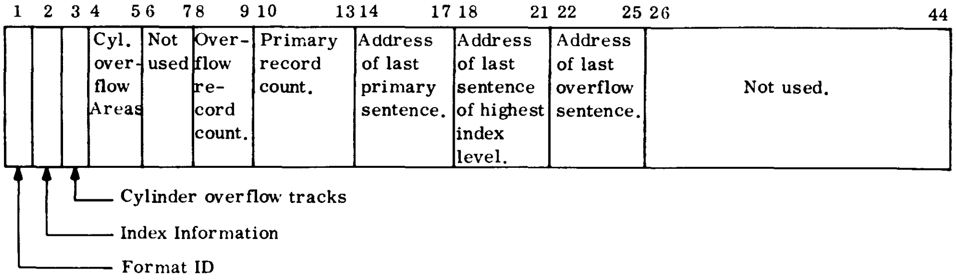
Format 1 Label *(Cont'd)*

Bytes	Content	Meaning
		Hex code — Meaning 04 = general index area of an indexed-sequential file. 40 = label cylinder; user label area and exchange area for indexed-sequential file buffers and buffer control blocks.
63	Extent Seq. No. (binary)	Indicates this extent's sequence in a multi-extent file.
64-67	Lower Limit (CCHH) (binary)	Cylinder and track address specifying start address of this extent.
68-71	Upper Limit (CCHH) (binary)	Cylinder and track address specifying ending address of this extent (70/564 and 70/567). Card, cylinder, and track(70/568).
72-81	Additional Extent	These fields have the same format as in bytes 62-71 above.
82-91	Additional Extent	These fields have the same format as in bytes 62-71 above.
92-96	Pointer	Contains the address of a continuation label if needed to further describe the file. If bytes 39-40 indicate indexed-sequential organization, this field will point to a Format 2 label. Otherwise it point to a Format 3 if there are more than three extents to be described for this file. Contains all binary zeros if no additional file label is pointed to.

Format 2 Label

◆ This record contains statistical, index, and overflow information for each indexed-sequential file.

1. *Key Field*



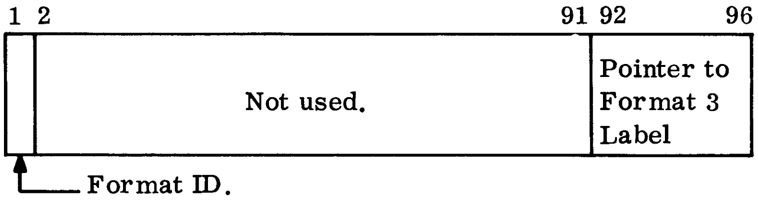
1	2	3	4 5	6 7	8 9	10 13	14 17	18 21	22 25	26 44
			Cyl. overflow Areas	Not used	Overflow record count.	Primary record count.	Address of last primary sentence.	Address of last sentence of highest index level.	Address of last overflow sentence.	Not used.

Format 2 Label
(Cont'd)

Bytes	Content	Meaning	
1	2	Label format identifier.	
2	Index information	Bit	Meaning
		0-3	If 1, indicates number of index levels, 0-3. (If bit 1 on, there is one index level; etc.)
		4	If 1, highest index level is a track index.
		5	If 1, cylinder index occupies less than one full track.
		6-7	Unused.
3	Cylinder overflow tracks	Number of cylinder overlfow tracks.	
4-5	Cylinder overflow areas	Number of full cylinder overflow areas.	
6-7		Not used.	
8-9	Overflow record count	Count of number of overflow records	
10-13	Primary record count	Count of number of primary data records	
14-17	Address of last primary sentence (TTTR)	Track number and record number of last primary sentence	
18-21	Address of last sentence of highest index level. (TTTR)	Track number and record number of last sentence of highest index level.	
22-25	Address of last overflow sentence.	Track and record number of last overflow sentence.	
26-44		Not used.	

Format 2 Label *(Cont'd)*

2. Data Field

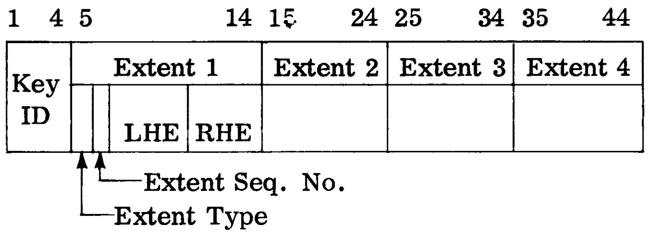


Bytes	Content	Meaning
1	2	Label format identifier
2-91		Not used.
92-96	Pointer (CCHHR)	Contains address of Format 3 label for this file if one exists.

Format 3 Label

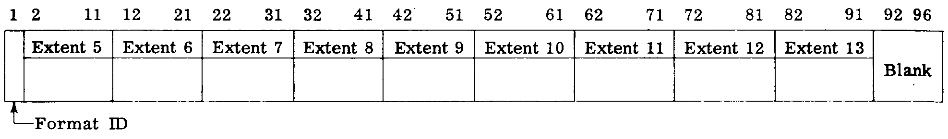
◆ This record is used to describe extents to a file that cannot be contained in a Format 1 label.

1. Key Field



Bytes	Content	Meaning
1-4	Key ID	Each byte contains Hex Code 03 to avoid conflict with a file name.
5-44	Extents in Key	Four 10-byte fields describing additional file extents identical in format to bytes 62-71 in Data field of Format 1 record.

2. Data Field



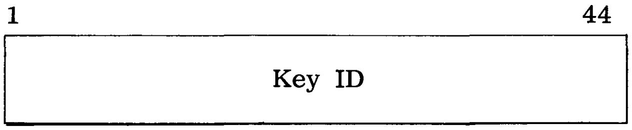
Format 3 Label *(Cont'd)*

Bytes	Content	Meaning
1	3	Label format identifier.
2-91	Additional Extents	Nine 10-byte fields identical in format to bytes 62-71 in Data field of Format 1 record.
92-96	Pointer (CCHHR)	Contains the address (in the form CCHHR) of another Format 3 label if one exists.

Format 4 Label

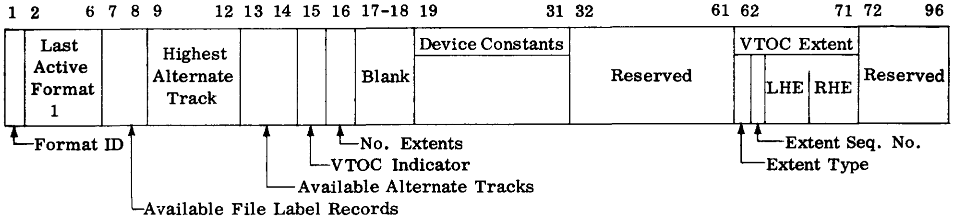
◆ This record is the label for the VTOC. It describes the size and limits of the VTOC and the location and size of the alternate track area.

1. Key Field



Bytes	Content	Meaning
1-44	Key ID	Each byte contains Hed Code 04 to avoid conflict with a file name.

2. Data Field



Bytes	Content	Meaning
1	4	Label format identifier.
2-6	Last Active Format 1	Contains the address (CCHHR) of the last active Format 1 file label.
7-8	Available File Label Records (binary)	A count of the number of unused records in the VTOC.
9-12	Highest Alt. Track	Highest address (CCHH) of block of tracks set aside as alternates for bad tracks.
13-14	Available Alt. Tracks (binary)	Number of alternate tracks available.
15	VTOC Indicators	Bit 0, if on, indicates no Format 5 label or the Format 5 label does not reflect the true status of the volume. Bits 1-7 are not used.

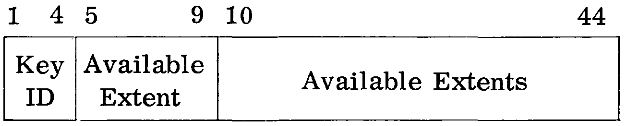
Format 4 Label
(Cont'd)

Bytes	Content	Meaning
16	No. Extents	Contains Hex 01 to indicate one extent in the VTOC.
17-18		Blank; reserved for future use.
19-31	Device Constants	This field describes the device on which the volume was mounted when the VTOC was created. 19-22 Device Size - number of cylinders and number of tracks per cylinder. 23-24 Track Length - number of available bytes on a track exclusive of home address and record 0. 25-30 Reserved for software use. 31 Labels/Track - number of labels that can be written on each track in the VTOC.
32-61		Blank; reserved for future use.
62-71	VTOC Extent	These fields describe the VTOC extent and are identical in format to bytes 62-71 in Data field of Format 1 record. Extent type is always 01; extent sequence number is always 00.
72-90		Not used.
91-92	Cylinders in magazine	Number of cylinders in the magazines excluding the flaw pool.
93-94	Cylinders available for allocation.	Number of cylinders in the magazine currently available for allocation.
95	Address of flaw pool.	Address of first card of flaw pool.
96	Indexed-Sequential indicator	If X '01', this is an indexed-sequential magazine.

Format 5 Label

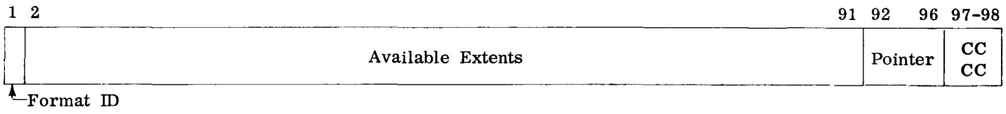
◆ This record describes available space on a random access volume.

1. *Key Field*



Bytes	**Content**	**Meaning**
1-4	Key ID	Each byte is a Hex 05.
5-9	First Available Extent	Indicates an extent of space available for allocation. Will be in either of two formats: (1) First two bytes are relative track address; next two bytes are the number of cylinders in the extent; last byte is 0. (2) First two bytes are relative track address; next two bytes are 00; last byte is the number of tracks available in this cylinder.
10-44	Available extents in Key.	Fields are identical to bytes 5-9 above. They are in track address sequence.

2. *Data Field*

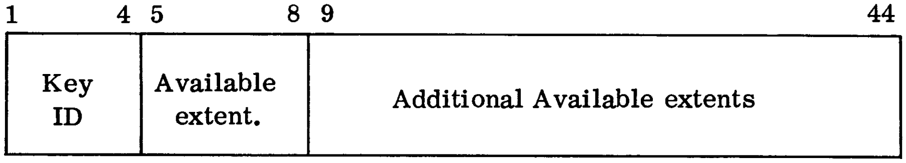


Bytes	**Content**	**Meaning**
1	5	Label format identifier.
2-91	Available extents	Fields are the same as bytes 5-9 in Key field above. There are 26 available extent fields in the Format 5 record (Key and data fields).
92-96	Pointer	Contains the address (CCHHR) of the next Format 5 label if one exists.

Format B Label*

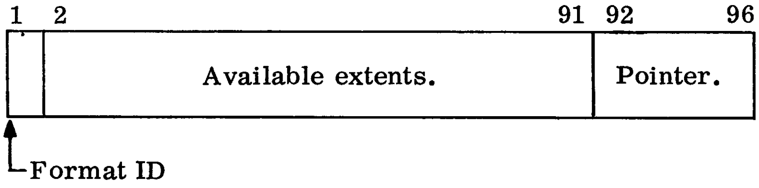
◆ This label is used to record available extents in a 70/568 mass storage magazine. Entries are in physical sequence.

1. Key Field



Bytes	Content	Meaning
1-4	Key ID	Each byte is a Hex OB.
5-8	Available extents	Indicates an extent of space available for allocation in this magazine. First two bytes are card number and cylinder number of left-hand end; next two bytes are card number and cylinder number of right-hand end.
9-44	Additional available extents	Fields identical to bytes 5-8 above.

2. Data Field



Bytes	Content	Meaning
1	B	Label format identifier
2-90	Available extents	Fields are the same as bytes 5-8 above.
91		Not used.
92-96	Pointer (CCHHR)	The address of the next format B label, if one exists. If no additional labels are needed, this field will be zeros.

Format C Label

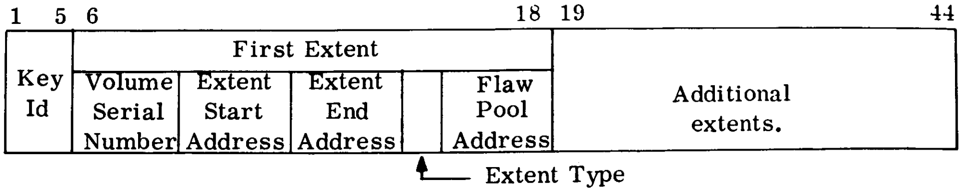
◆ This label is a catalog in extent sequence order of all the extents in all magazines containing this file. It is written in the VTOC of a disc or drum that contains the indexes for a mass storage indexed-sequential file.

---

*For 70/568 Mass Storage Indexed-Sequential files only.

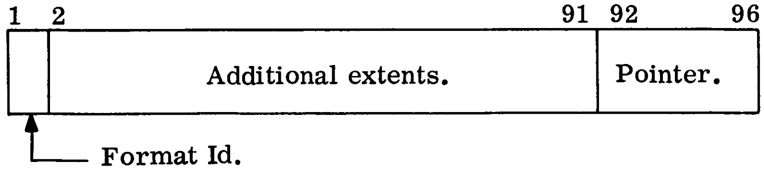
Format C Label
(Cont'd)

1. *Key Field*



Bytes	Content	Meaning
1-5	Key ID	Each byte is a Hex OC
6-11	Volume serial number	Serial number of magazine on which this extent of the file is recorded.
12-13	Extent start address	Start address of this extent. Card number and cylinder number.
14-15	Extent end address	Ending address of this extent. Card number and cylinder number.
16	Extent Type	Type of extent defined in the two previous fields. Hex code Meaning 01 = prmary data 02 = general overflow
17-18	Flaw Pool Address	The first and last card numbers of a series of cards in which alternate tracks will be assigned after CYLOF has been exhausted.
19-44	Additional extents	Fields are the same as bytes 6-18 above.

2. *Data Field*



Format C Label *(Cont'd)*

Bytes	Content	Meaning
1	C	Label format identifier.
2-79	Additional extents	Fields are same as bytes 6-18 in Key Field above.
80-91		Not used.
92-96	Pointer (CCHHR)	Address of next Format C label if needed to further describe this file.

APPENDIX D

MEMORY REQUIREMENTS

GENERAL

◆ The tables in this appendix list utility routing memory sizes, I/O block sizes (where applicable), and how each routine will use additional memory.

The routine memory size is what is allocated to the routine by the Executive when the routine is loaded normally. This memory size may be changed by specifying more memory in the E LOD message or by processing the routine through the Linkage Editor and changing the memory requirements by using the PROG parameter.

The block sizes given indicate the block length that will be processed by the routine with the normal memory size. The routine will double-buffer any block up to the size given and single-buffer any block between the double buffer size and single buffer size.

The remarks column describes how the routine will use additional memory. The memory required for a particular application may be calculated by using the information in the remarks column or the formulas given at the end of this appendix.

Note: ◆ Routine memory sizes may change when new TOS releases are made.

MEMORY TABLES

Peripheral Conversion Routines

Routine	Segments	Required Memory (bytes)	Input/Output Area		Remarks
			Max. Block Single Buffer (bytes)	Max. Block Double Buffered (bytes)	
CDPR	(ROOT) LOAD1 LOAD2 LOAD3 LOAD4 ITUTPB	12,280	IN: NA OUT: NA	80 132/160	Additional memory will be used for Field Select coding. (See note.)
CDTP	(ROOT) LOAD1 LOAD2 LOAD3 LOAD4 ITUTPB	13,552	IN: NA OUT: 1,000	80 500	Additional memory will be used first for Field Select coding, then for output area. (See note.)

MEMORY TABLES

Peripheral Conversion Routines (Cont'd)

Routine	Segments	Required Memory (bytes)	Input/Output Area		Remarks
			Max. Block Single Buffer (bytes)	Max. Block Double Buffered (bytes)	
DUP	(ROOT)	6,264	IN: 1,044 OUT: 1,044	NA NA	Additional memory is used for input/output area. Three bytes of memory are required for each byte in the input block over 1,044 bytes.
TPINIT	(ROOT)	4,808	NA	NA	None.
TPPR	(ROOT) LOAD1 LOAD2 LOAD3 LOAD4 ITUTPB	15,404	IN: 1,000 OUT: NA	500 132/160	Additional memory will be used <u>first</u> for Field Select coding, then for input area. (See Note.)
TPTP	(ROOT) LOAD1 LOAD2 LOAD3 LOAD4 ITUTPB	13,988	IN: 1,000 OUT: 1,000	500 500	Additional memory will be used <u>first</u> for Field Select coding, then for input/output area. (See note.)

Note:

The Peripheral Conversion routines generate Field Select coding according to the options given in the FS parameter(s). A 100-byte area is allocated for this coding. If more area is required, additional memory must be given to the routine when it is loaded. The additional field select area needed can be calculated as follows:

$$FS = 6\,(M + P + U + H) - 100$$

where

$M = \frac{s}{256} \uparrow$ for each field to be moved (s = size of field)

$P = \frac{n}{8} \uparrow +2$ for each field to be packed (n = size of field)

$U = \frac{n}{8} \uparrow +2$ for each field to be unpacked (n = size of field)

$H = \frac{n}{8} \uparrow +3$ for each field to be converted to hexadecimal (n = size of field)

If an option is not used, its value is 0.

Peripheral Conversion Routines (Cont'd)

Example:

$M \qquad U_1 \qquad\qquad U_2 \qquad\qquad H_1 \qquad\qquad H_2$

Δ FS Δ5,50,1/115 (U,8,16) 51/123 (U,25,50) 67/150 (X,2) 117/152 (X,2) 121

$FS = 6\,(M + P + U_1 + U_2 + H_1 + H_2) - 100$

$M = \frac{s}{256}\uparrow \quad U_1 = \frac{n}{8}\uparrow +2 \quad U_2 = \frac{n}{8}\uparrow +2 \quad H_1 = \frac{n}{8}\uparrow +3 \quad H_2 = \frac{n}{8}\uparrow +3$

$M = \frac{50}{256} \quad U_1 = \frac{8}{8} +2 \quad U_2 = \frac{25}{8} +2 \quad H_1 = \frac{2}{8} +3 \quad H_2 = \frac{2}{8} +3$

$M = 1 \quad U_1 = 3 \quad U_2 = 6 \quad H_1 = 4 \quad H_2 = 4$

$FS = 6\ (1+3+6+4+4) - 100$

FS = 8 bytes

Peripheral Conversion-Random Access

Routine	Segments	Required Memory (bytes)	Input/Output Area		Remarks
			Max. Block Single Buffer (bytes)	Max. Block Double Buffered (bytes)	
CDRA	(ROOT) LOAD1 LOAD2 LOAD3 LOAD4 ITUTPB	16,328	IN: NA OUT: 1,000	80 500	Additional memory will be used <u>first</u> for Field Select coding, then for output area. (See note.)
CDRAM	(ROOT) LOAD1 LOAD3 LOAD4 ITUTPB	20,196	IN: NA OUT: 1,000	80 500	Additional memory will be used <u>first</u> for Field Select coding, then for output area. (See note.)
RAINIT	(ROOT)	11,152	NA	NA	None.
RAPR	(ROOT) LOAD1 LOAD2 LOAD3 LOAD4 ITUTPB	15,904	IN: 1,000 OUT: na	500 132/160	Additional memory will be used <u>first</u> for Field Select coding, then for input area. (See note.)

Diagnostics

Routine	Segments	Required Memory (bytes)	Input/Output Area		Remarks
			Max. Block Single Buffer (bytes)	Max. Block Double Buffered (bytes)	
CARDCK	(ROOT)	368	N/A	N/A	None.
DATGEN	(ROOT)	7,848	2,000	N/A	Additional memory will allow a greater output block size.
DUMPRT	(ROOT)	2,424	N/A	N/A	None.

Peripheral Conversion-Random Access (Cont'd)

Routine	Segments	Required Memory (bytes)	Input/Output Area		Remarks
			Max. Block Single Buffer (bytes)	Max. Block Double Buffered (bytes)	
RARA	(ROOT) LOAD1 LOAD2 LOAD3 LOAD4 ITUTPB	16,976	IN: 1,000 OUT: 1,000	500 500	Additional memory will be used first for Field Select coding, then for input/output area. (See note.)
RARAM	(ROOT) LOAD1 LOAD3 LOAD4 ITUTPB	20,896	IN: 1,000 OUT: 1,000	500 500	Additional memory will be used first for Field Select coding, then for input/output area. (See note.)
RATP	(ROOT) LOAD1 LOAD2 LOAD3 LOAD4 ITUTPB	16,408	IN: 1,000 OUT: 1,000	500 500	Additional memory will be used first for Field Select coding, then for input/output area. (See note.)
TPRA	(ROOT) LOAD1 LOAD2 LOAD3 LOAD4 ITUTPB	16,832	IN: 1,000 OUT: 1,000	500 500	Additional memory will be used first for Field Select coding, then for input/output area. (See note.)
TPRAM	(ROOT) LOAD1 LOAD3 LOAD4 ITUTPB	20,688	IN: 1,000 OUT: 1,000	500 500	Additional memory will be used first for Field Select coding, then for input/output area. (See note.)

Note:

The Peripheral Conversion routines generate Field Select coding according to the options given in the FS parameter(s). A 100-byte area is allocated for this coding. If more area is required, additional memory must be given to the routine when it is loaded. The additional field select area needed can be calculated as follows:

$$FS = 6 (M + P + U + H) - 100$$

Diagnostics (Cont'd)

Routine	Segments	Required Memory (bytes)	Input/Output Area		Remarks
			Max. Block Single Buffer (bytes)	Max. Block Double Buffered (bytes)	
RAEDIT	(ROOT)	9,336	N/A	N/A	None.
TOSAID	(ROOT) CONRES AIDINT CONONE CONTWO CONTHR CONFOR CONFIV CONSIX CONSEV CONATE CONNIN CONTEN ANTONE AUTTWO AUTTHR AUTFOR AUTFIV AUTNIN AUTELV AUTTEN AUTTWL AUTTIR AUTFRT AUTFVT AUTSVT AUTATT AUTSTX AUTATE	9,912	N/A	N/A	Additional memory is used for working and parameter storage.
TPCOMP	(ROOT)	8,600	500	250	Additional memory is used for input area.
TPEDIT	(ROOT)	5,988	500	250	Additional memory is used for input area.

System Maintenance Routines

Routine	Segments	Required Memory (bytes)	Input/Output Area		Remarks
			Max. Block Single Buffer (bytes)	Max. Block Double Buffered (bytes)	
CLU	(ROOT)	9,744	N/A	N/A	None.
DDRL	(ROOT) ITURPM ITURDD ITURWR	13,448	N/A	N/A	None.
LLU	(ROOT)	21,192	N/A	N/A	Additional memory is used for processing tables.
LNKEDT	(ROOT) LINK1 LINK2 LINK3 LINK4 LINKX	32,768	N/A	N/A	Additional memory is used for processing tables (module, entry, extrn, load and V-type items).
MLU	(ROOT)	13,544	N/A	N/A	None.
OMLU	(ROOT)	26,264	N/A	N/A	The OMLU contains a table which can contain 100 entries. One entry is made for each module to be merged, extracted, or added. If more than 100 entries are expected, add 12 bytes to the memory size for each additional entry.
RAALLR	(ROOT)	10,512	N/A	N/A	Additional memory is used for internal processing storage.
RAINDX	(ROOT)	3,792	N/A	N/A	None.
RAMSUP	(ROOT) ITURPV ITURCU ITURSC ITURCP ITURRR ITUREO ITURCI ITURTR ITURSO	7,784	N/A	N/A	Additional memory must be allocated when using TANK function. See TOS Utility Manual page 7-26.

System Maintenance Routines (Cont'd)

Routine	Segments	Required Memory (bytes)	Input/Output Area		Remarks
			Max. Block Single Buffer (bytes)	Max. Block Double Buffered (bytes)	
SLU	(ROOT)	28,232	N/A	N/A	The SLU contains 1,000 bytes for storage of Level I action and reorder entries. Action entries use 20 bytes for each program named in a RENAME DELETE, EXTRACT, OUTPUT, PRINT, or PUNCH card. Reorder entries use two bytes for each device named in a REORDER card. If the total number of entries require more than 1,000 bytes, allocate additional memory as required.
TPMAIN	(ROOT)	24,252	4,000 (combined input/output)	N/A	Additional memory is used for input/output area.

MEMORY FORMULAS

Peripheral Conversion Routines

◆ $MR = S + \left[n_i (BS_i - B_i)\right] + \left[n_o (BS_o - B_o)\right] + FS$

where:

MR = memory size requirement

S = memory size of routine

n_i = 1 for single buffer input or 2 for double buffer input

BS_i = maximum input block size (cannot be > 4,095)

B_i = input buffer size (when $n_i = 1$, B_i must equal single buffer size; when $n_i = 2$, B_i must equal double buffer size)

n_o = same as n_i except for output

BS_o = same as BS_i except for output

B_o = same as B_i except for output

FS = additional field select coding area (see note following Peripheral Conversion routines table)

Example: ◆ The programmer wishes to run the TPTP routine; single buffering 1,500 byte input blocks and double buffering 3,000 byte output blocks. No additional field select coding area is needed.

$$MR = S + \left[n_i\,(BS_i - B_i)\right] + \left[n_o\,(BS_o - B_o)\right] + FS$$

$$MR = 14,032 + 1\,(1,500 - 1,000) + 2\,(3,000 - 500) + 0$$

$$MR = 19,532 \text{ bytes}$$

DATGEN ◆ $MR = S + (BS_o - B_o)$

Example: ◆ The programmer wants to generate variable-length blocks between 1,500 and 3,000 bytes.

$$MR = S + (BS_o - B_o)$$

$$MR = 7,848 + (3,000 - 2,000)$$

$$MR = 8,848 \text{ bytes}$$

TPCOMP ◆ $MR = S + \left[m_i\,(BS_i - B_i)\right]$

where:

m_i = 2 for single buffer input or 4 for double buffer input.

Example: ◆ The programmer wishes to double buffer 1,000 byte input blocks.

$$MR = S + m_i\,(BS_i - B_i)$$

$$MR - 7,768 + 4\,(1,000 - 250)$$

$$MR = 10,768 \text{ bytes}$$

LNKEDT ◆ $MR = S + 28L + 20M + 14\,(E + V + U) - 4,792$

where:

L = number of loads in the program

M = number of modules in the program

E = number of entries in the program

V = number of VCONS in the program

U = number of unsatisfied extrns in the program

Example: ◆ The programmer wishes to bind a program consisting of the following items:

48 loads
60 modules
400 entries
50 VCONS
37 unsatisfied extrns

$$MR = S + 28L + 20M + 14\,(E + V + U) - 4792$$

$$MR = 32,768 + 28\,(48) + 20\,(60) + 14\,(400 + 50 + 37) - 4,792$$

$$MR = 37,338 \text{ bytes}$$

TPMAIN

◆ $MR = S + \left[(I + O_1 + O_2 + O_3) - 4{,}000 \right]$

where:

I = maximum input block size

O_1 = maximum first output block size

O_2 = maximum second output block size

O_3 = maximum third output block size

Example:

◆ The programmer wishes to copy an input tape with 2,000 byte blocks to an output tape, reblocking to 3,000 bytes, also listing the input on a 132-character printer.

$MR = S + (I + O_1 + O_2 + O_3) - 4{,}000$

$MR = 23{,}656 + (2{,}000 + 3{,}000 + 132 + O) - 4{,}000$

$MR = 24{,}788$